Crossing the Chasm:

Deploying Machine Learning Analytics in Dynamic Real-World Scenarios

by

Som Shahapurkar

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2016 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Hasan Davulcu
Ashish Amresh
Jingrui He
Eugene Tuv

ARIZONA STATE UNIVERSITY

December 2016

ABSTRACT

The dawn of Internet of Things (IoT) has opened the opportunity for mainstream adoption of machine learning analytics. However, most research in machine learning has focused on discovery of new algorithms or fine-tuning the performance of existing algorithms. Little exists on the process of taking an algorithm from the lab-environment into the real-world, culminating in sustained value. Real-world applications are typically characterized by dynamic non-stationary systems with requirements around feasibility, stability and maintainability. Not much has been done to establish standards around the unique analytics demands of real-world scenarios.

This research explores the problem of the why so few of the published algorithms enter production and furthermore, fewer end up generating sustained value. The dissertation proposes a 'Design for Deployment' (DFD) framework to successfully build machine learning analytics so they can be deployed to generate sustained value. The framework emphasizes and elaborates the often neglected but immensely important latter steps of an analytics process: 'Evaluation' and 'Deployment'. A representative evaluation framework is proposed that incorporates the temporal-shifts and dynamism of real-world scenarios. Additionally, the recommended infrastructure allows analytics projects to pivot rapidly when a particular venture does not materialize. Deployment needs and apprehensions of the industry are identified and gaps addressed through a 4-step process for sustainable deployment. Lastly, the need for analytics as a functional area (like finance and IT) is identified to maximize the return on machine-learning deployment.

The framework and process is demonstrated in semiconductor manufacturing – it is highly complex process involving hundreds of optical, electrical, chemical, mechanical, thermal, electrochemical and software processes which makes it a highly dynamic non-stationary system. Due to the 24/7 uptime requirements in manufacturing, high-reliability and fail-safe are a must. Moreover, the ever growing volumes mean that the system must be highly scalable. Lastly, due to the high cost of change, sustained value proposition is a must for any proposed changes. Hence the context is ideal to explore the issues involved. The enterprise use-cases are used to demonstrate the robustness of the framework in addressing challenges encountered in the end-to-end process of productizing machine learning analytics in dynamic read-world scenarios.

DEDICATION

I dedicate this work to, my mother Aruna and father Suhas who molded me early in life, my wife Sushma and daughter Sanika who encouraged and supported me every step of the way, any dear friend Dharamarajan who was with when I needed a friend.

ACKNOWLEDGMENTS

I am grateful Dr. Huan Liu for being my teacher and guide for many years now. He was the first one outside family to encourage me that my dream of earning a doctoral degree while working full-time was indeed possible. He has been always encouraging and immensely patient. I thank him for guidance to me, his service to his students and the community. He is one of the best human-beings I know.

Dr. Hasan Davulcu is a role-model of receptiveness. In his classroom he was always open to ideas from all students. This made the learning experience rich and enduring. Dr. Ashish Amresh encouraged me during the most testing period of my journey. He imparted invaluable know-how on how to start and make progress drawing from his own work-study experiences. I thank Dr. Jingrui He for graciously agreed to join my committee when I was in dire need. Her work on application of machine-learning to semiconductor manufacturing is most impressive. The ASU center for entrepreneurship imparted to me the best education I have had outside of formal classrooms. It helped broaden my perspective on bring a product to market.

Dr. Eugene Tuv is my closest collaborator in analytics – I am honored to earn his respect. He calls be “brother” and truly has been one in terms of guiding me in the world of analytics. His trusted team-leader Alexander Borisov helped turnaround feature-requests and changes to analytics suite IDEAL used in this research with lightning speed. Randal Goodwin is my hired me into Intel and he is one of the best manager I have had. He encouraged me to pursue academic advancement by backing my application for tuition-reimbursement, extending flexible-time and opening up opportunities for me in the analytics space. He is an eternal ‘possibility-person,’ always full of ideas.

Jason Garcia is the patient project-manager for this project which spans several years. I envy his patience and attention-for-detail which he used to untangle the myriad of complex interdependencies to make deployment of the ML solution possible. A lot of this work would not have been possible if not for scripting wizardry by Tik Linh Loh – he had a large part to play in implementing the simulation framework. Another key participant is Arvind Krishna – he led the system-implementation efforts. He was always open to inputs and changes. His system requirements document is one of the crispest I have seen.

I have had the good fortune of receiving encouragement from all directions. Dr. Karl Kempf an Intel Fellow (technical Executive Vice President) took time out of his impossible schedule to encourage and guide me. His advice and unwavering encouragement put be back on track for my research. Dr. Rana Pratap on my team who later became my partner encouraged me to launch a startup using some of my research work. The experience greatly enriched my journey – I thank him for that. Additionally, I would like to remember the numerous friends, family, colleagues and well-wishers who helped me along the way by various acts of kindness or words of encouragement – it does take a village.

Lastly, I would like to thank Intel for supporting most of my doctoral education through the tuition reimbursement program. Intel is the creator of Silicon Valley and birther of venture capitalism - it was a great company to work for.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

The Internet of Things efforts have taken predictive analytics from a niche academic interest to mainstream visibility across enterprises [1] and even general public in the last few years [2]. This has led to an eagerness to deploy big-data analytics systems sometimes without much heed to ensuring technical viability or economic feasibility [3]. The expectation in industry seems to be that by mere use of large quantities of data and "smart" algorithms, valuable results will follow [4].

On the other hand, most machine learning undertakings in academia are focused on algorithm development to improve accuracy [5]. Typically, after identifying a domain, data is gathered, painstakingly cleaned and new algorithms or amalgamation of existing ones are fashioned. Their performance is compared to both data and algorithm benchmarks and the results are published [6]. However, only a small percentage result in viable products that add value as illustrated in Figure 1.

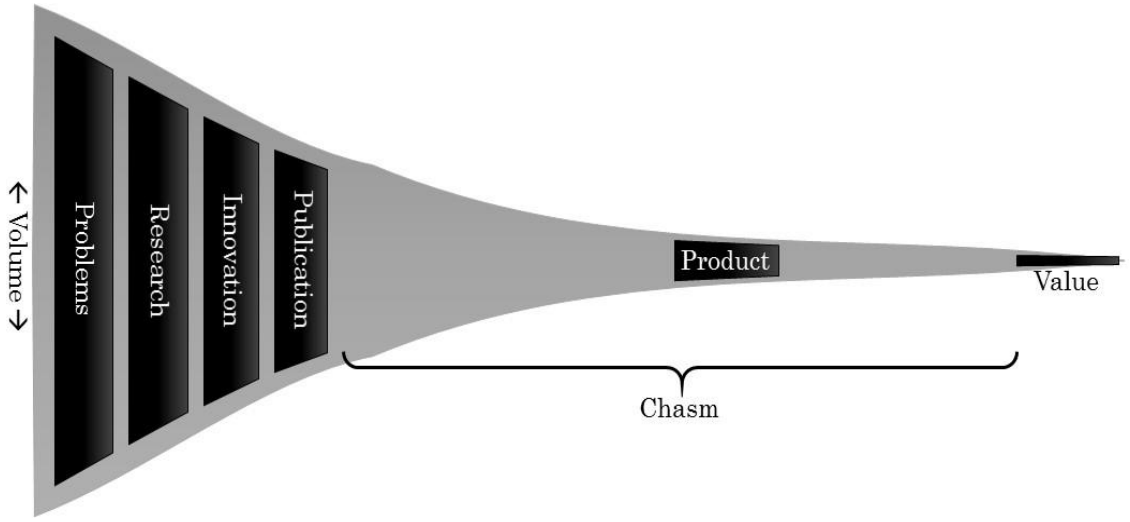


Figure 1. Small % of "New" ML Algorithms Become Products that Add Value

The results of the Netflix Challenge of 2006 are indicative of the state of affairs. Over 51,051 contestants from 186 countries competed over a period of two years with the goal of identifying a predictive algorithm that would beat the state-of-art method used by Netflix for movie-rating prediction by more than 10%. The organizers concluded: *"We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment."* [7]

Another headline from a prominent technical online publication in 2016 read, *"Facebook fires human editors, algorithm immediately posts fake news."* The article was referring to how Facebook promoted a story for 8+ hours on its "Trending" page before realizing that the news in question was false (due to a long chain of misquotes). Incidents like this breed mistrust and skepticism in machine-learning algorithms and are preventable if systematic evaluation and deployment processes are followed.

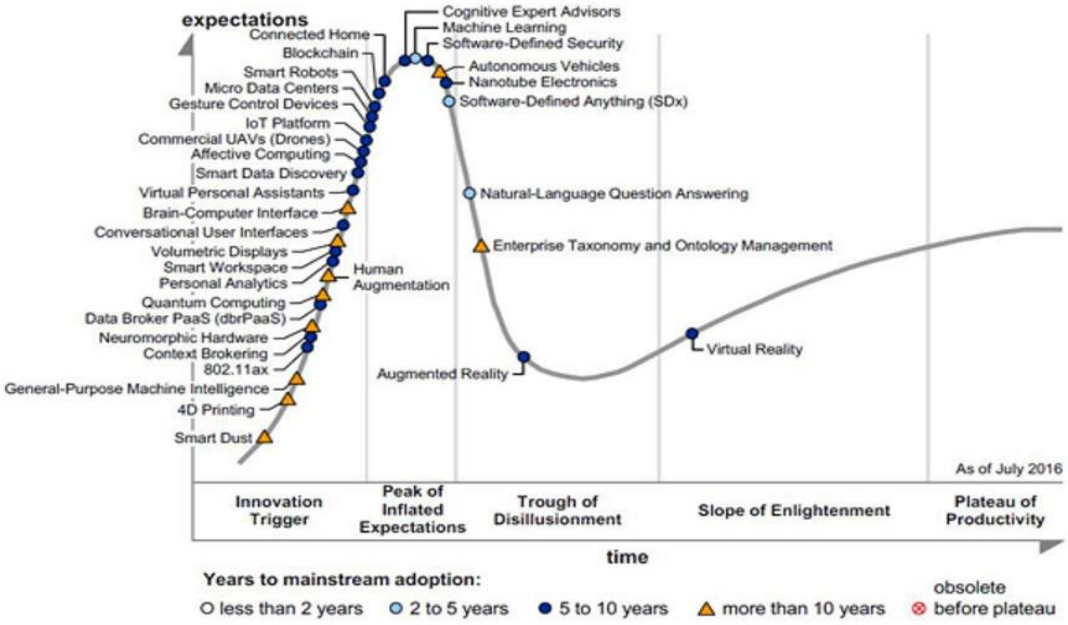


Figure 2. Gartner Hype-cycle (2016) for Emerging Technologies

The dichotomy is clear: whereas both academia and industry focus on the algorithm, there is little to show in terms of tangible results [8] [9]. The conundrum exists in part due to focus on the algorithms whereas the value lies in building deployable validated products [10]. The 2016 Gartner hype cycle for emerging technologies shown in Figure 2 puts Machine-Learning at the *'peak of inflated expectations'*. Thus while expectations are currently high, soon to follow is the *'trough of disillusionment',* which is characterized by reduced funding for projects. The technologies that have crossed this chasm to become mainstream, are those where the community has collaborated on a common standard investing in the often seemingly mundane tasks of stringent evaluation and integration methodologies.

Computer science went through a similar juncture before reaching its current form. In its early days, software development focused on creating programming languages. Continuous project planning delays, low productivity, heavy maintenance expenses and failure to meet user expectations had led by 1968 to the software crisis, the term coined at the first NATO conference on software development. This crisis was caused by the fact that there were no formal methodologies. The software community began to assimilate ideas from other fields of engineering into software project development - this was the origin of software engineering (SE). The field of machine-learning needs a similar supporting discipline due to the uniqueness of learning systems. The CRISP-DM (Cross Industry Standard Process for Data Mining) was proposed in 1999 to guide the industry on large-scale adoption of analytics. Although it has been widely employed since, it focuses on knowledge extraction and not necessarily value extraction. There is an urgent need to extended the framework to cover the unique demands of real-world scenarios.

1.2 Claims

This work is focused on identifying the challenges, pitfalls and nuances faced in taking an algorithm from the lab environment into the real-world. The thesis is that unless the many interesting algorithms are put to work outside of the lab and research world to serve humanity, the dream of an analytics driven world will remain a dream.

The second claim is that there is lack of a well-defined methodology for development, evaluation, deployment and maintenance of machine-learning analytics in the industry. Although frameworks like CRISP-DM exist, they were developed in an era where knowledge extraction was the primary focus of analytics and application of the results into the field was still assumed to be manual. In the authors interviews with some of the architects of CRISP-DM, they confirmed that the group agreed to declare that mission-critical systems were out of scope of the methodology.

The third claim is that the latter phases of evaluation and deployment of ML algorithms are heavily underserved. The current frameworks and methodologies focus on the front (business case and data understanding) and middle (data-precreation and model-development) phases. With the proliferation of cleaning tools and algorithm packages, the former phases are headed towards standardization with availability of PMML. The latter phases of evaluation and deployment remain underserved. The issue is being raised in industry conferences and leaders in large-scale users of analytics like Google, Facebook, LinkedIn, Netflix, Microsoft and Intel. However, there is not much activity in the research community although this is a ripe area for research.

The fourth claim is that the use of accuracy as a metric and cross-validation as a methodology for evaluation and comparison of machine learning algorithms is misguided. There could be cases where the accuracy is deceptively high whereas the algorithm need not be much better than a guess – in most rare-class scenarios [5] pp11. Similarly, cross-validation does not have the mathematical properties of hold-out validation that ensure the estimate is close to reality. As data-set size decreases, the variation of cross-validation skyrockets thus rendering it an unreliable estimate.

The fifth claim is that evaluations for real-world scenarios require that the temporal gaps and shifts be included. There are unavoidable time-gaps and shifts in the real-world between model training and prediction that have a large impact on algorithm performance. Lab evaluations of without these temporal adjustments are overly optimistic thus leading to lofty expectations of machine-learning algorithms.

The sixth claim is that all models need to be updated, refreshed or retrained in real-world scenarios as systems have multiple levels of dynamics that cannot all be included in the model realistically.

The seventh claim is that for successful deployment of machine learning algorithms it is critical that attention is paid various deterrents and address them in a systematic way using available incentives.

The eight claim is that machine-learning just like other types of development need a strong architecture that results in a happy transplant into the broader system and design-patterns that ensure sustainability.

The ninth claim is that analytics is a permanent function just like finance, IT, marketing, sales etc. Without such a function, analytics will never bear the full fruit of its potential.

1.3 Organization

The following chapters elucidate the claims made in the previous section and then propose solutions to address the same. The rest of this explains the domain of semiconductor manufacturing and why this is a representative case to demonstrate the key contributions of this work.

Chapter 2 starts by justifying why analytics frameworks are necessary in the first place. Next the current frameworks are mapped by their suitability for industry and academic as well as if they are human centric versus data-centric thus providing a mental map of the state of the art in section 2.2. As the idea is to propose a framework that incorporates all the good that has already been developed, each of the salient frameworks till date are scanned and unique contributions and advancements are highlighted in section 2.3. In general, the frameworks evolved by focusing on the central algorithm development theme, then expanding into the front-end business-case and only lately has back-end phases received attention. Thus gaps are then identified. Based on the preceding analysis, the *design for deployment* (DFD) framework is proposed in 2.5 with emphasis on continuous validation. The framework is then demonstrated in semiconductor manufacturing in section **Error! Reference source not found.** and results are discussed.

Chapter 3 is dedicated to the evaluation phase of analytics projects. Section 3.1 explains the problem and section **Error! Reference source not found.** scans the state of the art focusing on the purpose, metrics and methods of evaluation. The next section sheds light on the temporal characteristics in real-world that greatly affect evaluation results and are not necessarily considered or exist in a lab environment. Base the prior analysis, a three-pronged approach is recommended in section 3.2.4

for *dynamic evaluation* of machine learning algorithms such that the results closely estimate performance in real-world scenarios. Finally, the evaluation framework is demonstrated on the enterprise use-case and results are discussed.

Chapter 4 is dedicated to the deployment phase of the analytics process. Section 4.1 covers the problem statement and current state, hypothesizing the cause for the reason for the gap in robust deployment methodologies. Section 4.2, lists the real-world challenges, apprehensions and nuances of deploying machine-learning in real-world settings. Based on the previous analysis, a systematic four-pronged approach is recommended to ensure certain non-negotiables, enable certain capabilities, evaluate the important elements and establish processes and practices that ensure smooth and *sustainable deployment*. The approach is demonstrated on a deployment in semiconductor manufacturing with impressive results.

Chapter 5 provides a summary of the work and highlights the key contributions. Importantly, it points out areas for further research and cross community collaboration towards standards and practices that would benefit the analytics community and entire industry as a whole.

1.4 Domain

Semiconductor manufacturing is one of the most complex multi-step processes in the world [11]. It is a highly dynamic environment and easily one of the largest data-systems on the planet. The manufacturing environment places strict constraints on system reliability and stability [12]. Moreover, due to the high cost of change, proven value is hard requirement to any proposed changes to the system. Hence semiconductor context is ideal to explore industrial deployment issues [13]. The research was done in the context of semiconductor manufacturing at Intel.

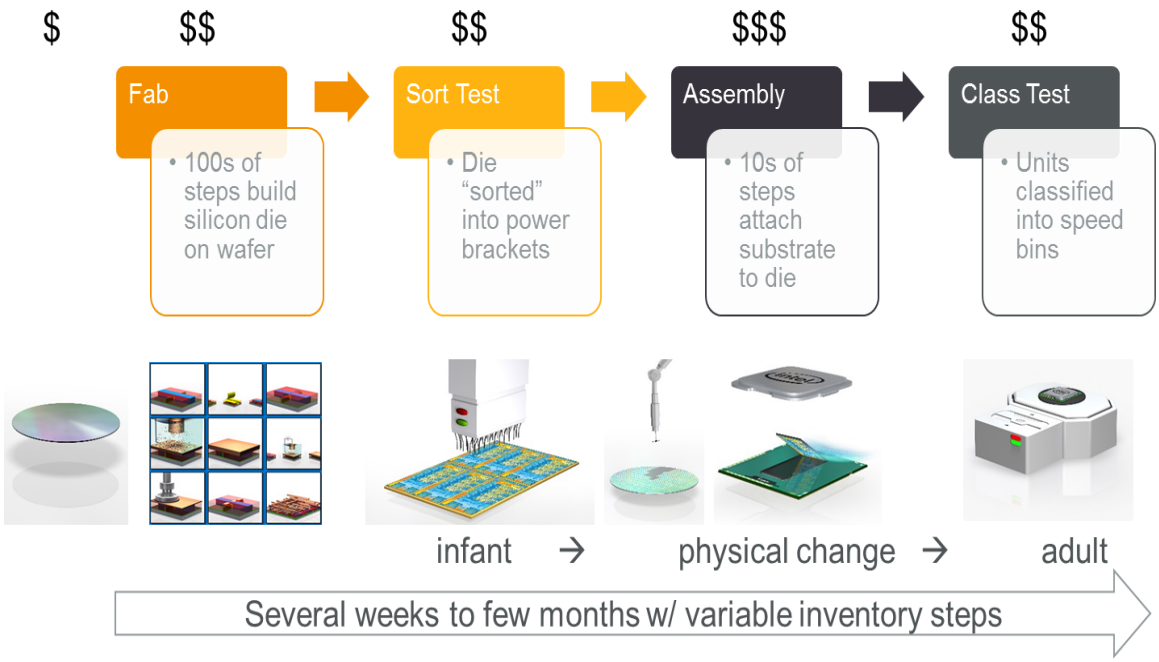


Figure 3. Simplified Semiconductor Manufacturing Process

A simplified Semiconductor Manufacturing flow is illustrated in Figure 3. Fabs "grow" transistors on bare wafers through hundreds of complex photo-electrochemical process steps. The transistors are arranged to form several-hundred die on the wafer – essentially each die ultimately becomes part of a microprocessor. At the end of the *Fab* process is a test step called *Sort-Test* that ensures the die are

good and sorts the die into different power buckets, based on the current drawn by the die. Power consumption is one of the defining characteristics of a microprocessor as it determines the type of market or application the microprocessor is sold into. Additionally, other measurements are sampled that help engineers obtain insights into the characteristics of the die as well as Fab process health. The sort-test results are stored in a database to be used later. Subsequent process steps laser-score and saw the die to separate them. Then the die are picked out in multiple sweeps of the pick and place robot. Each sweep picks die of the same power bucket (based on sort-test results from the database) onto a tape which is wrapped around a reel. The reels are stored as inventory or sent along (kitted) to the assembly factory based on customer orders for parts of certain power and speed bucket.

In the *Assembly* factory, die are picked off the reels and placed on substrates (AKA packages). Substrates are high density circuit boards that allow the semiconductor die to electrically connect to motherboards. The assembly process involves several steps like under-fill application, flux application, solder-ball attach, and heated in an oven for the solder-ball to reflow and thus establish contact between the bumps on the die with pads on the substrate. Thus the die goes through physical change during assembly and hence its electrical characteristics have also changed. A series of test steps ensure the die are good and the *Class-Test* step "bins" the packaged die (AKA 'Unit') into speed bins based on how fast the unit is able to run. The average selling price (ASP) of units is highly dependent on the power bucket and speed bin of the unit. Most of the Fab, Sort, Assembly and Test steps are batch processes that process lots. Multiple tools and chambers can be deployed for the production of this device [14].

Due to the variation introduced by material, machines, methods, metrics, personnel and environment of the manufacturing flow, the resulting speed and power bins of each unit can vary widely from lot to lot, wafer to wafer, within wafer, and lastly unit to unit. While process engineers in the factory try to keep this variation to a minimum to obtain higher yields [15] (proportion of within spec or "good" parts), residual variation is unavoidable especially at the early stages of a new process generation. However due to the continual efforts to minimize variation, and keep yields high, tests are removed and added and (controlled) process changes are being made. This makes the system highly dynamic [16].

The whole process can take anywhere from several weeks to a few months. The variation in processing times is partly due to inventory and "mix" optimizations in the flow. The other reason for the variation is because of the 'virtual factory' concept. To ensure manufacturing efficiency and efficiency of scale, Intel factories are distributed all over the world hence a certain processor could start its journey as a die in a Fab/Sort facility at one location in the USA, and then be assembled and class-tested in Malaysia. Each site and factory has its own database to store manufacturing and test data due to the size of the data involved and given that the data is used in subsequent steps of manufacturing which needs 24/7 uptime.

There is great value in being able to predict behavior of units down the manufacturing flow based on current measurements [12]. The predictions can be used for optimized processing [15]. There are a few 100 process steps and each step can generate several hundred test and monitoring readings. This combined with the millions of units manufactured each week results in several Terabytes of data generated every week that goes into Peta-Scale data systems.

CHAPTER 2

DESIGN FOR DEPLOYMENT (DFD)

This starts with illucidating the need for analytic frameworks. This is followed by a survey of existing frameworks highlighting the unique contributions of each. Lastly the shortcoming of the existing frameworks are discussed in the context of current nature of analytics. Thus a case is made for extending the frameworks. Finallly a new framework is proposed and then demostrated in a real-world case.

2.1 Need for Frameworks

It has been argued that all one needs to engage in data science is data and a willingness to "give it a try". Although this view is attractive from the perspective of enthusiastic data-science consultants who wish to expand the use of the technology, it can only serve the purposes of one-shot proofs of concept or preliminary studies. It is not representative of the reality of deploying data-science within existing business processes. In such contexts, one needs two additional ingredients: a framework or methodology, and supporting tools [17].

With increasing popularity of tools and readily available data, the risk of data-dredging still remains – drawing nonsense conclusions from blind application of algorithms to data. Standardization leads to higher trust in the results derived from proper application of the framework thus helping subside skepticism and promote acceptance of machine-learning based systems. Additionally, some analytics projects can take months to complete and hence the sponsors need to see milestones that can be met to ascertain progress in order to continue sponsorship [18]. This becomes especially challenging as data-science is a creative process requiring many iterations, skills and knowledge. Without a standard framework, the success or

failure of a data mining project is highly dependent on the particular person or team carrying it out and successful practice can not necessarily be repeated across the enterprise. Data-science needs a standard approach which will help translate business problems into data science tasks, suggest appropriate data transformations and modeling techniques, and provide means for evaluating the effectiveness of the results and documenting the experience [19]. There are several reasons to have a standardized framework:

1. The end product must be useful for the sponsor: A blind, unstructured application of machine-learning techniques to input data, called data dredging, frequently produces knowledge that, while interesting, may not contribute to solving the business problem. Thus leading to the failure of the project. Only through the application of well-defined analytic frameworks will the end product be valid, novel, useful, and understandable [18].
2. The sponsor must understand (be comfortable with) the results: Decision makers often do not want to devote significant time and resources to the pursuit of formal data-analysis methods, but rather prefer to rely heavily on the skills and experience of domain experts as their source of information. However, because they are ultimately responsible for the decision(s), they frequently want to understand (be comfortable with) the technology applied to those solutions. A well-defined logical framework can be presented to decision-makers who may have difficulty understanding the need, value, and mechanics behind analytics thus quelling any misgivings they may have.

3. The sponsor must see progress of the project to continue funding: Data analytics undertakings need significant amount of resources, effort and time. Thus they take time to bear fruit. The sponsors need to see milestones that can be met to ascertain progress in order to continue sponsorship. Without a standard framework, it would be difficult for the sponsors to ascertain progress as they are typically not well-versed in details of the process or the risks involved.
4. The sponsor must be made aware of the risks: Not all analytics projects result in success. Each phase comes with its own risks. For example, the data owned by the enterprise might not have all features needed to build a reasonable model. The model may not result in a significantly better performance than the incumbent method or may require cost-prohibitive changes to the process. Data science practitioners should ensure that the potential contributions are not overstated and that users understand the true nature along with their limitations [20]. Without a structured, model to highlight risks the sponsor could have unrealistic expectations and/or hold the data-scientist responsible for the failure.
5. Data-science projects require project management: Much like any other development effort, project-management for data-science effort needs to be grounded in a solid framework. Most data-science projects involve teamwork and thus require careful planning and scheduling. For most project management specialists, data-science is a new realm. Therefore, these specialists need a definition of what such projects involve and how to carry them out in order to develop a sound project schedule [18].

6. There is a widely recognized need for standardization of data-science: The challenge for modern data scientists is to come up with widely accepted standards that will stimulate major industry growth. Standardization of the frameworks would enable the development of standardized methods and procedures, thereby enabling enterprises to deploy their projects more easily. It would lead directly to project performance that is faster, cheaper, more reliable, and more manageable. The standards would promote the development and delivery of solutions that use business terminology rather than the traditional language of algorithms, matrices, criterions, complexities, and the like, resulting in greater exposure and acceptability for the field.

In its early days, software development focused on creating programming languages and algorithms that were capable of solving almost any problem type. The software crisis of 1968 was a culmination of evolving hardware, continuous project planning delays, low productivity, heavy maintenance expenses, and failure to meet user expectations. This crisis was caused by the fact that there were no formal methods and methodologies, support tools or proper development project management. The software community realized the problem and decided to borrow ideas from other fields of engineering. This was the origin of software engineering (SE). Process models and methodologies for software development projects began to materialize leading to considerably improved projects. This solved some of its earlier problems, and software development grew to be a branch of engineering. The shift meant that project management and quality assurance problems were being solved. Additionally, it is helping to increase productivity and improve software

maintenance. The history of knowledge discovery in databases (KDD), Data Mining (DM), now known as Data Science, is not much different [21]. While it is true that the number of applied projects in the data-science is expanding rapidly, neither all the project results are in use nor do all projects end successfully. The failure rate is actually as high as 60% [22]. On the flip-side, standardized frameworks can have advantages all-round the value-chain.

The market, will benefit greatly from a common framework. The framework can serve as a common reference point to discuss data analytics and will increase the understanding of crucial data analytics issues by all participants, especially at the customers' side. But most importantly, it will help establish data-science as a standard engineering practice thus avoiding skepticism around adoption by mainstream businesses. Stakeholders can have more reasonable expectations as to how the project will proceed and what to expect at the end. It will be much easier to compare different tool-providers and consulting offers to pick the best. A common framework will also support the dissemination of knowledge and experience within the organization [19].

Analysts performing data science projects can also benefit in many ways. For novices, the framework provides guidance, helps to structure the project, and gives advice for each task of the process. Even experienced analysts can benefit from check lists for each task to make sure that nothing important has been forgotten. But the most important role of a common framework is for recording and sharing of results. It helps to link the different tools and different people with diverse skills and backgrounds together to form an efficient and effective practice.

The vendors will benefit from the increased comfort level of their customers. There is less need to educate customers about general issues of data-science. The focus shifts from whether data science should be used at all to how it can be used to solve the business questions. Vendors can also add values to their products, for instance offering guidance through the process or reuse of results and experiences. Service providers can train their personnel to a consistent level of expertise.

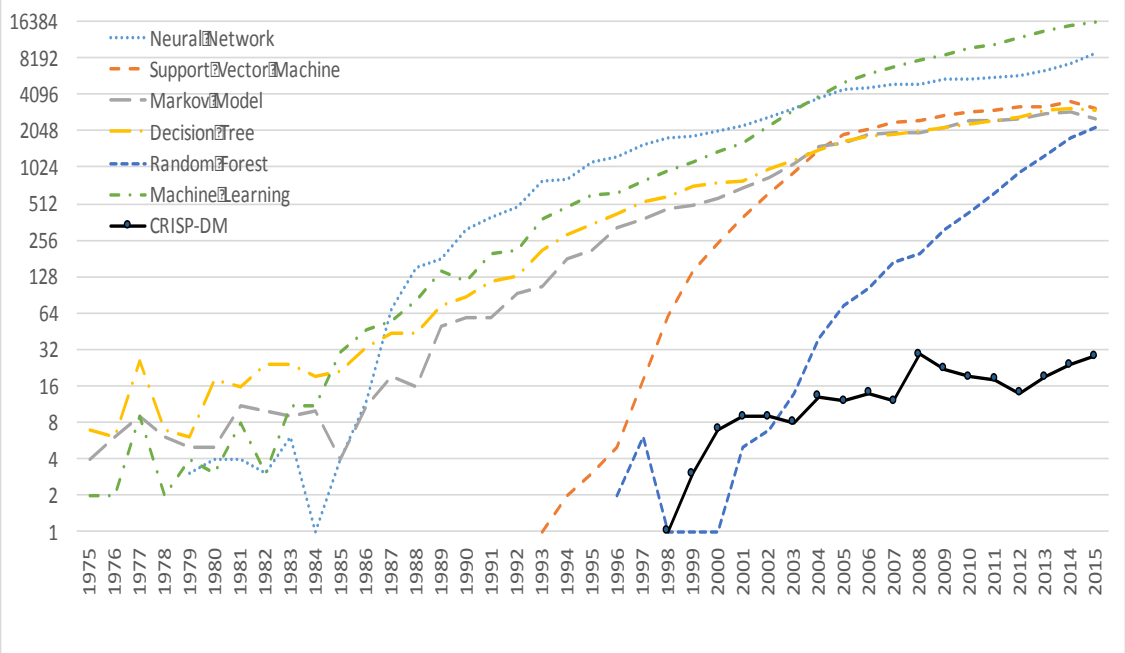


Figure 4. Research Articles for Common Algorithms and Analytics Process

Unfortunately, there is very little literature on the process of taking an algorithm from the lab-environment to production environment [23]. Figure 4 shows articles retrieved by year between 1975 and 2015 for some well-known machine learning algorithms and 'CRISP-DM' the widely accepted process for data-mining endeavors in the real-world [24]. It is clear that research on process is underserved by several orders of magnitude.

2.2 Classification of Frameworks

Several frameworks have been created since the dawn of analytics; many have found wide acceptance in the analytics community. There have been efforts to survey, compare and contrast frameworks, the best of which can be found in [25] and [21]. While Kurgan et al. [25], focus on data-mining process-models popularized in scientific publications that are subject to peer-review evaluation, Marbán et al. [21] pay attention to real-life applications in industry. The latest survey-centric publication by Mariscal et al. [26] is the most comprehensive covering all major processes including non-analytic methodologies like 6-sigma. Since the initial concept of an analytics framework in the early 1990s, several classifications have emerged.

Human-centric vs. Data-centric: The seminal work in [27] led to two major types of frameworks: human centric and data centric. The human-centric process was defined as a series of knowledge-intensive tasks consisting of complex interactions, protracted over time, between a human and a database. On the other hand, data-centric models are structured as sequences of steps that focus on performing manipulation and analysis of data and information surrounding the data, such as domain knowledge and extracted results [25]. For mainstream adoption of analytics, they need to be within reach of people not specialized in data-science which means simpler off-the-shelf tools that can be used across several application domains. With increasing focus on large scale undertakings with big-data there is a higher need for automation. There are exceptions to this trend in domains like medicine and mission-critical systems where due to regulatory or legal reasons, the final responsibility of the decision needs to lie with a human [28].

Industrial vs. Academic: The academic oriented methodologies focus on the central data-preparation and modeling steps and aim towards knowledge-extraction; whereas, the industrial methodologies focus on application starting with a business-goal and culminating in deployment. The industrial methodologies tend to include aspects of resourcing, program-management and integration.

Tool-agnostic vs. tool-specific: Some of the work in creating standard methodologies or framework is invariably driven by tool-vendors due to a strong business motivation for several reasons including establishing interoperability,

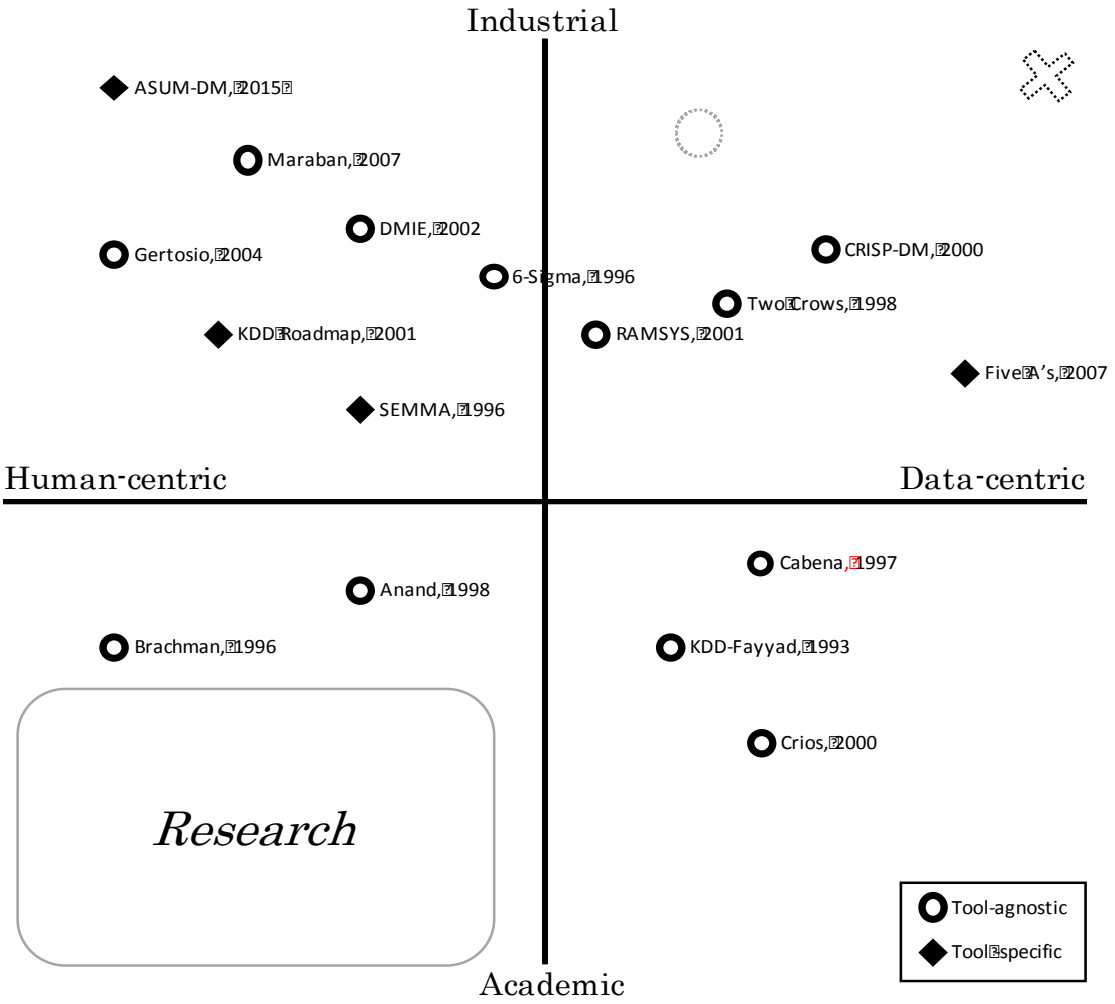


Figure 5. Major Analytics Methodologies by Classification

talent-pool, and, largely stickiness to the product or service offered. Whereas a big part of the analytic community favors tool-agnostic methodologies that can be adopted regardless of the tool-choice and adapted to their changing needs.

Figure 5 shows the methodologies and where they lie in quadrants defined by the classification mentioned above. One sees that there is a reasonable proliferation of methodologies albeit not as large as algorithms. The earlier methodologies are more research focused and as analytics became mainstream, more industry oriented methodologies have been developed. Additionally, the lower-half is sparser than the upper as the incentive to standardize methodology is not as high in academics. Particularly, the lower-left quadrant is the domain of research. The number of tool-agnostic methodologies exceeds the tool-agnostic. Surprisingly, the tool-specific methodologies are more human-centric than the tool-agnostic ones indicating the autonomous analytics is still a work in progress. The dotted 'X' indicates a theoretical goal of a data-centric highly autonomous self-learning system that needs minimum human supervision. The dotted 'O' shows where the recommendation of this research would map to.

2.3 Salient Contributions from Literature

The salient contributions in the evolution of frameworks are covered. Figure 6 shows the steps of each of the methodologies covered herein - they have either had a unique contribution or played a major part in forming the latest thought.

2.3.1 KDD, Fayyad et al. 1993

KDD is defined as the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data. The first reported KDD model consists of nine steps and was developed by *Fayyad et al.* in the mid-1990s [27] [29] [30]. KDD refers to the overall process whereas data mining step refers to the application of algorithms for extracting patterns from data. This methodology was the first to acknowledged that extracting knowledge from data involved more than the algorithm itself. It also includes the choice of encoding schemes, preprocessing, sampling and projections of the data before the data mining step. The KDD process is interactive and iterative with many decisions made by the user. Although the role of understanding the domain and data cleaning are highlighted, the roles of evaluation and deployment are not mentioned. The method is more academic-oriented with knowledge as result.

There have been many derivatives of the KDD process. *Cabena et al.*, defined the process of extracting previously unknown, valid, and actionable information from large databases to make crucial business decisions [31]. The key enhancement is an emphasis on *actionable* information geared towards business decision making. The human-centered model by *Brachman & Anand* on the other hand emphasized the interactive involvement of a data analyst during the process [32]. The human centric model was extended to industrial data by *Gertosio and Dussauchoy* [33].

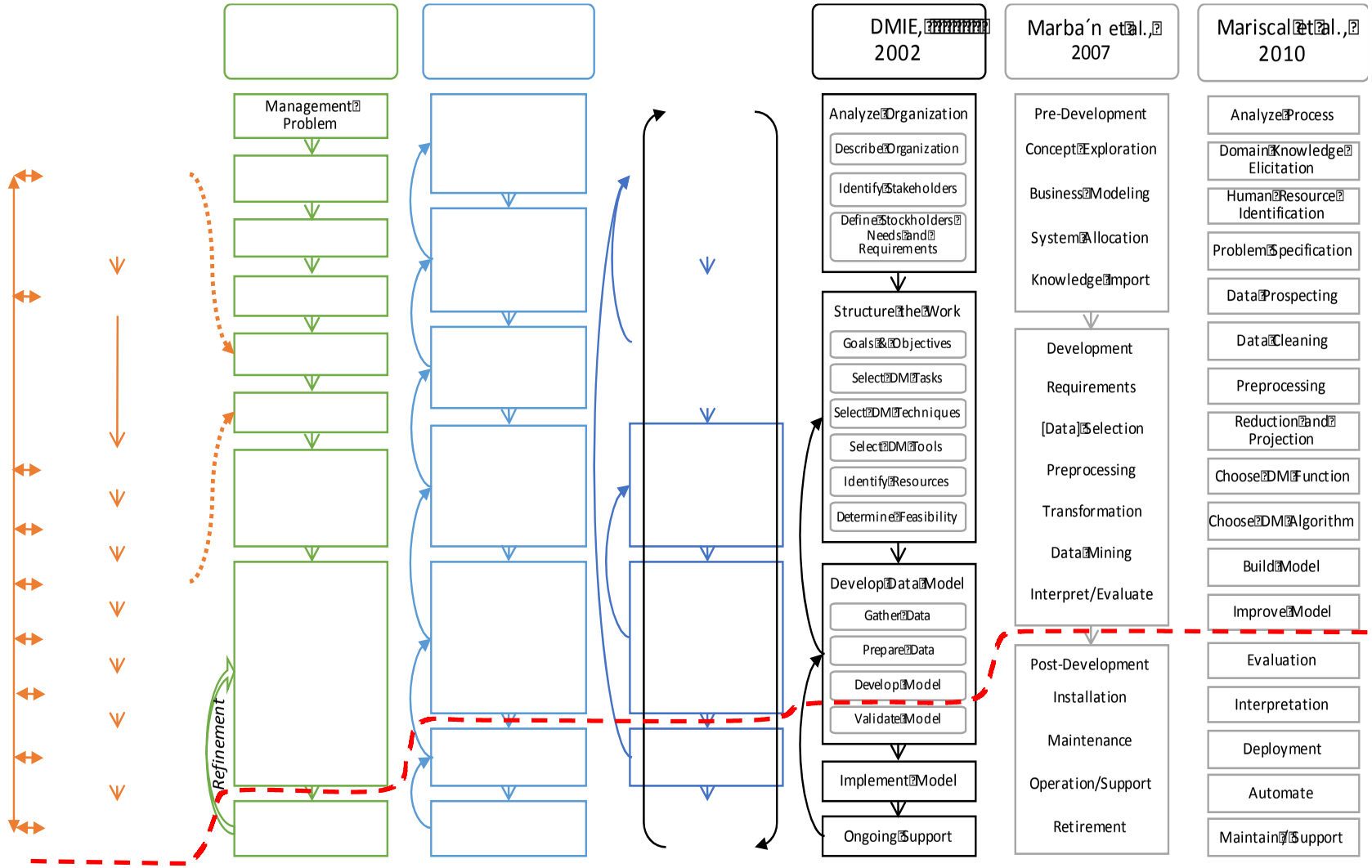


Figure 6 Comparing Steps of Major Methodologies Through Time

An example of tool-specific derivative is SEMMA (Sample, Explore, Modify, Model, and Assess). SAS Enterprise Miner commercial analytics software platform is built around it [34]. SEMMA pays no attention to the business problem or deployment [24]. SAS claims SEMMA is a logical organization of the functional tool set and not a methodology [35]. However, SEMMA has gained popularity recently because of the popularity of the tool and the tool-based step-by-step guidance offered to non-data-scientist practitioners of analytics.

2.3.2 Anand and Buchner, 1999

This model evolved through a couple of iterations before reaching its final stage as the internet/web-enabled knowledge discovery process [36] which consists of 9 steps as shown in Figure 6. The two major contributions of this model is the addition of the 'Management Problem' as the driver for data analytics undertakings and more importantly, the elaboration of the 'Knowledge Port-processing' step. In the elaboration, it is made clear that knowledge must be validated before it can be used [36]. Additionally, it is acknowledged that as the knowledge discovery process is *dynamic* and prone to updates, the discovered patterns have to be maintained [36]. Setting up a knowledge maintenance mechanism consists of re-applying the already set up process for the particular problem or using an incremental methodology that updates the knowledge as the data logged changes. It does not include the needed activities to use the discovered knowledge [26]. The mechanism to store and update the knowledge is not necessarily elaborated but the ideas is highly relevant to the dynamic environments of today. This contribution formed a key ingredient in the creation of the more widely accepted CRISP-DM methodology.

2.3.3 Two Crows, 1998

The Two Crows methodology was initially proposed by Two Crows Corporation in 1998 and then refined based on the (pre-release) CRISP-DM model [37]. This methodology recognized the iterative nature of the entire process as indicated by the return arrows in Figure 6. More notably, it proposed three new highly relevant steps for practical analytics well before its time – building and maintaining a mining database, exploring the data, model-evaluation and deploying the model. Following are details on these three steps. Note not all the steps are covered herein and can be found in [37].

2.3.3.1 Building a Data Mining Database

The data to be mined should be collected in a database. In general, it's not a good idea to reuse the corporate data warehouse. Instead, creating a separate data mart is recommended for the following reasons:

- Mining the data will involve highly active use of the data warehouse. It will often mean joining many tables together and accessing substantial portions of the warehouse. A single trial model may require many passes through much of the warehouse. Thus possibly causing resource allocation problems.
- It is often overlooked that mining almost entails modifying the data from the data warehouse. One may want to bring in data from outside the company to overlay on the data warehouse data or add new fields computed from existing fields. Additional data may be gathered through surveys. Other projects building different models from the data warehouse (some of whom will use the same data) may want to

make similar alterations to the warehouse. However, data warehouse administrators do not look kindly on having data changed in what is unquestionably a corporate resource.

- One more reason for a separate database is that the structure of the corporate data warehouse may not easily support the kinds of exploration needed to understand the data. This includes queries summarizing the data, multi-dimensional reports (sometimes called pivot tables), and many different kinds of graphs or visualizations.
- Lastly, performance, reliability and other considerations might necessitate that this data is stored in a data-store with different physical design.

Creating, loading and maintain the data-mining database is a multi-step process and will discussed later in 3.3.1.

2.3.3.2 Explore the Data

Before building good predictive models, one must understand the data. The goal is to identify the most important fields in predicting an outcome, and determine which derived values may be useful. Graphing and visualization tools are a vital aid in data preparation and their importance to effective data analysis cannot be overemphasized. Data visualization most often provides the *Aha!* leading to new insights and success [37]. In a data set with hundreds or even thousands of columns, exploring the data can be as time-consuming and labor-intensive as it is illuminating. A good interface and fast computer response are very important in this phase because the very nature of the exploration is changed when one has to wait even 20 minutes for some graphs, let alone a day [37].

2.3.3.3 Evaluation and Interpretation

The Two Crows methodology emphasizes the importance of model validation based on a few reasons. First, the accuracy rate found during testing (cross-validation) applies only to the data on which the model was built. In practice, the accuracy may vary if the data to which the model is applied differs in important and unknowable ways from the original data. More importantly, accuracy by itself is not necessarily the right metric for selecting the best model. One needs to carefully consider the type of errors and the costs associated with them. The confusion-matrix is recommended as means of estimating quality by combining the cost of different type of errors. The 'Lift' (gain) chart is another suggested tool that shows how responses are changed by application of the model versus the status-quo. The methodology presses for assessing the value of the model and admonishes that a pattern may be interesting, but acting on it may cost more than the revenue or savings it generates. Thus highlighting the importance of computing ROI (Return on Investment) to provide guidance to decision making by attaching values to the response and costs to the program. Lastly it is noted that there may be no practical means to take advantage of the knowledge gained [37].

As pointed out above, no matter how good the accuracy of a model is estimated to be, there is no guarantee that it reflects the real world. A valid model is not necessarily a correct model. One of the main reasons for this problem is that there are always assumptions implicit in the model. Also, the data used to build the model may fail to match the real world in some unknown way, leading to an incorrect model. Therefore, it is important to test a model in the real world by

running field experiments. The higher the risk associated with an incorrect model, the more important it is to construct an experiment to check the model results [37].

2.3.3.4 Deploy the model and results

Once a data mining model is built and validated, it can be used in one of two main ways. The first way is for an analyst to recommend actions based on simply viewing the model and its results. The second way is to apply the model to different data sets. The framework highlights some of the real-world aspects involved in deploying the model as part of an existing manufacturing or business process.

The data mining model is often applied to one event or transaction at a time. The amount of time to process each new transaction, and the rate at which new transactions arrive, will determine whether a parallelized algorithm is needed. While batch scoring might be sufficient for some processes, others might need near real-time scoring thus imposing limits on the response-time and hence the complexity of the model.

When delivering a complex application, data mining is often only a small, albeit critical, part of the final product. For example, knowledge discovered through data mining may be combined with the knowledge of domain experts and applied to data in the database and incoming transactions.

2.3.3.5 Model monitoring

Over time, all systems evolve. External variables such as inflation rate and geopolitical events may change enough to alter the way people behave. Shifts in environment or raw-materials could have an impact on manufacturing. Thus, from time to time the model will have to be retested, retrained and possibly completely

rebuilt. Charts of the residual differences between forecasted and observed values are an excellent way to monitor model results. Such charts are easy to use and understand, not computationally intensive, and could be built into the software that implements the model. Thus, the system could monitor itself.

Thus the Two Crows framework brought to light several challenges and considerations both before the model-building phase and more importantly in the post-model phase namely model evaluation, model deployment and lastly model monitoring and up-keep. However, some aspects are still lacking like the need for support in the post-model phase.

2.3.4 CRISP-DM, 2000

The CRISP-DM (CRoss-Industry Standard Process for DM) process model includes six steps was proposed in 1999 and published in 2000 by a consortium of four companies (Teradata, SPSS (ISL), Daimler- Chrysler and OHRA) that developed data mining projects. The model was released (version 1.0) in 2000 [29] [30] and it continues to enjoy a strong support among practitioners to this day [12].

It consists of a *Reference Model* and step-by-step *User-Guide* based on the experience and challenges they faced at the time. Whereas the *Reference Model* presents a quick overview of phases, tasks, and their outputs, describing *what to do* in a data mining project, the *User Guide* gives more details on each task within a phase and depicts *how to do* a data mining project [19]. The CRISP-DM model is designed to be a reference model and hence tries to maintain universality while ensuring room for adaptation to specific project needs. The methodology is described in terms of a hierarchical process model, consisting of sets of tasks described at four levels of abstraction, two of which are generic and the other two are specific.

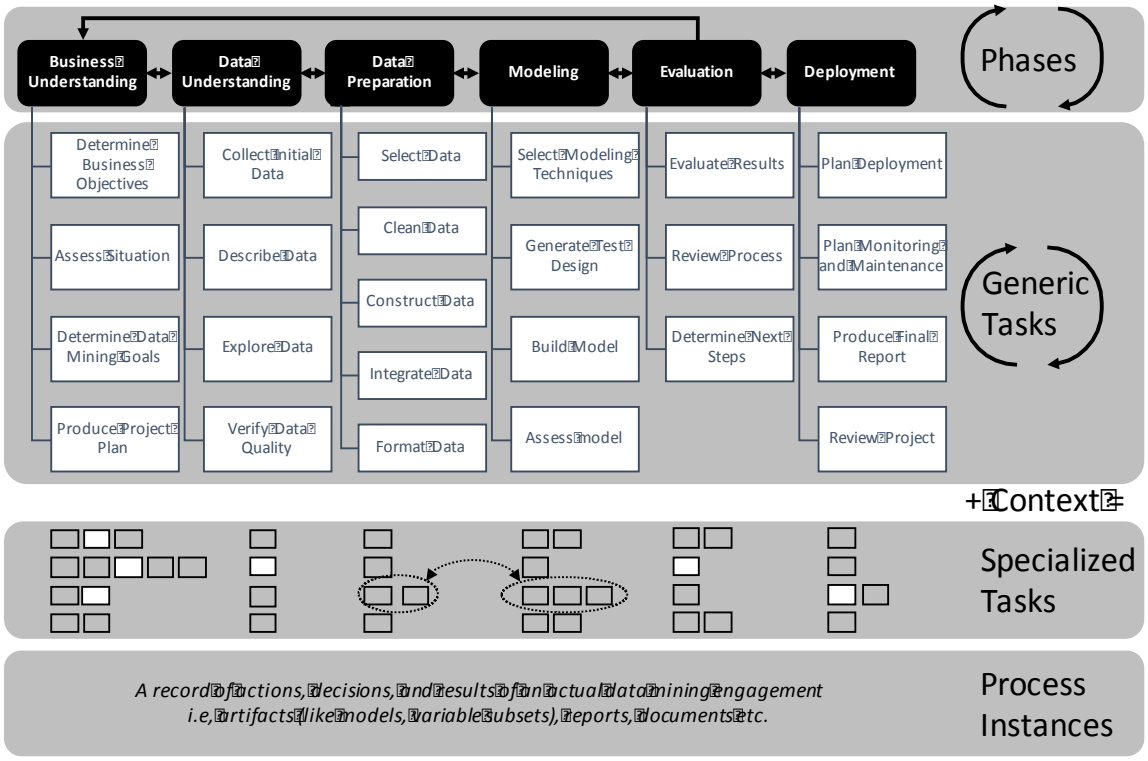


Figure 7. An Illustrative Presentation of the CRISP-DM Framework

The original documents describing the CRISP-DM framework illustrates the levels, phases and tasks using three separate diagrams. This has led to confusion in the interpretation of the framework and hence spawned apparent innovations that tend to reinvent the wheel. In Figure 7 an attempt is made to highlight the true contributions of the CRISP-DM model. The iterative nature of the phases, generic tasks, specialized tasks is highlighted. Whereas generic tasks tend to be universal, not all specialized tasks are relevant to every data-mining undertaking and thus one needs to apply contextual judgement to the relative importance of each of the specialized tasks. Lastly the process instances are a set of outputs of the data-mining undertaking that go beyond just the model or prediction [38].

The next sections elaborate these key contributions and identify areas where CRISP-DM needs to be extended to cover today's needs.

2.3.4.1 Data-Mining as a Continuous Iterative Process

As shown in Figure 7, the first level consists of six general *phases* that are meant to encapsulate the high-level progression of most data-mining engagements. The description of phases and tasks as discrete steps performed in a specific order represents is an idealized sequence of events. In practice, many of the tasks can be performed in a different order and it will often be necessary to backtrack to previous tasks and repeat certain actions. Despite the document stating in several places that the phases and tasks are not supposed to be strictly sequential, the presentation of the process-model phases inevitably creates the sequential impression amongst decision makers. Hence the project may be subject to very tight deadlines, which in the end leads to sub-optimal solutions [39]. To counter this gap, spiral versions of the CRISP-DM model have been recommended like the *'snail shell'* model in [40]. However, the iterative structure of CRISP-DM is already a fertile ground for application of Agile development methodology [41]. Thus one of the key contribution of the CRISP-DM is that it framed analytics as a long-term undertaking iterating through phases and possibly solving different business problems at each iteration.

2.3.4.2 Universality of Core Data-Mining Tasks

The second level consists of *generic tasks* which serve as a checklist to ensure all aspects of the analytics project are being considered. The generic tasks were intended to be as complete and stable as possible. Complete to cover both the entire data mining process as well as data mining applications. Stable to cover yet unforeseen developments like new modeling techniques. While this was a good goal, it is not practical to cover unforeseen developments and often results in dilution. Another major challenge is to put a project management framework around this

highly iterative, creative process with many parallel activities [19]. Thus the definition of completion of a task usually has a fuzzy answer.

2.3.4.3 Customizing the Framework to Context

The third, specialized task level, is the place to describe how actions in the generic tasks should be carried out in specific situations. Whereas generic tasks tend to be universal, not all specialized tasks are relevant to every data-mining undertaking and thus one needs to apply contextual judgement to the relative importance of each of the specialized tasks. Applying the context might also result in splitting one abstract task into several more concrete tasks [19]. Addition of tasks could also occur for example localization is not a task mentioned during deployment but might be highly relevant for global contexts. Thus CRISP-DM allowed the customization of the tasks to the particular context of application via a seed set of specialized task that could then be extended as needed.

2.3.4.4 More than a Model

The fourth, the process instance level, is a record of actions, decisions, and results of an actual data mining engagement. A process instance is organized according to the tasks defined at the higher levels, but represents what actually happened in a particular engagement, rather than what happens in general. The process instances are a set of outputs of the data-mining undertaking that go beyond just the model or prediction. Thus recognizing that the output of a data-mining undertaking goes well beyond the model and could give insights into varying aspects of the organization, industry or business. Moreover, capturing these artifacts helps continue the analytics process instead of an abrupt ending.

The CRISP-DM methodology phases and generic tasks are as below [18]:

1. Business understanding involves grasping objectives and requirements from a business perspective and converting these into a DM problem definition resulting in a preliminary project plan to achieve the objectives.
2. Data understanding starts with initial data collection and familiarization with the data followed by identification of data quality problems, initial insights into the data, and detection of interesting data subsets.
3. Data preparation covers all activities needed to construct the final dataset, which constitutes the data that will be fed into DM tool(s) in the next step. It includes Table, record, and attribute selection; data cleaning; construction of new attributes; and data transformation.
4. Modeling usually involves the trial and selection from several modeling methods to find a good fit followed by the calibration of their parameters to optimal values. The models are then assessed for goodness of fit. Since some methods may require a specific format for input data, often reiteration into the previous step is necessary.
5. Evaluation is where the model is evaluated from a business objective perspective to determine whether all important business issues have been sufficiently considered. At the end of this phase, a decision about the use of the data-analytics results should be reached.
6. Deployment can range from being as simple as generating a report or as complex as integrating the model into the current system or business process. The latter would involve creating a plan to monitor and maintain the system. A final review may be performed to capture key learning.

The CRISP-DM methodology is far from perfect – for example, the user-guide states that *"it will be the customer, not the data analyst, who will carry out the deployment steps."* This is far from the truth in most real-world experiences. However, CRISP-DM is a fine foundation on which to build and extend frameworks and hence has spawned several offshoots. For example, Cios et al. [42] extended the methodology by adding research-oriented description of the steps, introduction of explicit feedback mechanisms and a modification of the last step to emphasize that knowledge discovered for a particular domain may be applied in other domains. RAMSYS (RApid collaborative data Mining SYStem) [43] extends CRISP-DM where some of the generic tasks can be carried out in a remote collaborative mode. The current best understanding of the problem (set-of-hypothesis) is kept in the information vault, where information is shared between the different groups more akin to a GIT-repository for open-source software.

The other three CRISP-DM offshoot methodologies namely DMIE [44], Marbán et al. [22] [45] and Mariscal et al. [26] will be covered in more detail.

2.3.5 DMIE (Solarte), 2002

The DMIE methodology [44] is divided into five major phases as shown in Figure 6. Although it is specifically designed for data-mining projects with applications in industrial engineering, some of the steps are highly relevant in any industrial application of data-mining. A field application of predictive analytics almost always replaces an incumbent process. Hence there is natural organizational inertia from the owners of the incumbent process. Thus the 1st phase of 'Analyze Organization' becomes highly relevant. It is best to have a strategy to include the incumbents beforehand else this could easily result in failure of the program.

2.3.6 Marbán et al., 2007

Marbán et al.'s approach is based on the idea that data mining problems are taking on the dimensions of an engineering problem. Therefore, the processes to be applied should include all the activities and tasks required in an engineering process. This methodology enhances CRISP- DM by embedding other current standards, as suggested in [25], inspired by software engineering standards and practices. Figure 6 shows the general scheme of Marbán et al.'s process model as compared to CRISP-DM. The framework has three phases: pre-development, development and post-development.

The core development processes are inherited from CRISP-DM / KDD and the rest of management and development processes are based on two software engineering standard process models: IEEE 1074 and ISO 12207. This framework explicitly recognizes that productizing analytics in the real-world is inherently a large undertaking and the core development from data-selection to modeling constitutes less than a third of the total work involved. This is important to note because many of the analytics initiatives tend to heavily underestimate the time, resources and effort needed to achieve success. The same author has hence proposed a cost model to estimate the effort of data mining projects named DMCoMo [46].

In addition to calling out business understanding in pre-development the framework also has a placeholder for system allocation as was highlighted by the Two Crows as well as Anand & Buchner. Lastly, the post-development section gives the deserved importance to installation, maintenance, operations, support and retirement. The retirement phase must not be trivialized as resources can be held up if installations are not retired when no longer relevant.

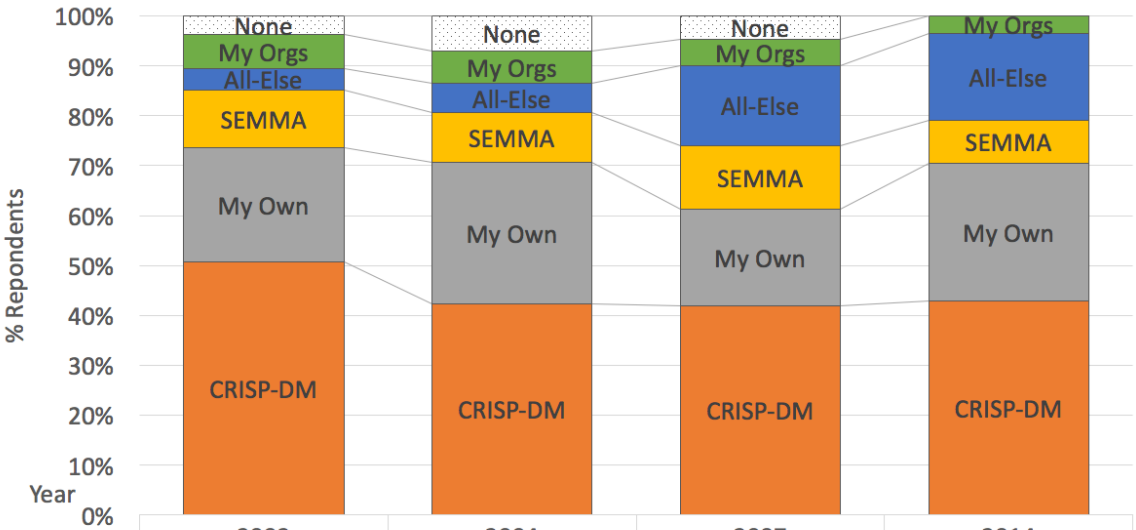
2.3.7 Mariscal et al., 2010

Mariscal et al. have published a great survey of many frameworks that have been proposed [26] that also include frameworks from other areas like Six-Sigma DMAIC (Define-Measure-Analyze-Improve-Control) framework. They then pick out salient steps from each of the frameworks and arrange in a logical order to propose a list of steps called *Refined Data Mining Process*. The proposed process has 3 main phases like in Marbán et al. but has 17 sub-steps. However, in doing so, the one-thirds importance given to the pre-development, development and post-development is diluted.

There are several other process models that made a less significant impact and thus are not discussed in in this survey [26]. A six-step KDDM process model by Adriaans & Zantinge; a four-step model by Berry & Linoff; a seven-step model by Han & Kamber; a five-step model by Edelstein; a seven-step model by Klosgen & Zytkow; seven-step model by Haglin et al.

2.3.8 Frameworks Summary

KDD and CRISP-DM have been two key turning-points in the history of data-analytics frameworks each spawning a multitude of frameworks that have adapted to changing analytics, domain and business needs. Over the years' emphasis has shifted from the core tasks of data cleaning and algorithm application to the pre-development activities of planning, stakeholder identification, resource allocation and business understanding. Only in the last five years' focus has shifted to the post-development activities of evaluating results against business goals, deployment, automation, maintenance, support and retirement.



	2002	2004	2007	2014
None	3.7%	7.1%	4.7%	0.0%
My Orgs	6.9%	6.5%	5.3%	3.5%
All-Else	4.2%	5.9%	16.0%	17.5%
SEMMA	11.6%	10.0%	12.7%	8.5%
My Own	22.8%	28.2%	19.3%	27.5%
CRISP-DM	50.8%	42.4%	42.0%	43.0%

Figure 8. Popularity of Data Analytics Methodologies in Recent Years

Although numerous frameworks have been proposed, the CRISP-DM tool-agnostic methodology holds the majority of the mind-share as determined by the popular data-analytics community site kdnuggets.com over several years since introduction [24]. The survey results are shown in Figure 8. Thus it is clear that augmenting one major methodology and campaigning for its wider acceptance is a better strategy to establishing a common standard than proposing yet another methodology that vies for the mindshare of the data-analytics community.

Thus the proposal in this research uses the CRISP-DM methodology and makes recommendations to augment and improve it in order to make it relevant to today's problems and challenges. Next the key shortcomings are covered.

2.4 Shortcomings of Current Frameworks

As DM & KD development projects became more complex, a number of problems emerged: continuous project planning delays, low productivity and failure to meet user expectations. Neither all the data analytics project results are useful, nor do all projects end successfully. Today's failure rate is well over 50% [21].

Many of the frameworks were proposed during an era where data-analytics and machine-learning were still dominated by research and had not become part of the mainstream. Hence the focus was on creating/matching the algorithm to the data and fine-tuning predictive performance – the methodology sufficed for the needs of those times. However, now with packages available in the programming language of choice [47], and vast computing resources at our disposal, the matching and tuning steps are not necessarily the most challenging and can be automated.

Another trend is of recent times have been the use of machine learning in online semi-mission-critical applications. Applications range from curating the news to self-driven automobiles. When CRISP-DM was envisioned, mission-critical applications were simply left out of scope. The Agile Manifesto came into being right around the time if CRISP-DM hence, although the methodology recognizes the iterative nature, it does not have elements of fast build-test-build cycles. The most used analytic process models at the moment (CRISP-DM and SEMMA [24]) have not yet been sized for real-world tasks. Concepts of security, privacy, reliability and maintainability are just being explored [48]. Although the interest and attention given to the evaluation and deployment steps has been increasing as seen by the dashed line in Figure 6, incorporating validation in every step of the process has not been considered.

2.5 Proposed Framework

Industrial systems and most real-world systems put a high premium on reliability. It is one of the biggest deterrents to widespread adoption of machine-learning systems. The second highest being maintainability. Reliability and maintainability however are not new needs; they have always been around in most engineering disciplines. Especially in mission-critical and 24/7 applications, the system engineering approach has been adopted to ensure designs are systematically validated for performance, quality and value. Concepts of design for test (DFT) and design for manufacturing have been thoroughly studied and widely practiced. The V-model for system development is quite popular in the defense industry and used by governments of various nations. The V-model has also been adopted for software development especially for mission critical systems.

This work takes the most effective aspects of data-analytics methodologies till date and combines them with system-engineering methodology to recommend a framework for data-analytics. The *Design for Deployment (DFD)* framework is shown in Figure 9. The arrows going from every step back to the previous highlight the highly iterative nature of the process in general. The phases on the left have double-arrows linking them to the validation phases on the right indicating tight a coupling between the corresponding phases on the two branches of the 'V'. The core idea being that the validation criteria or step be thought of before beginning on the corresponding understanding or building step so as to *'begin with the end in mind.'* [49]. In the reminder of this section the phases are explained one pair at a time to keep with the tight coupling mentioned before.

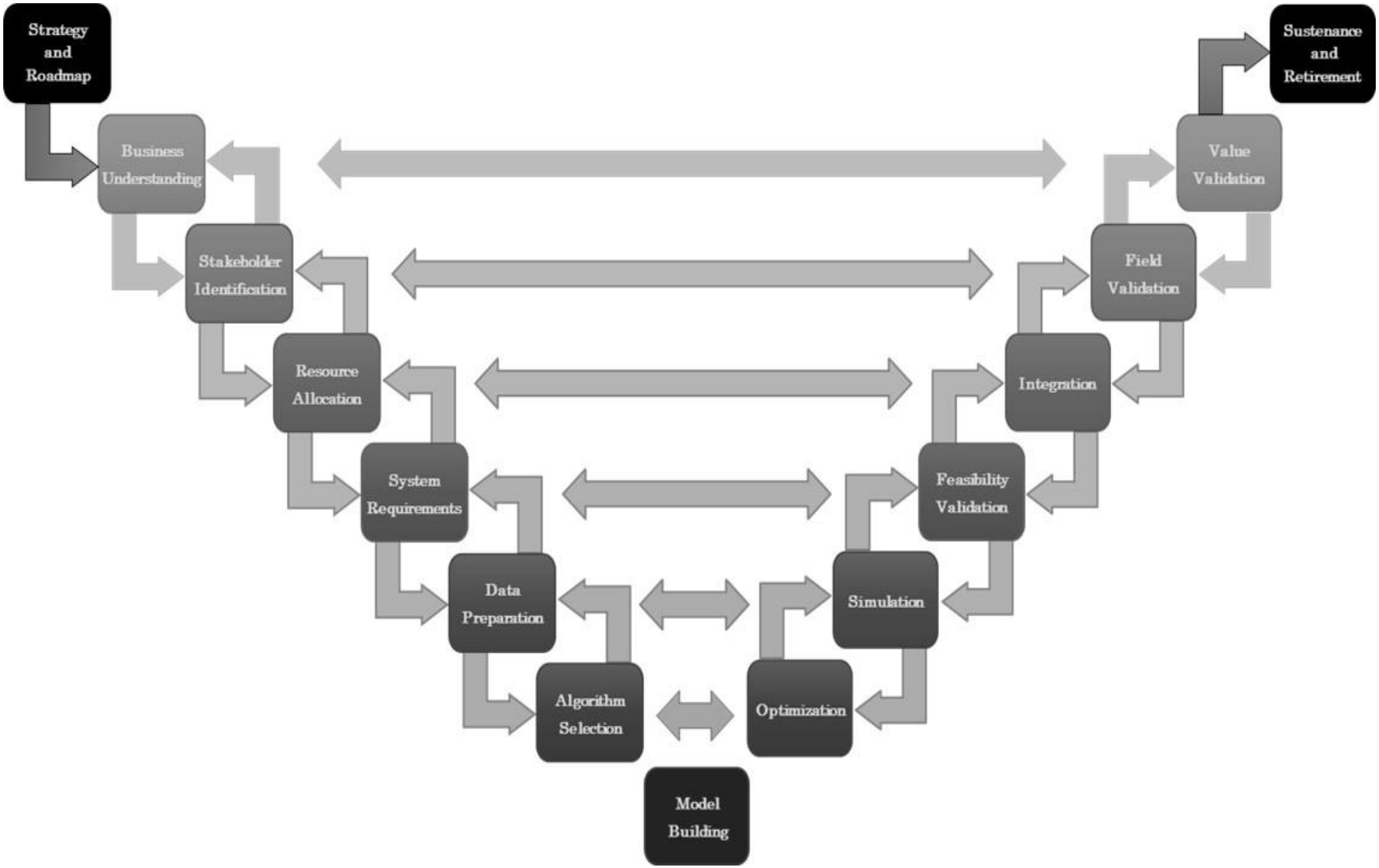


Figure 9. Design for Deployment (DFD) Framework for Deployable Analytics

2.5.1 Strategy and Sustenance

These two phases are more part of the business and operations than analytics per se. That said, they have a big impact on the success of analytics deployment. The strategy and roadmap of the organization determine how critical analytics is to the core business sustenance and thus will directly correlate to the investment and thrust put into making the analytics deployment a success. Often the analogy of a pain-killer versus vitamin is used in tech-startup circles. One cannot sustain much longer without immediate pain-relief; however, vitamins although important for long-term health, do not have the same sense of immediacy around them. The closer the analytics is to addressing an immediate and persistent need, the higher the likelihood of success. If the analytics is considered a one-time self-contained (siloed) project, it is unlikely to bring that much value to the organization except to check a "me-too" box.

Deployment of analytics is a business and cultural transformation and needs longer term commitment. Sustenance requires involvement of multiple stakeholders and a well-trained operational team that can upkeep the analytics system for day-to-day operations as well as make improvements over time and as technology matures. Several questions arise: What are the levers and knobs desired? What are the indicators that will be monitored? How will the system maintenance and upkeep be carried out? How long will the system be in production before being retired?

The three key actions here are establishing design-patterns, putting infrastructure in place for autonomous model deterioration detection and update/refresh as well as establishing the analytics function/department. These will be elaborated in Section 4.3.4.

2.5.2 Business Understanding and Value Validation

Industrial data-analytics is motivated by the desire to scale or automate the discovery process so it can ultimately be used to offer new products and services, expand the customer base, generate new revenue or lines of business. The initial phase is where expectations are set (or unfortunately assumed). CRISP-DM says, this initial phase focuses on understanding the project objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

CRISP-DM seems to imply that the data-scientist drives this process and the business knows exactly what they want. Often this is not the case – the business has a vague idea of 'leveraging data to gain new-insights'. They do not necessarily appreciate the complexity of models, needs for resources and data or expectations or limitations of the models. Additionally, the group that commissions the analytics projects is usually the leadership and the group that is responsible to provide requirements is the incumbent team of analysts (or database team) who might not be clear about the objectives themselves. Once the model is built, the incumbent-team is again responsible to interpret and summarize to leadership. The mismatch of expectations can lead to utter dissatisfaction at deployment. Another realization that is missed is that deployment of analytics also means that the business itself needs to transform in order to work with analytical models.

Due to the reasons mentioned above, the best strategy would be to envision the value validation phase and agree on how the value provided by the final deployed system would be ascertained. It involves education of the business parties involved on what analytical modeling is, what realistic expectations are from the

various approaches and how models can be leveraged in the business. Discussions on the business case identification involve topics around data availability, model integration complexity, analytical model complexity and model impact on the business [50]. Some topics to cover would be: What is the key business problem that the system will solve or significant opportunity it will help gain? What will success look like and who will need to sign-off? What is the projected ROI of the system? How reliable will the system expected to be? How much improvement over the incumbent system would be needed to justify a change? What changes would be needed to the business process? This is by no means an exhaustive list. From a list of identified cases in an area, the one with the best ranking on above mentioned criteria should be considered for implementation. The others are put on the aforementioned roadmap.

The core elements to evaluate are relative performance and reliability (compared to incumbent system) and return-on-investment (ROI). The three topics are covered in Section 4.3.3. By thinking of the value validation, one is forced to think in terms of ground realities and the discussion invariably leads to how things work currently and who owns those components. Stakeholder identification is next.

2.5.3 Stakeholder Identification and Field Validation

Leadership is driven by ROI and once leadership buys in, the rest of the company has to. However, in reality, there is still a lot of inertia to continue with the status quo. We must be clever in how we embed analytics into the business model and force analytics into the existing processes, involving stakeholders as early in the process as possible to garner their buy-in [51]. Several stakeholders have a role to play in each phase of the analytics process: (higher) management, to ensure the

right goal setting, IT for data availability, integration and deployment, the concerned receiving department for model relevance checking, data scientists building the model and the end users.

Creating a stakeholder map early in the process ensures that inputs from those sectors are not missed and surprises are avoided later in the process. It also drives involvement from the entire organization thus using the IKEA effect (where one places higher value on something one had a part in building) in favor of the machine-learning analytics system instead of against. The map helps identify who owns what area and thus controls what decision and/or resource thus time is not spent in tracking this down when the need is urgent. The map also helps identify the paths of influence which can be highly valuable to obtain the go decision for deployment. Lastly it provides a starting point to train the organization to better understand, interpret and work with machine-learning analytics. Identifying the stakeholders and their requirements will allow analysts to completely recognize the critical elements of the project, including true intentions and expected results [44].

By making note of all the pitfalls and nuances, a validation plan starts to emerge on whether the system is ready to be launched. The elements from this validation plan are used later to conduct the field validation once integration is completed. Field validation is where the machine-learning analytics system is run as if it were deployed but without necessarily exposing the results to end customers (or doing so in a limited fashion). The run is done in parallel with the incumbent system to evaluate the chosen metrics. The objective of the evaluation is more to determine any unexpected behavior as well as shortlist the indicators based on their effectiveness that will be used to monitor the system in production.

2.5.4 Resource Allocation and Integration

Integration of a newly developed capability into the existing process and system requires two types of resources: human (team) and system. Thus when thinking from integration point of view, one can easily identify both system as well as the team resources needed. To derive the teams and expertise needed, one can simply start from the stakeholder map developed in the previous phase and then ask for representatives and/or volunteers from each of the major stakeholder groups to be part of the effort. While at first this might seem like a huge overhead to manage a matrixed team, the advantage in terms of diversity of thought and strong links to the incumbent departments ensures that the IKEA effect works in favor of the analytics deployment than against it. For the volunteers to be effective contributors, some (or most) could need training on interpreting machine-learning results. The recent hype around data-science should help with having people willingly invest the time into learning. The instruction and guidance however, should be led by the analytics team. Knowledge-sharing is a key component in this approach and must be addressed proactively through training. More on the three-pronged approach (Involvement of teams, Interpretability and Insights) for enabling the incumbent team is covered in Section 4.3.2.

System resource allocation tends to follow once the right individuals are on board as they bring along spares from their respective departments to aid the effort. However, a formal assessment of resource requirements must be done and presented to the budgeting group to ensure that the resources can be procured well before they are needed. Additionally, resources like system-nodes need a considerable lead-time for procurement and setup and hence should be planned well in advance.

2.5.5 System Requirements and Feasibility Validation

In the system specification phase, the set of conditions are explored that need to hold for the model to be permissible and viable in the business process. Topics of discussion include constraints and boundary condition handling. The constraints are the non-negotiables like security, privacy and any legal considerations. The boundary conditions are compatibility conditions: side conditions that need to hold, consistency checks that need to hold, handling of unexpected predictions, or unexpected input data, requirements about the availability of the scores, the timing of scores (and the data) and the frequency of refresh of the scores. Initial ideas around model reporting can be explored and finally, ways that the end users would like to consume the results of the analytical models [50]. The output of this phase is ideally a feasibility validation plan; however, q requirements document would suffice. The system is then architected to meet the requirements and/or pass the feasibility validation plan.

One can identify the reviews and gates that any new system or installation needs to go through before entering the current production environment. The apt analogy is that of an organ transplant – without proper precautions, the likelihood of rejection is high. Thus like the medical-team, the analytics team should consider and employ all avenues of reducing the likelihood of rejection. Feasibility validation would then be the execution of plan that came out of the system requirements exercise. The key areas to be considered are outlined in Section 4.3.1. Feasibility validation cover the necessary conditions for deployment. Whereas, the subsequent phases reviewed before cover the sufficiency. Although the process so far might seem validation-heavy the rigor is required for deployable systems in industry.

2.5.6 Data Preparation and Simulation

In the data preparation phase, the discussions revolve around data access, data location, data understanding, data validation, and creation of the modeling data. This is a phase the data scientist will need help from subject matter experts, IT/data administrators/DBA's to closely work together to prepare the data in a format consumable by the data scientist [52]. Preparing the data involves first understanding it: how is it stored, where does it originate, how frequently is it updated, how often do updates fail; what are the encodings, the relationships and data-dependencies; which are the raw versus computed values, what do we know about the process that collected the data, what was the measurement algorithm, and many more.

Some of this discovery can happen by careful planning but some of it happens as the modeling proceeds. The process is often highly iterative: the data scientist tries out various approaches on smaller sets and then may ask IT to perform the required transformation in large. Understanding of the operational data required for modeling and scoring, both from an availability (cost) and timing perspective are the goal. Section 3.2 describes the real-world challenges and Section 3.2.4 makes a case for building a simulation infrastructure that includes a data-mart. The recommended simulation infrastructure allows one to mimic the realities of the production scenario where data is often shifted in time and includes dynamics where the relationship between the features and predicted variables can change in more ways than one. Thus performance evaluations become representative of the real-world and hence more credible. The infrastructure allows heavy experimentation without disturbing the production system or staff.

2.5.7 Algorithm Selection, Model building and Optimization

The goal of the of algorithm selection phase it to find the model that best suits the data for the goals identified. In the model-building phase, an analytical model is built and evaluated against available (historic) data for good fit. Then in the optimization phase, the model is tuned to suit the characteristics of the data, system constraints and business priorities. As one can tell the three phases are highly iterative and co-dependent. Several restarts and trips to the proverbial drawing-board are to be expected. Hence it is important to communicate the reality clearly: it is scientific research, with all its struggles and need for meticulous experimentation. Results are not guaranteed, they depend on the quality of the data and the (unobservable) knowledge that the data contains about the phenomenon to be modelled, the time spent on the creation of the solution, the current state of the art of analytical models as well as the quality of the data scientist.

The data scientist may need to connect to end user to validate initial results, or to have discussion to get ideas which can be translated into testable hypotheses/model features. The best strategy is to have an infrastructure in place and design experiments that facilitate systematic yet rapid learning. In section 3.2.4, a simulation infrastructure is recommended that would have the data and looping abilities in place so the experimenter can simply setup the design and the data is ready when the simulation is done. Additionally, the ability to introduce temporal shifts and time-walking ensure the data obtained is realistic and representative of the real-world challenges. Design of experiment techniques are discussed in Section 3.3.3 so learning happens in the most efficient means possible.

Next the said real-world challenges are elaborated and the evaluation infrastructure is proposed.

2.6 Demonstration

In this section the proposed Design for Deployment (DoD) analytics framework is demonstrated in the area of semiconductor manufacturing. Semiconductor manufacturing is a multi-step process which takes several weeks from sand to silicon to shippable unit – see Figure 3. As one proceeds further in that flow, material, time and effort are added to each part and thus the sunk cost into that unit keeps rising. Assembly and final-tests account for roughly 50% of the total material and manufacturing cost [13].

2.6.1 Strategy and Roadmap

There is great value in being able to predict the behavior of packaged units early on in the manufacturing flow based on in-line measurements or Sort-Test results [53]. There was a long list of possible applications that were brainstormed. The salient ones are listed below:

1. Tool Anomaly Detection: Use machine learning to back-trace the anomalous tool that might have caused the defective pattern [54].
2. Statistical Process Control: Use SPC on the predicted values made of virtual fab-lots to detect drifts in the fab-process [55], [56].
3. Wafer GFA Diagnostics: Detect and classify patterns in gross failure analysis at a wafer level [57], [58].
4. Test-time Reduction: Use predictive analytics to intelligently alter test-flow at unit-level to reduce overall test-cost.

5. Selective Sampling: ML models are used to determine a risk level for each unit and the system-test sampling-rate is modulated accordingly.
6. Statistical Bin Limit (SBL) Dispositioning: Historical baseline data and current lot data are used for lots that triggered SBLs to determine the signal was caused due to sort/test process issues or upstream processes.
7. Inventory Management: Predicted data such as class test bin is used to group material for assembly/die prep kitting &/or to expedite in-line assembly lots to meet factory out for specific bins/SKUs.
8. Tool downtime scheduling: Manufacturing data is input to ML models that recommend optimal scheduled tool down time, product conversion, & material kitting schedules.
9. Selective die-kill: Identify die with a high propensity of failing after going through assembly and scrap them beforehand to save cost.
10. Die-matching: Match the speed of die to be paired on the same package.

As each of these has potential of saving millions of dollars, it resulted in a healthy roadmap and hence piqued interest from leadership to explore the possibilities.

2.6.2 Business Case

Based on preliminary analysis of the data, the selective die-kill project was chosen to explore first. Semiconductor manufacturing is a multi-step process which takes several weeks from sand to silicon to shippable unit. As one proceeds further in that flow, material, time and effort are added to each part and thus the sunk cost into that unit keeps rising. Assembly and final-tests account for roughly 50% of the total material and manufacturing cost [13]. Not all units pass the final tests, the

failing units are scrapped after failure validation and analysis – this is termed yield-loss. Clearly, yield loss means financial loss – due to sunk cost of manufacturing as well as opportunity-cost of a unit that could be sold for profit [15]. However, if one could detect a die that has the propensity of failing towards the end of the manufacturing flow, the sunk-cost could be saved. Depending on the accuracy of detection, the cost-savings could run into millions of dollars.

The criteria for selection was driven more by the fact that the predicted variable was binary and thus the classifier did not have to deal with spatial or multi-class data as for the other candidates. In retrospect however, the venture would have greatly benefitted from exercising patience to determine the ROI before proceeding. The main reason for not using the potential ROI was that financial cost and benefit information was not easy to come by. Firstly, the information was not centralized and spread across various geographically disperse groups. It existed in multiple currencies due to global operations. Tax and other financial considerations were at play like proportion attributed to R&D expense versus cost-of-sold-goods (COGS). Additionally, unit level financial data is quite revealing especially in manufacturing and hence guarded closely. Justifying and obtaining approvals to get hold off the data can take a while. This is where identifying the right stakeholders and making them part of the effort has huge impact on velocity.

2.6.3 Stakeholder Identification

Creating a stakeholder map from the deployment perspective immensely helped understand how the decisions were made currently. It also piqued curiosity in the representatives interviewed and they ultimately wanted to join the effort. They each brought their skills and experience to the table from data-extraction and

scripting to factory planning and financial analysis. The mapping exercise itself was not trivial and needed full-time focus. However, it did lead the project to the golden ratio that determined the ROI – it was > 2:1, true-positive to false-positive. That is the instantaneous slope of the ROC curve (see Section 3.1.2) had to be greater than 2 for the financials to make sense.

2.6.4 Resource Allocation

The core needs of the project were well supported: data scientists, experimental computing resources although sound justification was needed. However, whenever the needs intersected with production flow or existing infrastructure like databases, script-hosts, factory resources, the direction from leadership insisted on forming partnerships with the existing groups instead of designated personnel. This turned out to be a blessing in disguise as it helped the team learn about the various processes involved. However, it did contribute to slowing the velocity of progress as each new group that got involved needed training on machine-learning. Thus the project leads, authored internal coursework on machine-learning using generic as well as specific examples [59]. The training sessions were immensely helpful for the participants as well as the facilitators as they sparked discussions and ideas that led to many breakthroughs in the project.

2.6.5 System Requirements

There are strict demands on uptime in semiconductor manufacturing. Any system either slowing down the manufacturing flow or that cannot maintain 24/7/365 uptime is disallowed in factory systems. Designing a 24/7 model-building and scoring system would be impractical due to the data-availability constraints,

model build-time, as well as the uncertainty around being able to obtain a high-confidence model every-time. Hence the challenge was being able to design an architecture where an off-line (non 24/7 SLA) system would interface with and provide decisions to an online (24/7 SLA) system. Shown in Figure 10. "Clutch" Architecture to Meet System Requirements. There were 5 overarching system integration requirements that had to be resolved in the design prior to receiving stakeholder ratification of the system.

1. *Off-line Dependencies*: The high level architecture was designed such that tool processing and the on-line/mission critical systems are not impacted when the off-line portions of the system are not available. When prediction data is not available due to anomalies such as network connectivity, off-line system downtime, data integrity, etc. the process module will revert to legacy/non-ML optimized, processing conditions.
2. *Triggering:* The system monitors lot "move-in" and "move-out" transactions that are recorded in the manufacturing database. At a configured frequency *(hourly, daily, etc.)* the ML system queries the manufacturing DB and initiates predictions for lots that have "moved-out". The predictions are executed in background and the completed results are loaded into the ULT database before the lot reaches the specific process module several hours later where the decision can be implemented.
3. *Quality assurance and fail-over*: Quality assurance for model performance was another requirement for the system. During the model generation process training data is divided into distinct observed data (OD) and unobserved data (UD) separated in time. The OD data is used to build the

model and the UD set is used to generate predictions. These are compared to the UD actuals to compute prediction accuracy. The accuracy is checked against a configurable limit and only models within spec are released to production. The response for failing models is configurable to allow the system to continue to use the previous valid model or disable predictions until a valid model is generated. Additionally, there are several anomaly &/or exception conditions that can disable predictions for a given lot. Whenever a lot has been disabled, no data for that lot will be loaded to the on-line systems and only lot level data will be loaded to the off-line systems.

4. *Validation:* As the requirements were defined they were tested with respect to potential future prediction use cases. This was done to ensure that a broad set of system configurations, data types, exclusion conditions, etc. are readily available and scalable to future applications.
5. *Model/Data Aging:* On-line data volume and retention timeframe concerns needed to be addressed in the system design. Expirations dates can be defined for the predictions, models, and training-data since it was generated. The expiration windows for these parameters are configurable and can be set by the system operator. When an expiration date for any of these parameters is reached the ULT database is able to purge the data.

Other specific requirements for general manufacturing systems, features such as recipe management, off-line analysis, reporting, etc. were extensively captured it the system requirements document and have been left out of this document for brevity.

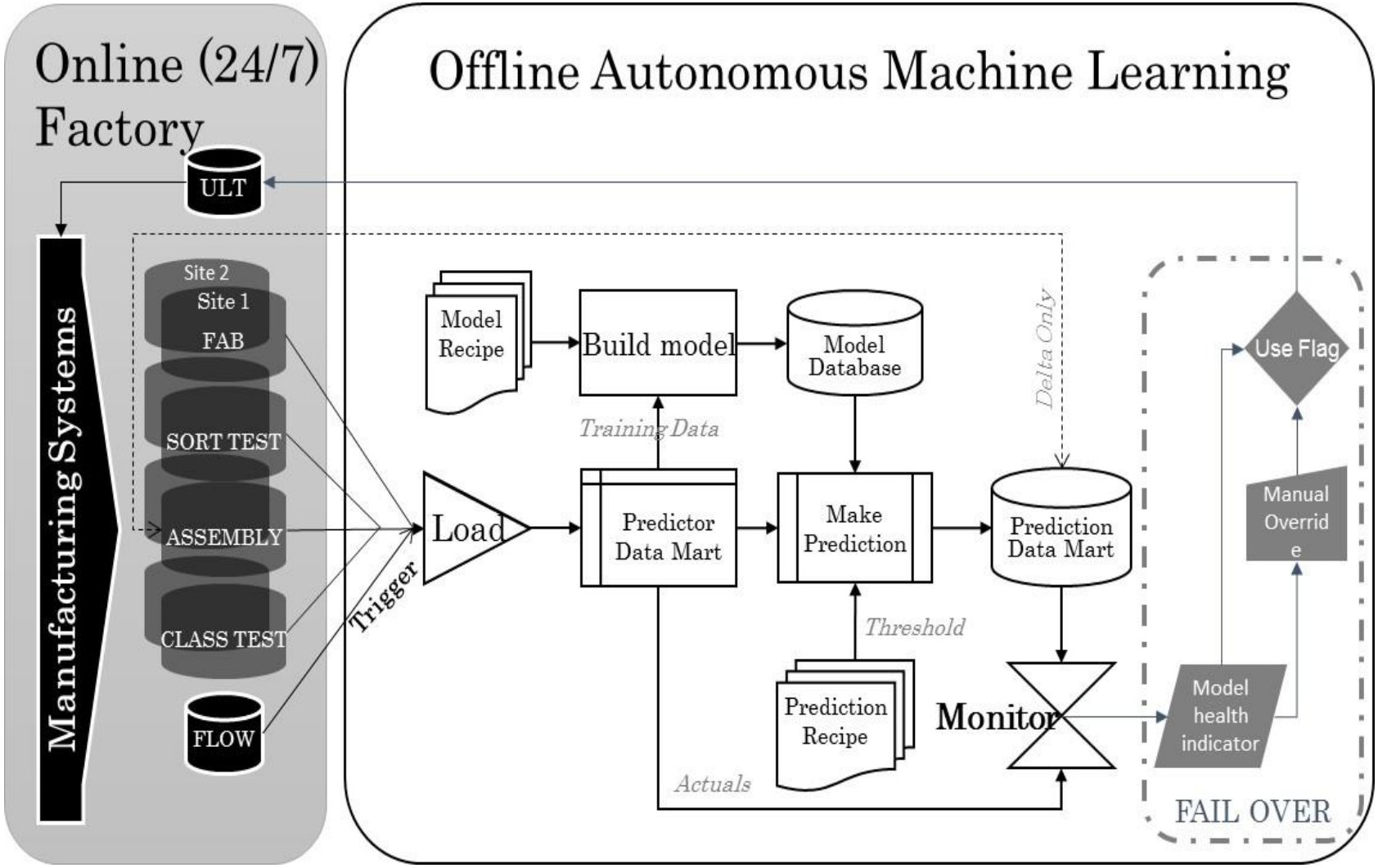


Figure 10. "Clutch" Architecture to Meet System Requirements

2.6.6 Data Preparation

As mentioned before, Intel's manufacturing operations stretch across the globe using the concept of a "virtual factory" for efficiency and cost reasons. Each site and factory has its own database to store manufacturing and test data due to the size of the data involved and given that the data is used in subsequent steps of manufacturing which needs 24/7 uptime. Thus it is impractical to obtain data just in time due to dependencies on network speeds and other uncertainties. Hence a data-mart was built to consolidate the data from multiple sites and data-sources.

The relationship between a fab-unit and the substrate is established not when the die is placed on the substrate but rather when the unit is tested for the first time and the DIE_ID is read electronically from a fuse within the die and SUBSTRATE_ID is read with a camera and, both are uploaded into the ULT database. Thus data for the die is in the fab-sort database, data for the assembly and final tests are in the assembly-test database and the link is in the ULT database, all three databases could be in different geographical locations. The raw data (Etest, Sort, Class etc.) is stored in a data mart that is periodically loaded with data for all models by going to all configured sites and collecting any new data. Any necessary cleanup or transformation, data validations, and data integrity checks are done.

It was through this effort that the team noticed that there are significant temporal shifts and dynamics in the data-stream used to build the model. Firstly, multiple product lines with slightly different manufacturing recipes and/or mix and match of die to substrates exist. Additionally, the process and test engineers continuously tweak their recipes to eek the last bit of room from the process to maximize yield or reduce cost. Hence the process is always changing.

CHAPTER 3

DYNAMIC EVALUATION FRAMEWORK (DEF)

The importance of design-for-deployment was covered in the previous chapter. It is clear that validation or testing cannot be relegated to the end of the data analytics process and needs to be incorporated within each step in order to improve the likelihood of success. Additionally, by developing an infrastructure that aids the discovery of new hypothesis and applications, one ensures that the team can pivot instead of declaring failure or disbanding. Thus a solid evaluation framework and methodology are critical to ensuring that data-analytics undertakings result in value creation for the customer or enterprise. Key challenges exist in how evaluations are done today.

3.1 Problem and Current-state

An analysis was done by Demsar [60] of the papers accepted at ICML and percentage that used accuracy with cross-validation to perform comparative evaluation. Table 1 shows part of the findings. It is clear that the majority use accuracy and resort to cross-validation. A de facto evaluation culture has pervaded experimental verification and comparative evaluation of learning algorithms. The approaches utilized to do so proceed along the following lines, with some minor variations:

1. Select an evaluation metric, the most often used one being accuracy without much thought put into the selection.
2. Select a large-enough number of datasets [the number is chosen so as to be able to make a convincing case of apt evaluation and the datasets are

generally obtained from a public data repository, the main one being the University of California, Irvine, (UCI) machine learning repository]

3. Select the best parameters for various learning algorithms, a task generally known as model tuning but mostly inadvertently interleaved with evaluation.
4. Use a k-fold cross-validation technique for error estimation, often stratified 10-fold cross- validation, with or without repetition
5. Apply paired t tests to all pairs of results or to the pairs deemed relevant (e.g., the ones including a possibly new algorithm of interest) to test for statistical significance in the observed performance difference.
6. Average the results for an overall estimate of the algorithm's performance or, alternatively, record basic statistics such as win/loss/ties for each algorithm with respect to the others [5].

There are significant implicit assumptions being made in each of these steps that could have an adverse effect on the success in the real-world.

Table 1. Accuracy and Cross-validation Current De-facto for Evaluation

	1999	*2000*	*2001*	*2002*	*2003*
Total accepted	*54*	*152*	*80*	*87*	*118*
Relevant papers	**19**	**45**	**25**	**31**	**54**
Evaluation Metric (%)					
Accuracy	74	67	84	84	70
Exclusively Accuracy	68	60	80	58	67
ROC, AUC	0	4	4	13	9
Evaluation Method (%)					
Cross validation, k-fold	22	49	44	42	56
Bootstrapping	11	29	44	32	54
Separate subset	5	11	0	13	9

To illustrate, the trifling role the above de-facto methodology plays in a machine learning system, one such typical evaluation setup is shown from Japkowiz & Shah [5] in Table 2. At first it seems that there is quite a difference between algorithms with the minimum accuracy at 62.1% for Support Vector Machine (SVM) on the 'Glass' dataset and the maximum is 99.55% with Random Forrest acting on the 'Anneal' dataset. However, as one goes down the table, it is clear that that the accuracy depends more on the dataset than on the algorithm itself.

Table 2. Accuracy Measurement Across Benchmark Datasets and Algorithms

Data Set	1NN	NB	BAG(REP)	SVM	C45	RIP	RF
Anneal	99.11	96.43	98.22	99.44	98.44	98.22	99.55
Audiology	75.22	73.42	76.54	81.34	77.87	76.07	79.15
Balance scale	79.03	72.3	82.89	91.51	76.65	81.6	80.97
Breat cancer	65.74	71.7	67.84	66.16	75.54	68.88	69.99
Contact lenses	63.33	71.76	68.33	71.67	81.67	75	71.67
Pima diabetes	70.17	74.36	74.61	77.08	73.83	75	74.88
Glass	70.5	70.36	69.63	62.21	66.75	70.95	79.87
Hepatitis	80.63	83.21	84.5	80.63	83.79	78	84.58
Hypothyroid	91.52	98.22	99.55	93.58	99.58	99.42	99.39
Tic-tac-toe	81.63	69.62	92.07	99.9	85.07	97.39	93.94
Average	77.69	78.14	81.42	82.35	81.92	82.05	83.40

On doing a variance component analysis as shown in Figure 11, we observe that only 3% of the variation (as measured by sum-of-squared variance) is contributed by the algorithm, the rest 97% is due to the datasets. Thus minor difference in algorithm accuracy especially as measured on benchmark datasets by accuracy with cross-validation are insufficient to select an algorithm for a production machine-learning system. Rather, a comprehensive match-making process is needed based on characteristics of the data and the demands of the domain.

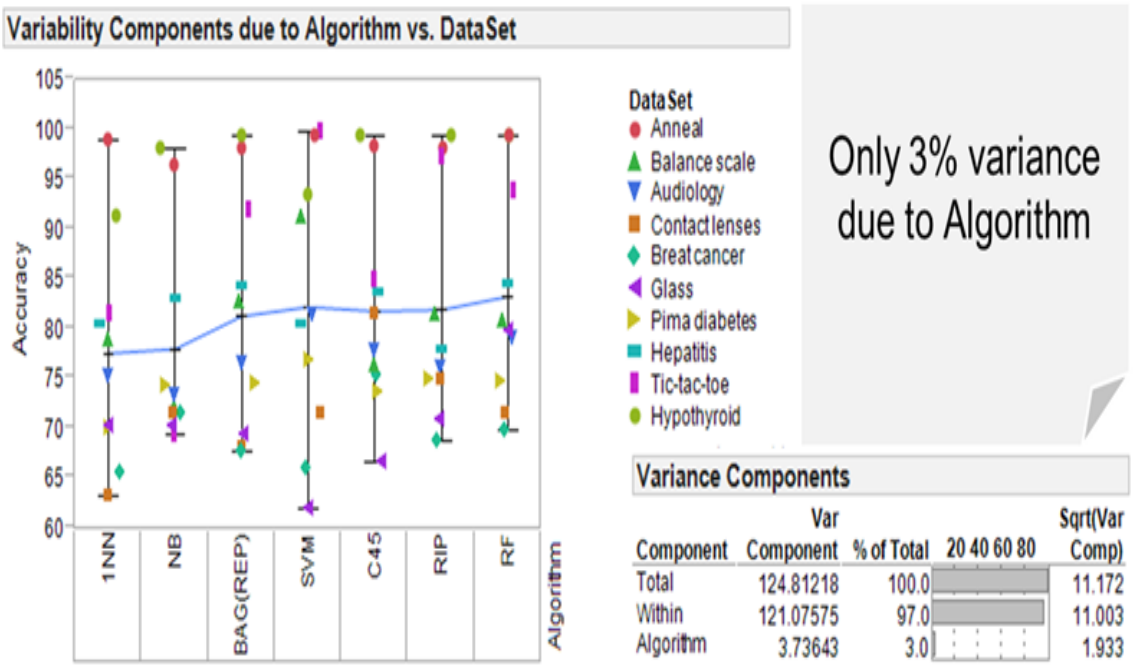


Figure 11. Variance Component Analysis of Algorithm Accuracy Across Data-sets

Other characteristics such as scalability, structure and safety mechanisms take priority over small gains in accuracy. Moreover, key differences exist between the real-world and lab environment with respect to temporal shifts and distribution shifts that quickly make static evaluation results inapplicable to the real-world. This comparative study of the current state in evaluation and makes recommendations; these are then demonstrated on the enterprise use-cases.

The following sections start with surveying the current thought on evaluation and compares metrics used. The focus of the comparison is to identify unique characteristics of the methods and metrics that would be most suitable to evaluation at each phase of the recommended data-analytics framework. There are three main components of evaluation: purpose, metric and method. The following sections will examine current thought along these three vectors, identify gaps and make recommendations.

3.1.1 Purpose of Evaluation

In most domains evaluation is driven by purpose. For example, the metrics, methods and test will be quite different based on if a person is being evaluated for general physical fitness versus being evaluated to be admitted on to a sports team. Hence it is claimed that the evaluation must be based on purpose. There are a few common purposes of evaluation which based on the focus If the focus is the machine-learning algorithm, following purposes are listed [5]:

- Comparison of a new algorithm to other (may be generic or application-specific) classifiers on a specific domain (e.g., when proposing a novel learning algorithm)
- Comparison of a new generic algorithm to other generic ones on a set of benchmark domains (e.g. to demonstrate general effectiveness of the new approach against other approaches)
- Characterization of generic classifiers on benchmarks domains (e.g. to study the algorithms' behavior on general domains for subsequent use)
- Comparison of multiple classifiers on a specific domain (e.g. to find the best algorithm for a given application task)

On the other hand, in real world scenarios some form of evaluation exists at each phase of the data-analytics process:

- Feature selection: To choose the subset of attributes or independent variables that will be used to build the production model (e.g., select a subset of 20 attributes from the available 200 measurements taken of a piece of manufactured auto-part so as to predict early-failure)

- Algorithm selection: To select the subset of candidate algorithms that stand a good chance of meeting the performance requirements of the domain and data-set (e.g., select a classifier to be used on a heterogeneous sparse dataset of demographics to predict the propensity to buy a product)
- Model tuning: To determine the optimal setting for parameters of the model in order to ensure performance on the data set (e.g., for a decision-tree algorithm, select the max-tree-depth, minimum-split-size, minimum-leaf-size and minimal-gain and split-criteria)
- Stopping criteria: To ensure the model does not over fit the data by measuring performance on an independent test dataset (e.g., for finding the epoch at which to stop the training – usually when test error stops dropping or starts increasing after initial drop)
- Ensemble selection: To select the subset of candidate models that will make it into an ensemble and/or the trial that will be used as the production model (e.g., in a random forest, determine the subset of trees that will provide input into final decision)
- Value validation: To ensure that the business or customer objective of the data-analytics undertaking has indeed been met or likely to be met (e.g., as a result of predictive targeted marketing using analytics, the sales projected sales for next quarter are 50% higher than previously expected)
- Optimization: To ensure that other constraints are not violated due to the analytics being introduced into the current system (e.g., to ensure that predictive maximization of turnover in a department-store does not lead to overall revenue reduction due to drop in advertising revenue)

- Field validation: To ensure that the analytics system can function stably and reliably in a mission-critical environment (e.g., predictive optimization of machine maintenance does not lead to safety issues)
- Model refresh: To detect if the deployed model is relevant and performing as expected after being in the field for some time (e.g., predictive flu spread rates differ greatly from actual during the general-election period)

It clear that Evaluation is a key step in the analytics process. Evaluation is where the model is evaluated from a business objective perspective to determine whether all important business issues have been sufficiently considered. At the end of this phase, a decision about the use of the data-analytics results should be reached. Thus this is the step when the sponsors of the analytics project or venture see if the investment so far will likely bear fruit.

3.1.2 Metrics

Metrics have received the most attention among all aspects of evaluation. Each metric tries to quantify performance of the algorithm. Performance is an often overused term that has a different meaning to different stakeholders. For example, it may mean resource efficiency to the systems engineer while it means speed efficiency to the web-developer. Below are some aspects that come into the picture in real-world scenarios:

- Predictive Accuracy: closeness of the predicted to actual responses
- Interpretability: output and operations understandable to humans
- Complexity: how compact (simple) is the learned model
- Robustness: capability of handling noise, missing values etc.
- Stability: robustness over time with the changing "world"
- Efficiency: time and memory needed for the training and test phases
- Scalability: How much the system's performance (e.g., speed) is sensitive to the size of the data set

Figure 12 shows an ontology of popular metrics for measuring prediction performance. To highlight the fact that there are other aspects of performance besides accuracy, the other aspects of performance mentioned above are shown. As most real-world applications of a machine-learning algorithms are for classification, determining predictive accuracy is the focus of most evaluations. Hence many metrics have emerged in this arena. They can be organized into ones that apply to the type of classes: discrete binary, discrete multi-class and continuous. Further, classification is based on how the metrics are represented: as a scalar, raw-ratio, chance-corrected-ratio, bi-number, graphical or information-theoretic.

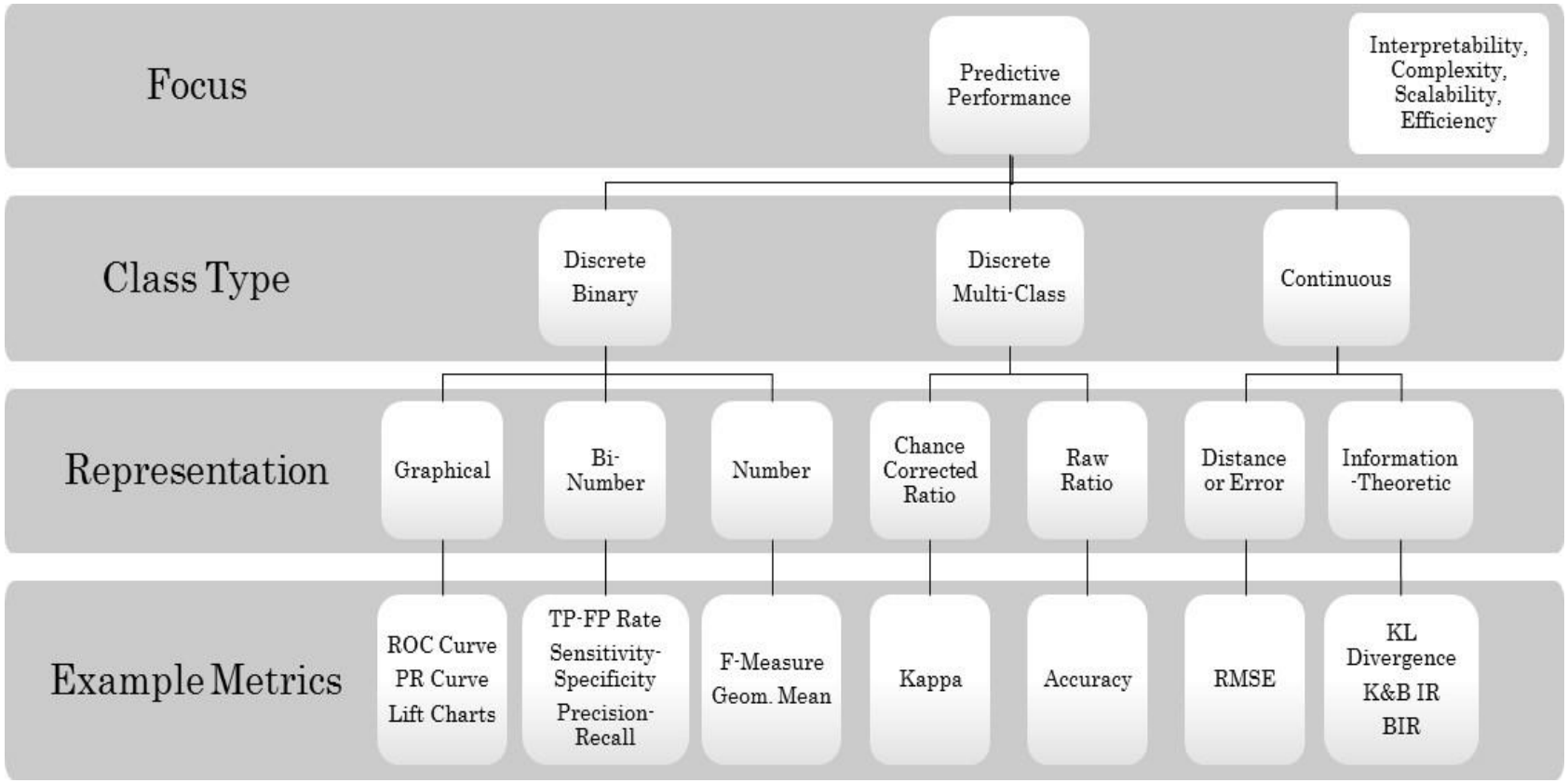


Figure 12. Ontology of Popular Metrics for Prediction Performance

The prediction performance metrics are best explained using the case of a binary class problem. Let us assume a problem with two classes: a positive-class and a negative-class. The scoring results are tabulated as shown in Table 3. - *aP* and *aN* represent the total proportion of positive and negative examples in the test dataset. *pP* and *pN* are then the predicted positive and negative proportions respectively. The two correctly predicted categories are true positive *(tP)* and true-negative *(tN)*. The other two categories, false positive *(fP)* and false-negative *(fN)* form the Type-I and Type-II errors respectively. Several ratios can be calculatd as shown in Table 3: specificity, sensitivity (aka recall), precision. F-ratio is harmonic-mean of precision and sensitivity. Accuracy is ratio of correctly classified to total-tested. Kappa is accuracy with correction applied for the proportion of rightly classified by pure chance. The positive and negative liklihood ratios are formed from cells below them.

Table 3. Popular Metrics Illustrated with a Binary-class Problem

F-score 2tP / *(aP + pP)*	***Predicted Positive (pP)***	***Predicted Negative (pN)***	Positive likelihood ratio pLr=(tPr / fPr)	Negative likelihood ratio nLr=(fNr / tNr)
Actual Positive (aP)	*True Positive (tP)*	*False Negative (fN)*	Sensitivity* tPr = (tP / aP)	Miss-rate fNr= (fN / aP)
Actual Negative (aN)	*False Positive (fP)*	*True Negative (tN)*	Fall-out fPr = (fP / aN)	Specificity tNr= (tN / aN)
Total tested (aP + aN)	Precision (tP / pP)	Negative Predicted (tN / pN)	Accuracy (tP+tN) / (aP+aN)	Kappa (Po - Pe) / (1 - Pe)
			Pe = ((aP)(pP) + (aN)*(pN)) / (aP+aN)²;*	*Po = (tP+tN) / (aP+aN)*

**Sensitivity = Recall = Hit-rate*
Informdness = Sensitivity + Specificity - 1
Markdness = Precision + Negative Predicted -1

Accuracy is the most widely used metric for quantitatively assessing the classifier predictive performance. Accuracy can be used with continuous as well as discrete classes. It is most easily interpreted, however sometimes dangerously so. Accuracy suffers from some serious shortcomings as below:

- Accuracy can be highly misleading when there is class imbalance especially if one class is of particular interest like in fraud detection.
- Accuracy does not take asymmetric misclassification costs into consideration. That is the inability to distinguishing between the "direction" of misclassification (False-Positive vs. False-negative or Type-I vs. Type-II errors). In most practical scenarios, the direction of misclassification matters and there is almost always an unequal misclassification cost associated with each class.
- The other problem with accuracy is that it does not take into account the effect of chance i.e., the portion of correct classification that could be achieved just based on chance. However, this serious shortcoming is often ignored in many machine-learning projects.

Accuracy is best used for gross-reality check of algorithms in the early stage of data exploration. It also is suitable during the development phase to formulate algorithm convergence criteria. However, its use to determine the suitability of an algorithm for a particular purpose or data set is limited and hence one needs to exercise extreme caution in putting too much faith in raw accuracy values [61]. Accuracy is certainly not suitable to compare algorithms to one-another as averaging across multiple trials or data sets is not meaningful [62]. Table 4 shows a comparison of the advantages and shortcomings of popular alternatives to accuracy.

Table 4. Comparison of Common Measures of Predictive Performance

MT	ADVANTAGES	SHORTCOMINGS	USAGE
ACCURACY; RMSE	- Universal - can be used for discrete and continuous classes and scales to multi-class problems - Easy interpretation	- Does not discern (uneven) costs of misclassification - Misleading if class imbalance exists - Does not discount for chance agreements	- Uniform misclassification-cost problems with balanced class distribution - As a gross reality check - Determine convergence of algorithm
PRECISION-RECALL	- Gives granular view for the particular class of importance regardless on class-imbalance	- Not a scalar - two numbers need to be reported and hence not straightforward to interpret - The other classes are ignored	- Suitable for problems where a particular class is of utmost interest (like anomaly detection) - Not suitable for multi-class problems - Information retrieval
F-MEASURE	- Combines precision and recall into one metric with relative weighing between the two - Scalar between 0 and 1 - easy to compare - Easier interpretation	- Ignores other classes - Relative weights are usually unknown and uniform weight assumption is not representational	- Suitable for problems where a particular class is of utmost interest and importance of precision versus recall is known - Not suitable for multi-class problems as the other classes are ignored
SENSITIVITY-SPECIFICITY	- Measure that gives recall indication about both classes - Can be extended to multi-class	- Not a scalar - two numbers need to be reported - Extending to multi-class result in same number of measures as classes	- Suitable for problems where the misclassification cost is not known beforehand but needs to be applied later. - Use to compare or tune algorithms on a per-class basis
ROC CURVE	- Graph that gives a trade-off curve between sensitivity (tPr) and (fPr = 1-specificity) - Information-rich over operating-range - Not prone to class imbalance	- Not a scalar so there is interpretation overhead - Needs evaluating over the operating range - Curve resolution limited by available data - Mainly applied to two-class problems	- Highly useful in tuning and optimization of the algorithm to "dial-in" optimal performing region. - Not suitable for multi-class problems
AUC	- Scalar summary of the ROC curve - Averages performance over all cost-ratios - Related to Gini index	- AUC does not correct for agreement by chance - Misclassification costs and direction ignored. - Computing is cumbersome	- Use instead of accuracy. - Best when reported with the actual AUC curve - Not suitable for multi-class problems
KAPPA	- Corrects for chance agreement - credibility - Normalized on a -1 to 1 scale (almost 0 to 1 for larger class imbalance) - Affected aptly by class imbalance - Relates well to ROC	- Does not discern (uneven) costs of misclassification - Experts differ on how to compute the chance agreement	- Realistic metric for problems with imbalanced class - Works well when one class is of utmost importance - Superior to accuracy as single-number measure - Compare and choose algorithms, tuning and costless evaluation

Precision & Recall restrict focus to one class of interest thus cannot be extended to multiple-classes. They can be effectively used to address situations like fraud-detection where one class is of utmost interest and costs mainly revolve around true-positive and false-positive. This metric is rightly popular in information retrieval applications but requires reporting of two-numbers. The F-measure is a single-number representation of precision & recall.

F-measure is the harmonic mean of precision and recall resulting in a scalar that can range between 0 and 1 hence allowing easy interpretation: 1 is a perfect classifier. A weight can be applied to balance the relative cost of true-positive and false-positive although the performance on the negative class is ignored. F-measure is not extendable to multi-class problems and focus remains on the particular class. Thus again suitable for information retrieval and fraud-detection applications.

Sensitivity & Specificity separate the correctly classified proportions thus allowing one to distinguish the direction of misclassification unlike in accuracy. However, the price-paid is in terms of losing the simplistic scalar indicator and replacing it with two-numbers that need to be interpreted in unison. The concept itself can be extended to multi-class problems essentially reporting the proportion of correctly classified of that class. This also means one has to deal with as many indicators as classes – can become difficult if used for comparison of algorithms. However, effective if used for tuning the performance to favor certain classes without needing to know exact misclassification costs beforehand.

The *ROC* (Receiver Operating Characteristics) curve is graphical plot of true positive rate (sensitivity) against false positive rate (1-specificity) [63]. Consequently, a ROC curve is a collection of various confusion matrices over

different varying decision thresholds for a classifier. The advantage is that it allows characterizing the classifier across its operating range of classification thresholds instead of a point estimate like accuracy. The trade-off can be made between the true-positive-rate desired and the lowest tolerable false-positive-rate. Once plotted, the curve can be used for tuning and trade-off based on relative costs of misclassification which can be applied after the fact. However, plotting the ROC at finer resolution requires large quantities of test data. The curve itself is not prone to class-imbalance as both the axis are normalized ratios to the class-proportion. The ROC is well understood in the engineering domain and hence familiar although a two-dimensional graph is not convenient for quick comparison of algorithms. The area under the ROC curve known as the AUC is the remedy.

The *AUC (Area Under the Curve)* has been used as a way to reduce the information to a scalar ranging from 0 to 1: perfect classifier has AUC of 1 and an AUC of < ½ means it is probably worse than a random guess; because, the 1:1 line would represent a random-guess algorithm [64]. Although, an AUC of ½ could be due to other reasons. The AUC represents the performance of the classifier averaged over all the possible cost ratios. AUC represents the ability of a classifier to rank a randomly chosen positive test example higher than a negative one. The AUC is closely related to the Gini cost-function as well. The AUC can be used effectively instead of accuracy to compare performance of different algorithms due to immunity to class-imbalance as compared to accuracy. However, misclassification costs are not taken into account. Additionally, AUC does not correct for agreement by pure-chance. This can be remedied somewhat by only considering the area above the 1:1 line. However, the problem remains.

Kappa corrects for the agreement by chance. Kappa is the ratio of the difference between the observed and chance agreements, to the maximum possible agreement that can be achieved over and above chance - Table 3 shows the equation for Cohen's Kappa. Alternatives re available around how the chance-agreement is computed based on the purpose of evaluation; however, Cohen's Kappa is most commonly used. These measures that correct for change originate in statistics and are called agreement measures (or interclass correlation statistics or interrater agreement measures). The agreement measures take the marginal probability of label assignments into account to correct the estimated accuracy for chance. Like accuracy, Kappa can be extended to multi-class problems. Unlike accuracy, Kappa is not prone to providing overly optimistic results especially in existence of class imbalance where the important class is in minority. Because of these characteristics, Kappa should be utilized instead of accuracy to ensure viability of true benefits of deploying the ML algorithm based system in the field.

Kappa is known to be affected by class-imbalance. To examine if this effect is meaningful and desired, the relationship between Kappa and ROC space was examined. The desired behavior of a metric would be that while, it is not overwhelmed with poor performance on the unimportant class, it still shows degradation when the performance is poor on that class to indicate the burden of false-positives. Kappa was computed over the entire ROC place using a 11x11 at all points on a 11x11 grid (resolution of 0.1) for six different (positive to negative) class imbalance (skew) ratios. The results are shown in Figure 11 as six 3D gradient surfaces over the 2D ROC plane for various class ratios.

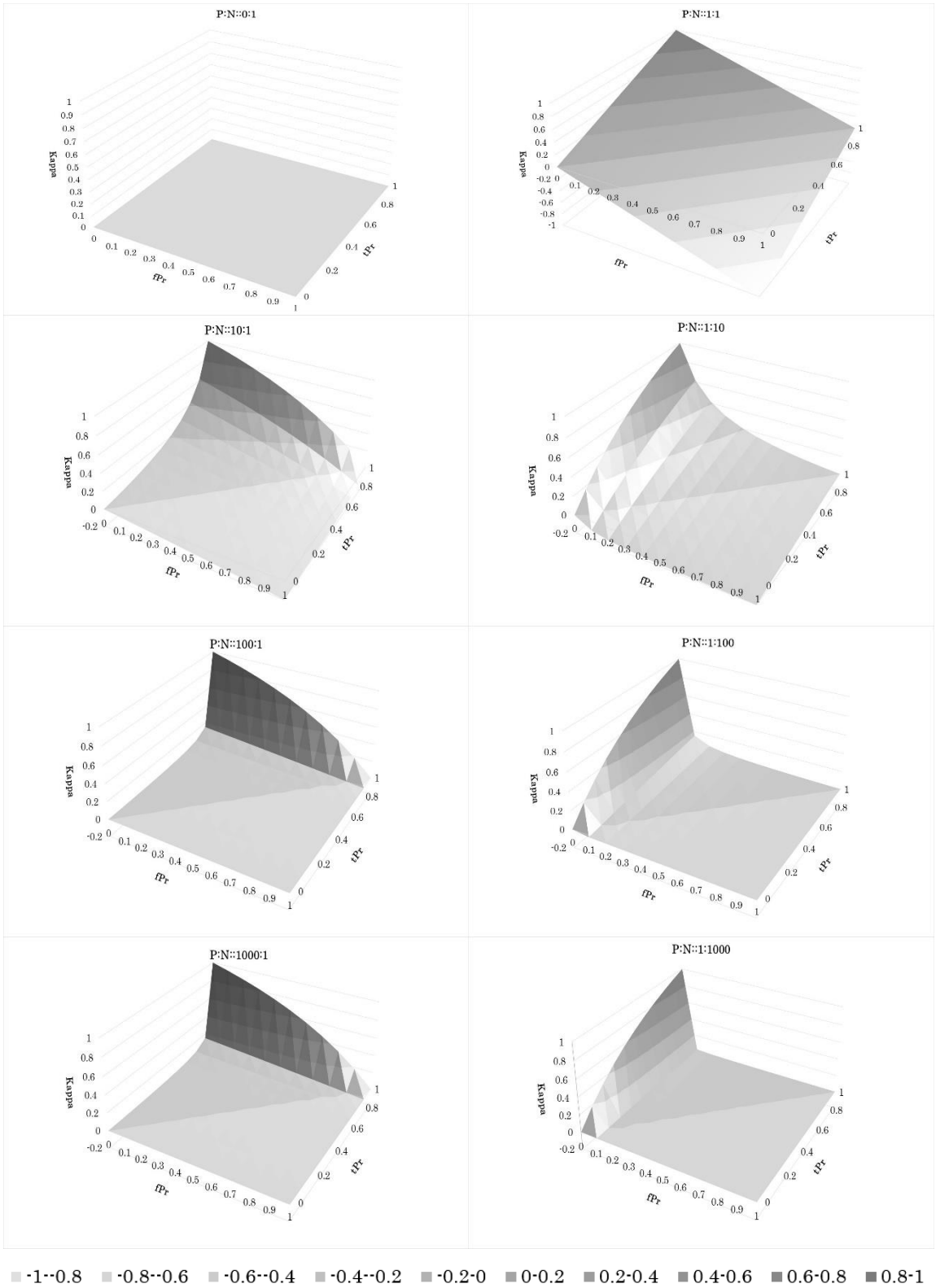


Figure 13. Kappa Profiles over ROC Space for Class-imbalance Ratios (P:N)

For the 1:1 (balanced-class) case, Kappa is seen to be highly representative of the ROC space where it is zero along the slope=1 line which represents the random classifier. Kappa increases as we progress diagonally towards the (0,1) point which is the perfect classifier and hence Kappa takes the maximum possible value of one. On the opposite side is the (1, 0) point where Kappa is -1 hence rightly indicating poor performance.

Now when we move the class imbalance to 1:10 – there are 10 times as many negative examples as there are positive; the positive being the class of interest. Once can observe that this has in-fact affected Kappa. The slope=1 diagonal line still holds the Kappa of 0; however, the negative values do not go beyond -0.2. As we would like most classifiers to be better than random, this is not too consequential. The positive values have taken a concave shape wherein, Kappa values closer to the diagonal are much lower than in the balanced-class case but exponentially rise to 1 as we approach the (0,1) perfect classifier corner. This thesis proposes that the behavior is exactly what is desired in a good metric as explained next.

Table 5. Two Examples Illustrating the Desired Bias in Kappa

	predicted Positive (pP)	predicted Negative (nP)		
Example 1:				
actual Positive (aP) = 10	9	1	tPr = 90%	*Accuracy = 98%*
actual negative (aN) = 100	1	99	fPr = 1%	*Kappa =0.89*
Example 2:				
actual Positive (aP) = 10	9	1	tPr = 90%	*Accuracy = 90%*
actual negative (aN) = 100	10	90	fPr = 10%	*Kappa =0.57*

Consider two examples as shown in Table 5. In both examples, the class imbalance is 1:10, positive to negative. In the first example, there is exactly one example misclassified in each direction with true-positive-rate (sensitivity) at 90% and false-positive-rate at 1%. Hence the performance is quite good and correspondingly, accuracy is at 98% and Kappa is at 0.89. In the second example, the class imbalance remains the same and so does the true-positive-rate (sensitivity). However, now the false-positive-rate has increased ten-fold to 10%. The accuracy, enamored by the class-imbalance, remains over 90%; however, Kappa has dropped to 0.57. Given there has been a ten-fold increase in false-positives, the drop is well justified especially if the positive class is of utmost importance like in fraud-detection, cancer-detection and error detection in the industrial space.

Now we move our attention to the 1:100 imbalance case and observe that Kappa goes higher than 0.1 only when the false-positive-rate (fPr) is lower than 10%. This is apt as any false-positive rate higher than 10% would not be tolerable for a 1:100 data set. Once in the region of fPr < 0.1, Kappa rises as the true-positive-rate (tPr) goes up, reaching 1.0 at the (0,1) point. In the 1:1000 imbalance case, the behavior of "flat" almost zero at all places on the ROC plane except closer to the fPr=0 line, becomes highly pronounced. Again, this is the exact behavior desired. In semiconductor manufacturing the detect rates are measured in DPM (defects per million), similar to PPM in biochemistry. Thus class-imbalance is severe. So is the case in many real-world scenarios. As the behavior of Kappa is attuned to reality, it is proposed that Kappa is an apropos measure for real-world evaluation of machine learning algorithms for discrete classes. Class imbalance in the other direction results in similar behavior with respect to the fPr=1 line thus Kappa is symmetric.

There are a multitude of measures that have been proposed that are suitable to the objective and specific needs of the domain. While Lift Charts (true-positive against sample-size) are used in marketing domain, precision-recall curves are used in document retrieval and can be more relevant in heavily imbalanced data [65]. Cost curves plot the error rate (1 - accuracy) against probability of positive class. They are point-line duals of ROC curves in that each point on the ROC convex-hull is represented by a line. Informedness and Markedness are used in psychology [66].

RMSE (Root Mean Square Error) is used as a general purpose performance measure where the predicted variable is continuous. RMSE is highly correlated to accuracy when a threshold is applied and also correlates with AUC across thresholds. It also has the same limitations as accuracy.

The *KL divergence* information theoretic measure that quantifies the difference in entropy between the learned classifier and true class distribution. While valuable in theoretical treatment of classifiers, as true distributions are seldom known, it has limited practical values. *Bayesian Information Reward (BIR)* remedies this shortcoming by using the empirical class distribution instead of true class distribution and adding a penalty for misclassified classes [67]. Kononenko and Bratko's *information score* is another information based approach that measures the decrease in the information needed to classify the instances as a result of learning the classifier [68]. It is analogous to Kappa in that it denotes the effectiveness of the learning process over and above the information conveyed by the empirical priors.

In summary, there is no one-size-fits-all metric and thus it is advisable to put thought into selecting a proper metric based on the data set, purpose and phase of evaluation. That said, Kappa should be strongly considered instead of accuracy [62].

3.1.3 Methods

This section is devoted to scan commonly used methods of validation and identify shortcoming with respect to real-world scenarios and make recommendations. There are a handful of methods and a multitude of derivatives used for evaluation of machine-learning algorithms sometimes without a thorough understanding of the assumptions and pre-requisites. This section examines the popular methods of evaluation and proposes simple guidelines for method selection based on size of the dataset. It also identifies the combinations that should be refrained from to ensure that false conclusions are not drawn from flawed analysis. By following these guidelines once can ensure that evaluations at all phases in the analytics process leads to the right decisions thereby improving the chances of success and/or indicate early-failure to pivot accordingly.

The simplest (and cleanest [69]) method for evaluating learning algorithms is the hold-out method. The available data-set is divided (randomly) into two sets (usually a 2:1 split). One set (usually 2/3rd) is used to train the algorithm and the rest is used for testing. While this is the cleanest way, often it is cited that in situations marred by lack of data, not using all of the available data for training robs the algorithm of its best chance of reducing bias. As academia usually suffers from shortage of data, most of the effort in research has gone into proposing clever ways to reuse or multi-use the data for training as well as evaluation. Some of these are covered below.

Random-subsampling is where the hold-out method is repeated a certain number of times and the resulting metric from all the runs is averaged. Although this allows use of a larger amount of data for the learning algorithm, for small data

sets, the models during all of the runs turn out to be close to each other thus not helping with the bias. Additionally, as the same data samples are being reused, false-replication occurs and any measure of variance either high or unreliable. As performance is worst on small data sets, it defeats the purpose of using the method.

Cross-validation attempts to remedy the problem by (randomly) dividing the data set into *k* equal folds (portions). At each run, one of the k folds is held-out for testing and the other (*k-1*) folds are used for training. The process is repeated k times and the results are averaged. As distinctly different test-sets are used on each run, the bias and variance are tolerable for mid-sized (hundreds) data sets. The most popular value for k is 10 due to good balance between bias and cost of multiple runs. Below 5, bias tends to be high. Furthermore, if the number of classes is high, the variance tends to shoot up. Stratified sampling can be used to control the variance and benefits the bias as well. Stratified sampling is where one ensures that each of the k folds' have the same class distribution.

Leave-one-out is an extreme case of cross-validation where k=N, the number of samples in the data. The error estimate is almost unbiased because the training takes place on all but one training example of the available data and because the testing sets are completely independent. This estimate, however, suffers from high variance on small samples because of the extreme behavior of the tested classifiers on the one-case test sets. For example, a completely random data set with binary classes would result in an accuracy of 0 and a misleading variance of 0. For larger sets, the technique gets expensive, for smaller sets, variance is a problem. Thus there is little practical use for this method.

Bootstrapping is a method where instead of splitting the set randomly, N samples are selected with replacement from the data set and the samples that did not make the selection are used for testing. Thus the algorithm has the benefit of N samples although some are repeated. Probabilistically, 36.8% of the samples will go unselected during one and thus 63.2% of the samples are in the training set. The process is repeated and the results are averaged. The bias is known to be pessimistic and thus an adjustment is applied based on the training error. After the adjustment, the bias can become optimistic. Bootstrapping has been shown to perform well in the cases in which the sample is too small for cross-validation or leave-one-out approaches to yield a good estimate. In such cases, a bootstrap estimate shows less variance; however, again this could be due to false-replication.

Permutation test first creates 'bogus' data by taking the genuine samples and randomly choosing to either leave their label intact or switch them. Another method is to keep the labels as-is and scramble the feature values. Once this 'bogus' data set is created, the classifier is run on it and its error estimated. This process is repeated a very large number of times in an attempt to establish whether the error estimate obtained on the actual data is truly different from ones obtained on 'bogus' data sets.

The random-subsampling and leave-one-out have limited use due to poor bias and variance respectively on small data-sets. Although in cases of extremely small datasets, the k-fold cross-validation often does not perform as well as bootstrapping (and using more folds do not help), it does not suffer from drastic problems the way bootstrapping does in terms of increased bias or when the true error expectations are not met. Moreover, the bias of bootstrapping depends on the algorithm as well because some algorithms use duplicate samples but others do not. It has also been

shown that the k-fold cross-validation does not estimate the mean of the difference between two learning algorithms properly [70].

Table 6. Bias, Variance and Suitability of Popular Evaluation Methods

	Description	*Bias*	*Variance*	*Suitability*
Hold-Out	Randomly sub-divide into separate training (2/3) and test-set (1/3)	Low (high for small sets)	Medium (high for small sets)	Large data-sets (thousands); not-practical for small data-sets
Random subsampling	Hold-out method repeated m-times	High for small sets	High for small sets	Not recommended
k=10, 20 k-fold Cross validation	Sub-divide into k distinct "folds"; Do k-times: one fold for test and other folds for training.	Low if stratified	Medium (high for multi-class)	**Recommended for most cases with limited data-size (hundreds)**
k≤5		High, pessimistic	Medium	Not recommended
Leave-one-out: k=N, N/2, N/5	Each fold has exactly one sample	Low	High	Not recommended - Ex. random bi-class data has 0 accuracy and 0 variance
Boot-strapping	Randomly sample N with replacement for training set; the ones not sampled are test-set (0.368N)	High, optimistic	Low (due to false replication)	Small Data sets (tens); Memorizer would give high accuracy on random data

Table 6 shows the bias, variance and suitability of sampling methods. The choice of a particular resampling method also affects the bias–variance characteristic of the algorithm's error. This then can have important implications for both the error estimation and subsequent evaluation of classifiers (both absolute evaluation and with respect to other algorithms) and its future generalization as a

result of the impact of this choice on the process of model selection. Thus most sampling methods are not appropriate for reliable comparisons. Next, real-world implications on evaluation methods are covered.

3.2 Real-World Challenges

Key challenges exist in how evaluations are done today mainly due to difference in demands and priorities of a lab environments versus industrial real-world scenarios. These challenges if not understood and addressed lead to the failure of data-analytics and machine-learning efforts after much has been sunk in terms of resources and time into the previous development phases.

- The most prominent one is that of *time-traveling* the lag and leap between model-training and prediction that invariably exists in real-world systems an often overlooked but critical aspect of evaluation.
- The second is that most evaluations are done in a static environment where the test-set is siphoned from a data set that is itself is a snapshot in time. Thus making the implicit assumption that the real-world application systems are static. Real-world scenarios are seldom this form.
- The third is that although there has been a fair coverage of evaluation methods and metrics, there is a lack of clarity on their sensitivities. Predictions are seldom presented with confidence intervals. Thus it is never clear whether the prediction is within the system error tolerance.
- Lastly, most evaluations are focused on the prediction performance and do not pay much attention to the other aspects of performance.

Each of the challenges are described herein using a simple data science application in marketing: propensity modeling. Propensity modeling is an

application wherein the machine-learning model is chartered with identifying those customers who are most likely to exhibit a certain behavior (like responding to marketing material or at risk of churn etc.).

3.2.1 Prediction Leap

In many real-world scenarios there is a leap. Leap is the period into the future that the machine-learning algorithm needs to predict. In the propensity modeling scenario, say the model needs to predict customers likely to churn in the next month (a = 30). Whereas the subscriber could cancel anytime between within the *actualization window (a)* anytime between day-one (t_1) and last day of the month (t_{30}), the predictions need to be available on day zero (t_0). Thus, assuming that there is no lag in data-collection and models can be built instantaneously, the only data available to build the model is from the previous-day and going into the past ($t < t_0$) as shown in Figure 14. The actual customer choices are determined only after day-30 ($t > t_a$) at least for the ones who canceled – this is the actualization window. The ones who stayed may very well cancel after day 30; however, this has to be out of

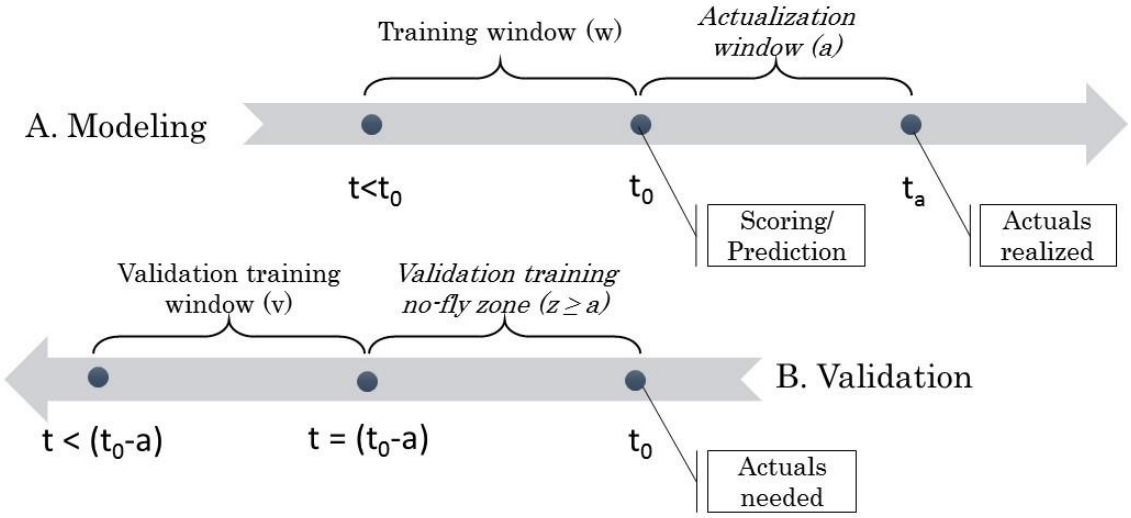


Figure 14. Actualization Windows and Faithful Recreation During Evaluation

scope of the analysis in-order to bound the problem. In reliability engineering, this type of data is referred to as 'right-censored' data [71]. Data from the actualization windows will not be fully available at the time when the model is built (at $t=t_0$).

Now consider the evaluation scenario to determine the viability of the chosen algorithm before deployment to the field (say at time t_0). One would need an actual test-set where customers have already made their choices over a one-month period. Thus for the model evaluation to be true to life, one needs to ensure that the test-set is shifted forward (into the future) from the training set by thirty days. As one cannot go into the future, instead, the validation-set must be limited prior to thirty days into the past ($t < t_0$-a). Furthermore, no data between t_0-a and t_0 can be used for training during validation (as shown by the *'no-fly zone'* in Figure 14 as this data will not be available when the actual model will be built. Using this data for training would contaminate the model with information not available to it deployment. Thus in this scenario, cross-validation is simply not representational.

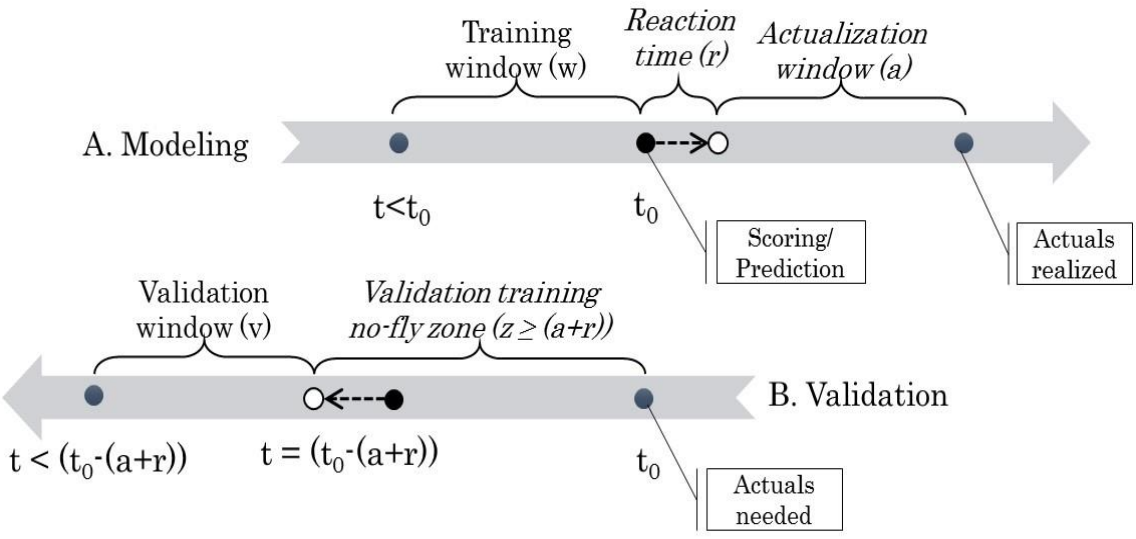


Figure 15. Reaction Time in Addition to Actualization Window

There is yet another component that adds to the temporal shift between training and test sets for validation as shown in Figure 15. This shift is due to possible *reaction-time (r)* after the prediction has been applied to the system. For example, in the propensity model, say we have identified the customers that are most-likely to churn. We would like to run a proactive retention campaign to determine if that helps in any way. Thus now the time to run the campaign and reaction time must be added to the delay between prediction and actualization. This results in the no-fly zone extending from ($z \geq a$) to ($z \geq a+r$) as shown in Figure 15.

In simple linear regression models the confidence interval around the line fans out as we move away from the mean, similarly the confidence in predictions from a machine-learning model will decrease as we move into the future. Thus a large prediction leap is highly demanding on the algorithm's abilities whereas $a+r=0$ is the simplest case. In real-world scenarios the leap ($a+r$) is seldom small; yet, most evaluations implicitly make the assumption that $a+r = 0$ by using parts of the same dataset for cross-validation. In the propensity example, using cross-validation will not make sense as the data is time-sensitive. However, if such a technique it applied, it would result in great accuracy. The spectacular results could lead to unrealistic expectations which will ultimately not be met in the field. Thus resulting in failure of the data-analytics project and loss of confidence in the methods.

The practice of reusing a fraction of the contiguous data set to perform evaluation (like in cross-validation and bootstrapping) perhaps comes from the fact that most academic research is marred by the scarcity of datasets [72]. However, today both data storage or processing are not cost-drivers and sensors spew more data than can be utilized. Thus small data sets are not necessarily the problem.

However, it is critically important that evaluations are done in conditions that represent reality with high-fidelity. Some assumptions made in lab environments might seem trivial but have a huge impact in the field.

3.2.2 Prediction Lag

The prediction leap is not the only temporal aspect to be considered in real-world scenarios. Couple of other temporal shifts exist that are associated with data gathering. In real-world scenarios, attributes are assimilated from disparate sources, and combined to form the set that has the best predictive power for the application of interest. For example, weather data from the NOAA database may be combined with data from the building for HVAC optimization. Thus by the very nature, delays might exist wherein at the moment the model is being built not all data up to that moment is available for consumption as seen in Figure 16.

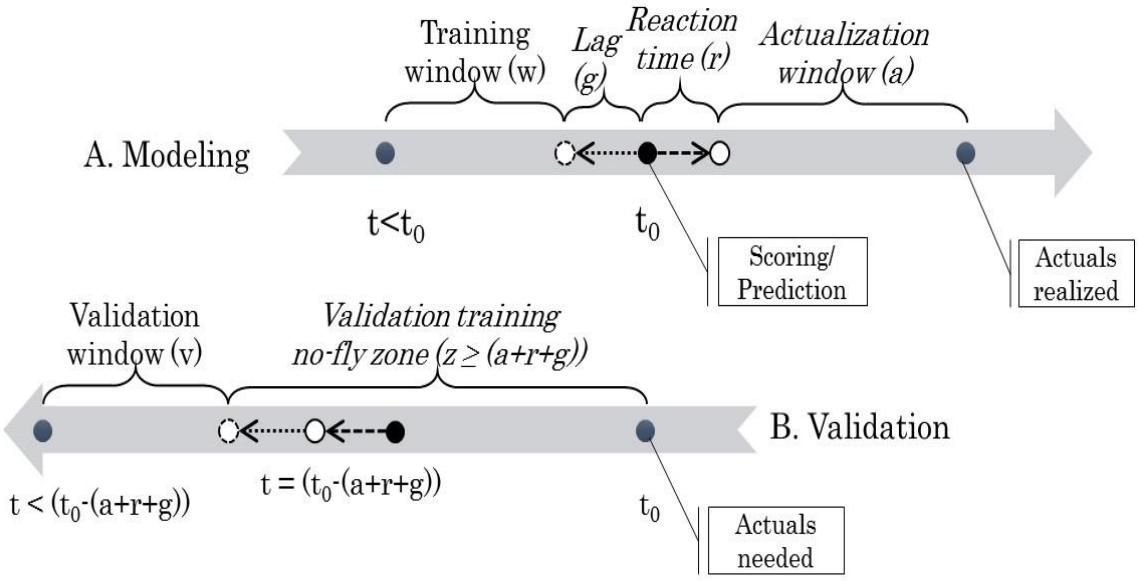


Figure 16. Overall Time-shifts in a Prediction System

In the propensity modeling example, some key attributes for modeling might not be available in the database and need to be manually rolled up after field-sales

representatives enter their data. As this is a manual process there is bound to be a delay in data availability. Thus further adds to the shift between modeling and prediction. Additionally, the modeling process might add further delay. It is important to note that while the prediction lag can be reduced through system improvements, the prediction lead can seldom be affected without changing the fundamental business or engineering process.

In summary, as real-world scenarios involve non-zero temporal shifts, it is highly recommended to perform out-of-time validation in order to ensure that the results obtained in the lab environment are in fact going to be realized when the model is deployed in the field [73].

3.2.3 Window Size

The previous sections dealt with the temporal-shifts in evaluation of an algorithm; however, it is equally important to carefully consider the windows: training window and actualization window.

The *actualization window* in many cases is strongly influenced by the turnover or cycles in the domain of application. For example, financial domain might implicitly have a quarter or year as the cycle, while an hourly or daily cycle is common in HVAC scenarios [74]. There are significant implications to the model which are important to account for and understand. In the churn propensity model, too short of an actualization window might mean too-few churn examples resulting in imbalanced data. Whereas a large actualization window might mean some of the churn is cause by long-term trends and hence consequently, the training window needs to be extended to capture those trends. Next, *training-window* is discussed.

As most are enamored by the challenge of obtaining data, the tendency is to use most if not all available data for training. However, this might not always be the best choice when it comes to building production-worthy models. In the churn propensity model example, although data might be available extending several years into the past, all of it may not be relevant to the prediction task at hand.

Many factors affecting churn might have changed in the recent past. The product-profile might have changed, customer-service might have been outsourced, new competition might have entered the market. While these factors might be relevant to the churn problem as a whole, they do not necessarily apply to the immediate task of predicting churn propensity for the next 30 days. Including all data in the model may make learning more difficult as the modeling process now needs to weed out factors that are not immediately relevant to the task. Another illustrative example is overall average reviews of a software app versus the review in the last week or month. As software is updated frequently, the past reviews may not apply to the product available for download today. Thus many questions arise about the training window that need careful consideration and may also require experimentations. Some are listed below:

- What span of time should be used to form training data?
- What size of training data is acceptable: speed vs. accuracy?
- Is sampling of the training data tolerable? If so, what rate?
- Should the sampling be weighted by time (to simulate fading "memory")?
- What attributes should be included in the model versus branching into categories and building a model per category?

The list above is not meant to be exhaustive. However, one is best served by thoroughly analyzing the modeling task at hand to ensure all aspects of selecting the length and nature of the training window are considered carefully without resorting to "default" options. There are aspects specific to the model-building process itself like feature-selection, cost-function, parameter-tuning, etc. which are out of scope of this dissertation but are important to consider.

3.2.4 Dynamics

Real-world systems are mostly dynamic by nature. Characteristics of the system change over time, cyclically, in trends, shifts and composition. In the churn propensity example, although the demographics of the customers do not by themselves change drastically, other influencing facts do, like, product features, support-quality, competitive landscape. In the building HVAC optimization, the external temperature and humidity change on a daily cycle as well as a seasonally. In addition, there could be changes to the characteristics like addition or removal of blinds. Thus all models have a shelf-life and based of the application, this can range from few minutes to several weeks. Changes can be the following:

- Change in attribute ($\mathbf{x}$) distribution (cyclic, trends and shifts) – i.e., the probability distribution $p(\mathbf{x})$ changes but $p(y|\mathbf{x})$ does not.
- Change in relationship between attributes and predicted (y) – i.e., the probability distribution $p(y|\mathbf{x})$ changes but $p(\mathbf{x})$ does not.
- Change in attribute set relevant to predictor – i.e., is z is the new attribute set and f and g are models, $||y - f(x)|| \gg ||y - g(z)||$.

Another type of drift is caused by the introduction of the machine-learning algorithm itself. It is the result of the unintended loops that might occur in machine-

learning deployments. Let us say a model A is being designed by the media placement group to prioritize advertisements. Prior to that, the marketing group had setup model B to score customer interest in products based on the length of time the customer spends on the page. Creators of the system A find that an 'interest' score is available in the database and use it as one of the features for their model. Thus an unintended loop is formed which could cause the system drift and runaway.

One approach is to attempt to build a model that would incorporate the cycles and drifts and maintain stability over time [75]. However, this approach can only go so far and ultimately the model will need to be refreshed. The straightforward answer seems to be to setup monitors to detect deterioration. The key question is: what to monitor? It can be difficult to establish useful invariants, given that the purpose of machine learning systems is to adapt over time. Often a simplistic metric like prediction bias can be useful to detect change in relationship between the features and predicted variables.

Many other questions arise around the model-refresh: time-based or deterioration-based; if time based, what frequency; if deterioration-based, what is the measure of deterioration and what threshold etc. Additionally, it is often necessary to pick a decision threshold for a given model to perform some action. The threshold is often picked manually to achieve a certain tradeoff between metrics like precision and recall. Thus if a model updates on new data, the old manually set threshold may be invalid and hence also now needs to be updated. Due to dynamics, thus performance must be validated across a long enough time-span to be credible.

3.3 Proposed Evaluation Framework

The evaluation framework proposed is in line with the analytics framework shown in the previous chapter. Just as a software development team has a sandbox or development environment to try new features and techniques before releasing them to production, the evaluation framework must allow experimentation and thorough evaluation of the machine-learning solution before deployment to the field. The best development environment is typically the one that represents the production environment with high-fidelity yet allowing flexibility for experiments. Thus it needs to consist of the key elements found in the real-world production environment. Three key elements make-up the predictive-analytics flow: data gathering and cleaning, data modeling and prediction. Accordingly, the three key elements of an evaluation framework are a data-mart, a time-looping mechanism and a way to specify model generation and testing for experiments.

3.3.1 Build Data-Mart

It is well known that most of the analytics project time is spent in gathering and cleaning the data. If one enters the analysis with the expectation that this is a once and done activity, it is bound to lead to disappointment. Because, as the project proceeds into the modeling phase, it is most likely that an additional attribute or piece of data, aggregation or transformation is needed that will have a much bigger impact on the model performance than the algorithm itself. If this new need cannot be fulfilled in a reasonable timeframe, it could lead to project delays and ultimate failure. Additionally, in order to address the need for a dynamic evaluation, it is imperative that both the training and scoring data can be obtained automatically with a simple query or import that does not require manual manipulation. Thus a

large enough sample of the data should be collected in a purpose-built dedicated database. It not a good idea to reuse the corporate data warehouse. Instead, creating a separate data mart is recommended for the following reasons [37]:

- Modeling the data will involve highly active use of the data warehouse. It will often mean joining many tables together and accessing substantial portions of the warehouse. A single trial model may require many passes through much of the warehouse. Thus causing resource allocation issues.
- It is often overlooked that modeling entails modifying the data from the data warehouse. One may want to bring in data from outside the company to overlay on the data warehouse data or add new fields computed from existing fields. Additional data may be gathered through surveys. Other projects building different models from the data warehouse (some of whom will use the same data) may want to make similar alterations to the warehouse. However, data warehouse administrators do not look kindly on having data changed in what is essentialy a corporate resource.
- One more reason for a separate database is that the structure of the corporate data warehouse may not easily support the kinds of exploration needed to understand the data. This includes queries summarizing the data, multi-dimensional reports (sometimes called pivot tables), and many different kinds of graphs or visualizations.
- Lastly, performance, reliability and other considerations might necessitate that this data is stored in a data-mart with different physical design.

Thus regardless of the actual physical-design, a separate data-mart is essential to improve the likelihood of the success of the venture. The steps recommended in building a data-mart for analytics are shown in Figure 17. The 1st phase is an investigative process followed by a transformative phase 2 and the final realization phase 3. Note that these tasks might not be performed in strict sequence and are chosen and re-performed as need arises. Phase 1 consists of data collection, description, selection and inspection.

- Data Collection: From the Data-Understanding, the data needed for modeling is identified. A data-gathering phase may be necessary because some of the data may never have been collected. There may be a need to acquire external data from public databases (such as census or weather data) or proprietary databases (such as credit bureau data). Special security and privacy laws may govern the data and the data-mart may inherit those restrictions. For example, many European data sets are constrained in their use by privacy regulations that are far stricter than those in the United States. Techniques like masking and scrambling may be employed to ease restriction.

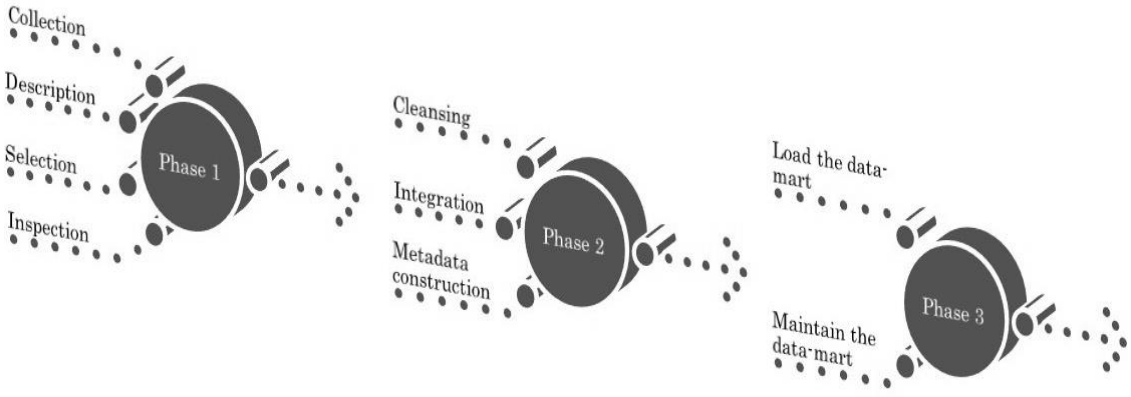


Figure 17. Three Phases and Steps in Building a Data-mart for Analytics

- Data Definition: Each of the attributes and targets shortlisted for modeling must be understood in 'terms of units, data-type (continuous, discrete, nominal), levels, range, frequency, order, coded-meaning (like 9999 means unknown). It is advisable to document so the whole team is on the same page regarding the variables involved.
- Selection: This is not the same as feature selection. While feature selection is specific to the target variable, in this step, two decisions are made. The first is the time-span of the data to extract and second is the capture of all variables that may be relevant to the main business problem at hand. The goal is somewhat opposite to feature-selection in that a broad net is cast to ensure comprehensiveness. It can be detrimental to discover later that a variable needed for modeling or post-analysis is not in the data-mart. The decision of time-span mainly depends on the longest cycles in the system or business-process being modeled. For a HVAC optimization, one-years data is needed to prove seasonal changes are accounted. For propensity-model this may be a product revision/release to the next.
- Inspection: A data quality inspection identifies characteristics of the data that will affect the model quality. There are a number of data quality problems including missing values, incorrect values, inconsistencies, duplicate-values, etc. Most of the inspection and cleaning tools go hand-in-hand. However, the decision still remains if all data is worth cleaning.

Phase 1 forms the planning phase of building the data-mart where most decisions about the data are made and a plan is created that is executed in the next two phases. Phase 2 consists of cleaning, integration and metadata construction.

1. Cleaning: Once a decision is made to clean the data, the simplest is to exclude the row from modeling. In sparse datasets, this may not be feasible. Some common strategies for calculating missing values include using the modal value (for nominal variables), the median (for ordinal variables), or the mean (for continuous variables). A more complex strategy is to build sub-models to predict the missing values. One such novel method used in such contexts is that of surrogate variables. Where values can be inferred from other attributes based on general covariate trends observed. Then two models are built, one that uses the surrogate and one that does not – the superior one is chosen.
2. Integration: The data needed may reside in a single database or in multiple databases sometimes spread geographically. The source databases may be transaction databases used by the operational systems of the company. Other data may be in data warehouses or data marts built for specific purposes. Still other data may reside in a proprietary database belonging to another company such as a credit bureau. Data integration and consolidation combines data from different sources into a single mining database and requires reconciling differences in data values and granularities from the various sources. For example, there are often unit incompatibilities, in data from different countries. While outdoor temperature is needed per building, the weather data may be available

only by zip-code. A subscriber may have moved thus spanning two sales-areas. Improperly reconciled data is a major source of quality problems.

3. Metadata Construction: The information in data description document is the basis for the metadata infrastructure. In essence this is a database about the database itself. It provides information that will be used in the creation of the physical database as well as information that will be used by analysts in understanding the data and building the models. The metadata is updated with details from the cleaning and integration steps.

The final phase is that of loading the data-mart and maintaining it. Having collected, integrated and cleaned the data, it is now necessary to actually load the database itself.

- Loading the data-mart: In most cases the data should be stored in its own database. For large amounts or complex data, flat files are inadequate. Latest advancements in data-storage retrieval and processing like HDFS should be leveraged. However, a balance should be reached between the size requirements and simplicity requirements in making decisions about storage mechanisms. High degree of normalization must be avoided to ensure extraction speed. Cleaning and integration may be done during the loading process itself. Additionally, the loading should be scripted and manual steps should be avoided for rollback and replication ease. Many ETL tools come with basic mathematical operations and abilities to bolt-on scripts that allow other complex manipulation.
- Maintain the data mining database. There are two aspects of maintenance – the data and the database. Often, changes in direction or

pivoting of the data-analytics effort means capturing additional data. When designing the data-mart, updatability and extensibility needs to be factored in. In general, the schema chosen must be flexible and scalable. Once created, the database needs to be backed up periodically; its performance should be monitored; and it may need occasional reorganization to improve performance. For large complex databases, the maintenance may also require the services of information systems professionals.

By building and maintain a data-mart that encompasses the relevant data that is at core as well as periphery of the task at hand allows extensive experimentation at all stages of the data-analytics process as proposed in the previous chapter.

3.3.2 Time Walking

The real-world challenges that affect predictive analytics systems were discussed earlier - temporal-shifts and time-variance have a large impact on the performance of machine-learning algorithms. Evaluations done in single static datasets do not represent the performance to be expected in the field. This section proposes a time-walking simulation technique that incorporates the temporal-shifts (lead and lag), window-size and time-variance (dynamics) nature of real-world applications. Additionally, this technique allows running experiments about other variables affecting the performance of the algorithm in the field; namely, training-window, data-sampling scheme, time-shift sensitivity, test-window sensitivity, robustness, stability, hyper-parameters and so on. The experimental methods themselves are discussed in the next section.

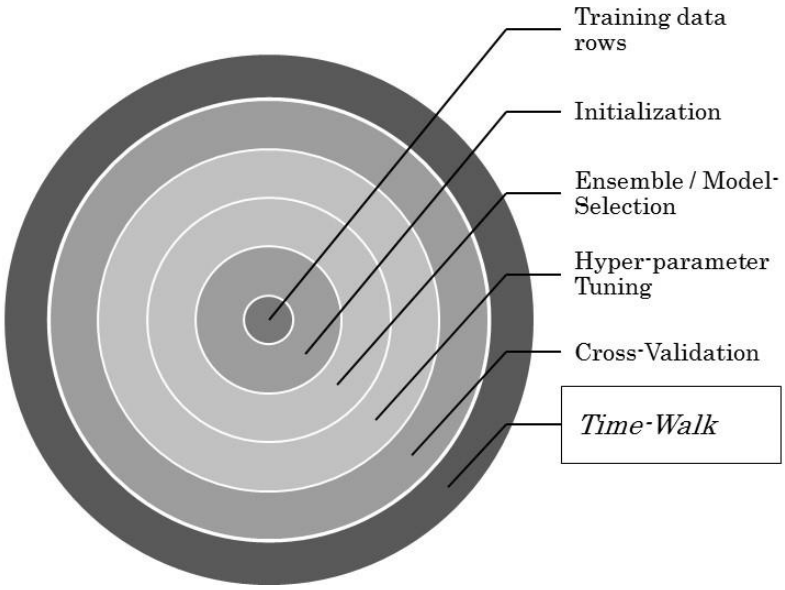


Figure 18. Conceptual Nested-loops Used in Machine Learning

Typically, the development of machine learning algorithms includes several nested loops (conceptual or actual) as shown in Figure 18. The inner-most loop iterates on the rows of data in the training data set. Around are loops for any random initialization, model-selection or ensemble creation, selecting among set of hyper-parameters (tuning), and finally, cross-validation. The process iterates through options usually with intent of picking out a best option. This dissertation proposes adding a loop for time-walking.

The concept of time-walking is illustrated in Figure 19Figure 19. Time Walking Through the Data to Achieve Representative Evaluation. One of the iterations are as a flow-cart to the left. As can be seen it includes all steps of the model building and prediction (test/scoring) process. Training data is extracted from the data-mart, the model is trained and tuned. Then the test data is extracted such that it is time-shifted into the "future" by as much as would be experienced in the

field as described in sections 3.2.1 and 3.2.2. The test-data is scored against the model, the performance metrics are recorded and the whole cycle is repeated.

Note that the time-walk simulation requires large quantities of data across a sufficiently long time-span. However, usually in semi-mission critical systems like manufacturing lines of 24/7 web-services, much is at stake if things do not go as planned. Thus collecting the data and running the time-walk simulation is a justifiable expense of time and effort.

Although the batch training process is assumed in the illustration above, the time-walk simulation technique applies equally well to incremental training scenarios and even time-series data [76]. The main difference being how the training set is constructed and if the model-training starts at each simulation from a blank slate or has priors in terms of the previous model and or distribution.

A key advantage over cross validation is that the test data is truly separate from the training data and hence the assumption of independence is better met. Due

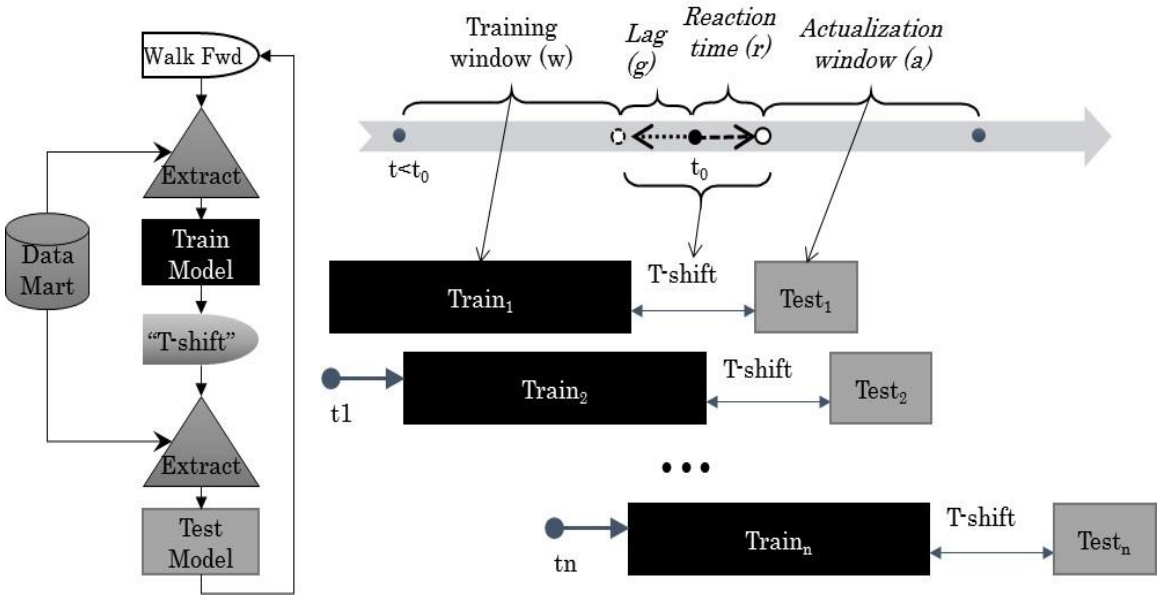


Figure 19. Time Walking Through the Data to Achieve Representative Evaluation

to the time-shift applied, the results from the evaluations are highly representative of reality. As we obtain multiple trials of training and testing each time with data sets that are different, one could compute confidence intervals around the performance metrics if overlap is minimized. The time-walk allows the analyst to experiment with much more than the model-building process itself. For example, one might wish to experiment with the longevity of the model or metrics to use to determine model-staleness for more intelligent refresh. Additionally, the analyst could use the data over time to prove stability of the system to skeptical decision makers. The machine-learning system could be compared to the incumbent systems along various metrics by running them side-by-side. ROI could be established over time to convince potential clients of the benefits of the system. The next section covers how Design of Experiments can be used to get the most from the setup.

3.3.3 Design of Experiments (DOE)

Typically, the machine-learning community evaluates algorithms against variables either one factor at a time (OFAT) or in the all-permutations on grid (APOG) form. In the one factor at a time method, all variables are fixed (typically at nominal value) and the one variable being studied is varied over its range of values,

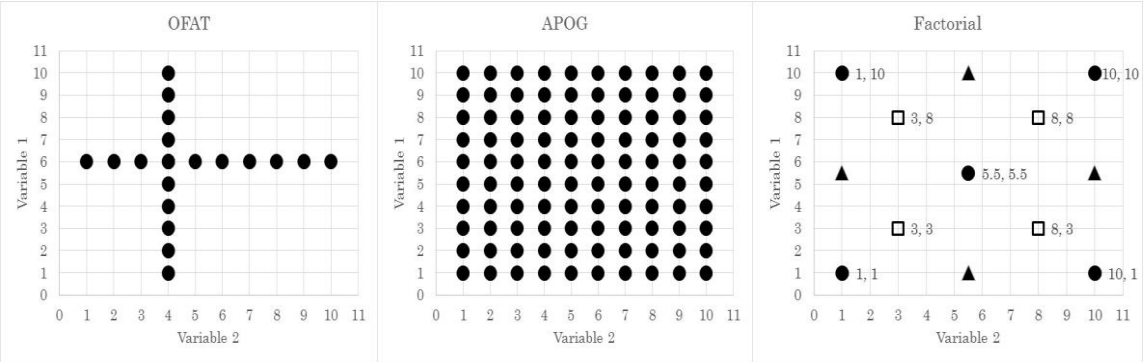


Figure 20. Three Options to Run Experiments: OFAT, APOG and Factorial

typically at least at ten different levels. For example, when studying the performance of neural nets, the learning rate is varied while keeping the momentum is varied followed by keeping the momentum constant and varying the learning-rate. If each is set at 10 different levels, we have 20 readings in all. Continuing with the same example, in the all-permutations on grid method, a 10x10 matrix of all permutations of the 10 levels of 2 variables is formed and 100 readings are obtained.

There are issues with either option. As seen in Figure 20, in the one factor at a time method (OFAT), the search space is sliced at two locations parallel to the axis thus missing out on any data in the rest of the search-space. On the other hand, in the all-permutations on grid method (APOG), one is taking 100 readings to study the effect of just two variables. What would happen if there are 6 variables – we would be need to take a million readings. If one reading took a minute, we would need over two-years to complete the study. Fortunately, we do not have to necessarily pick from either extreme. An entire area of applied statistics called Design of Experiments (DOE) is devoted to the art of characterizing processes. Surely once could also treat this as an optimization problem and apply any number of optimization algorithms. However, if the intent is to understand the effect of variables, DOE is a compelling choice.

Design of Experiments is a methodology of actively collecting data from the process by deliberately varying the process input parameters at certain well-designed combinations. It is a systematic method of collecting data to build a regression model that broadly describes a process or system. The beauty of the method lies in the fact that data can be collected in stages without expending all effort at once. As more is learned about the nature of the system, the direction of

data-collection can be altered. However, it takes careful planning and thought to conduct a well-designed experiment.

If generally smooth behavior can be assumed, there is opportunity for using DOE even if a regression model may not be meaningful. The Factorial DOE design is shown in Figure 20 labeled 'Factorial'. In this design, the experimenter can first start with just 5 readings denoted by the circular markers. Then a statistical test for curvature in behavior is performed. The tests for curvature identify whether nonlinear terms are needed. If they are not, it is unnecessary to collect more-data. If there is curvature, additional readings can be taken at the permutations denoted by the hollow squares and further from at the triangle markers.

With just 15 readings, one can learn a great deal more about the effect of the variables than the 20 readings usually taken with OFAT. The OFAT method also misses on capturing the interaction effects that might be present. Interaction effects are where the effect of one-variable depends on the setting of another – the factor multiplication terms.

Design of experiment techniques have options for fractional-factorial designs that can scale linearly in number of readings needed as number of variables increase. By sacrificing precision on higher-order terms, which are seldom significant, operational efficiency is achieved. Data is collected to determine the terms in the regression model only if the statistical tests indicate that the model is insufficient to describe the system – like in step-wise regression. The details of how all this is achieved is out of scope of this work. One can refer to the following as a starting points [77] pp475, [78] and [79].

With data collected from a DOE, once a satisfactory regression model is achieved, the optimal setting for parameters can also be identified by factor profiling.

3.4 Demonstration

The evaluation framework described will now be demonstrated on the selective die-kill use-case in semiconductor manufacturing from Section 2.6.2: identify die with a high propensity of failing after going through assembly and scrap them beforehand to save cost.

3.4.1 Purpose of Evaluation

The business problem, transformed to the data-analytics problem can be stated as follows:

1. Can we build ML models that are capable of predicting final Pass-Fail result such that, sufficient quantities of fails are correctly identified in order to make the effort feasible? That is, TP > M; where M is a number determined based on die-area, process-flow, volumes/week, average-selling price and other factors.
2. Does the ratio of realized opportunity (true-positive, TP) to overkill (false-positive, FP) put us in the positive ROI region – say greater than 2? The ratio also depends on several factors: product-mix, current yield in that product-line, assembly/test cost, market-segment etc.
3. How do we setup the ML engine so that we can achieve *sustained* ROI?

With a highly scalable tree ensemble algorithm using bagging and boosting [80], the initial experiments with static data sets were promising. As seen in Figure 21, on most runs, the true-positive percentage (of all tested) denoted by dots (F, F) is twice as large as the false-positive percentage (of all tested) denoted by 'X' (P, F). The training set was fixed and different sizes of test-set were used (50k, 100k, 150k and 200k). For each size, the experiment was repeated 19 times.

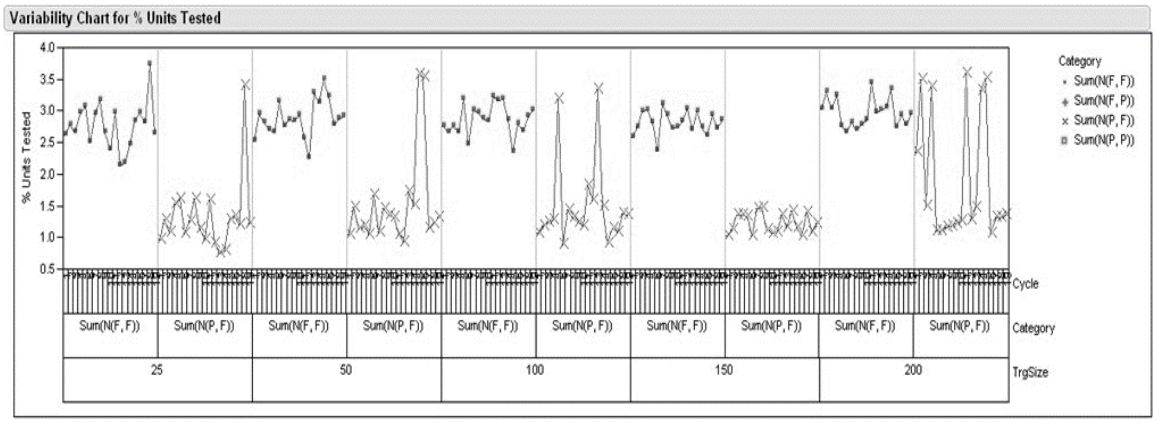


Figure 21. Results with Static Training and Test Sets

One also notices that the 2:1 ratio of TP to FP is not consistent and there are trials where the ratio is 1:1 for the same size of test-set – this was worrisome. The purpose of evaluation here was not necessarily to find one dataset where the algorithm would give the desired results but to show that sustained ROI was possible. Thus one had to demonstrate the performance over time and other process dynamics.

3.4.2 Dynamics

Note from the data preparation phase the team noticed that there are significant temporal shifts and dynamics in the data-stream used to build the model. Firstly, multiple product flavors exist with slightly different manufacturing recipes and/or mix and match of die to substrates as shown in Figure 22 by different shades of gray. As seen, the product volumes ramp across weeks and ultimately ramp-down. Product lifecycles may last anywhere from 1.5years for mobile to 7years in IoT. The point to note is that the flavor mix evolves every-week – new flavors are introduced and some are discontinued. The question for the data-scientist is if models should be created by flavor or should flavor be one of the features in the model. Additionally,

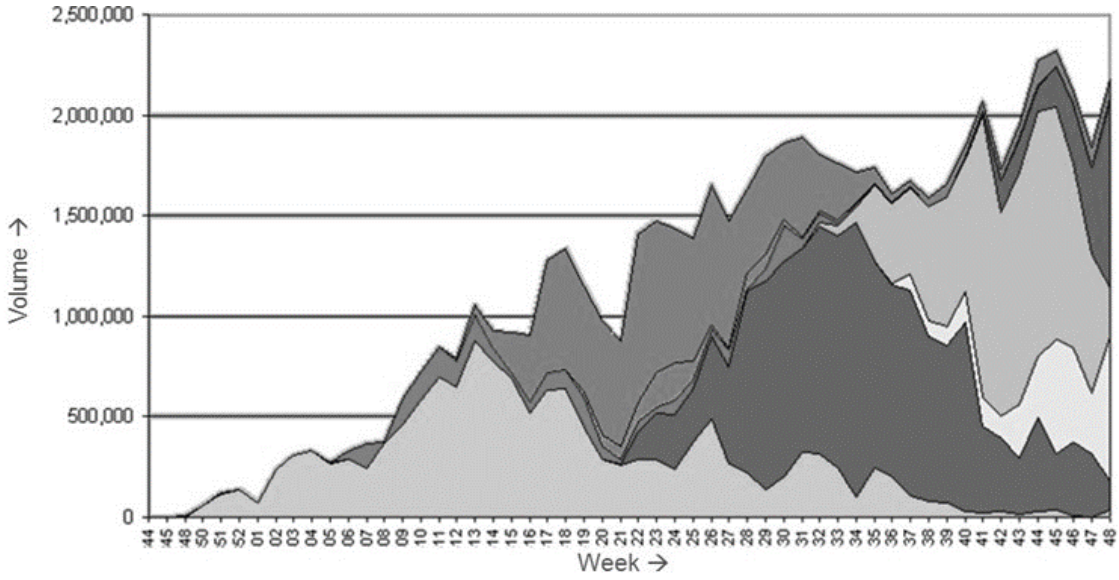


Figure 22. Multiple Flavors of a Single Product in the Factory

the process and test engineers continuously tweak their recipes to eek the last bit of room from the process to maximize yield or reduce cost, hence the process is always changing. As explained in Section 3.2.4, it is not always possible to include every variation as a feature and hence ultimately models will need to be updated or refreshed. Will the ROI be sustainable across a dynamic evolving process?

3.4.3 Temporal Shifts

As mentioned before, semiconductor manufacturing happens in a batch-process with the lot being the batch – a given process-step processes one lot at a time. Each lot at a given step faces a certain queue-time where it is waiting to be processed and the actual processing-time. The sum of these two are known as cycle-time (CT). The delta between the end-time at one step and end-time at another subsequent downstream step is known as throughput time (TPT for short).

For the selective die-kill use-case, the physical removal of the die identified to have a high propensity for failure could happen at two possible locations in the

manufacturing flow: at TRDS (Tape-Reel-Die System) or at CAM (Chip Attach Module). The TRDS step is where the die are picked off the wafer and placed onto a reel. The robot arm could be instructed to skip-over and not the ones with high propensity for failure as identified by the ML algorithm. The CAM module is the one that then picks up the die from the reel and places them onto substrates. Again, the robot arm could be instructed to skip-over and not the ones with high propensity for failure. The difference between the two locations is an in-between inventory step.

The TRDI (Tape-Reel-Die Inventory) is where reels are stored and then shipped to assembly factory as factory planners determine when and what to "kit" based on market demand, factory loading and many other aspects. The material sitting in the TRDI is considered unfinished inventory and has tax implications. Minimizing the inventory has financial implications thus it would be advantageous to remove the failure-prone die before TRDI. Extensive analysis was conducted on the TPT and CT between operations and found to have significant variation.

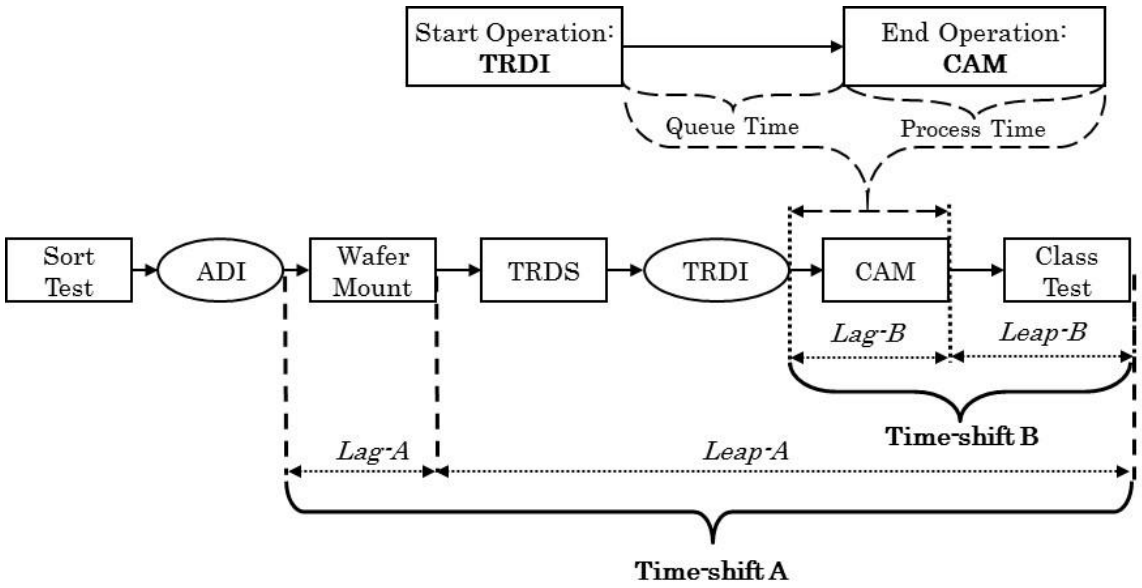


Figure 23. Temporal Leap and Lag and Time-shift Options

The median periods between operations spanned days if not weeks. Although specific times cannot be published due to being sensitive trade-secrets, a discussion on relative terms is still possible. The temporal lag and lead times (as defined in Sections 3.2.1 and 3.2.2) are shown in Figure 23. As shown, *temporal-shift-B* is more than twice as long as *temporal-shift-A* and it could be as high as 4 or more times temporal-shift A. Question is whether the machine- learning model could perform as well with these temporal shifts. To validate this question, the simulation framework described in the next section was used.

3.4.4 Simulation/Evaluation Framework

The overall architecture of the simulation setup is shown in Figure 24. The key components are the data-mart, script-host and experiment recipes. The team chose to create an offline data-mart by one-time extraction, linking and cleaning of relevant data from the Fab, Sort, Assembly, and Test operational databases across the globe. One years' data for a single product line amounted to 20 million units with over 200 features or attributes stored in third normal form using a star-schema.

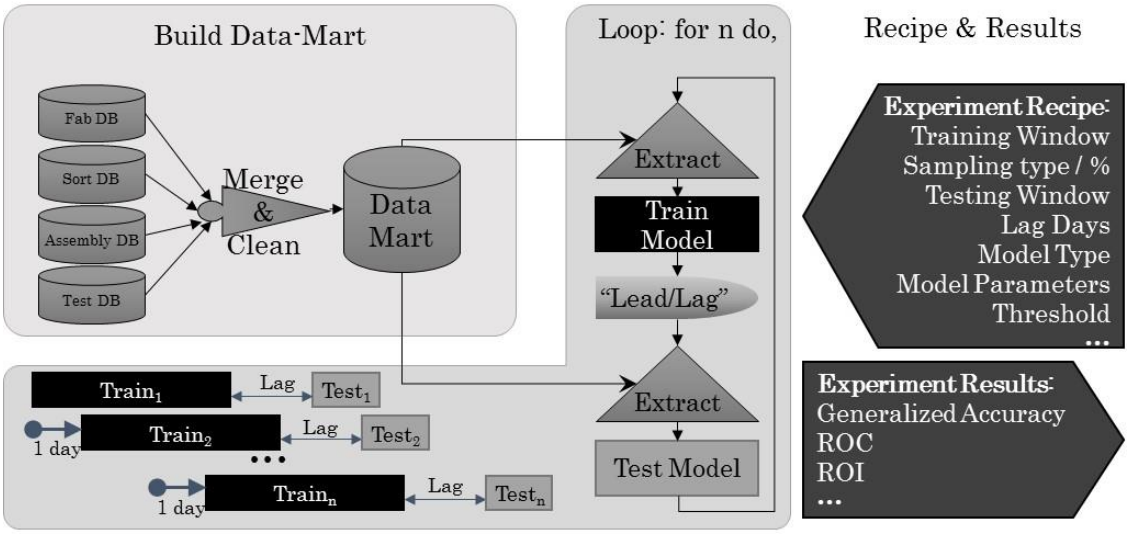


Figure 24. Simulation Platform Setup for Experimentation

The script-host was built to allow setting up of various experiments by simply providing a configuration file. The script flow involved extracting training data from the data-mart with adjustable sampling schemes and rates. Then the desired feature-selection and modeling algorithms are called with adjustable algorithmic parameters. Next, an adjustable temporal lag can be applied to mimic the lag in the manufacturing flow. Finally, the model is applied to the extracted test data with the given lag and adjustable thresholds. Actuals are compared to get true prediction performance. The process is setup to loop and "walk" across time to represent real-world scenarios. The system was designed to be scalable to support a large number of simultaneous and disparate applications using predictive models. The infrastructure later served as a great template for the production predictive analytics system. Preliminary results showed that non-algorithmic variables like the duration of training data, sampling strategy, prediction thresholds mattered greater than algorithmic hyper-parameters.

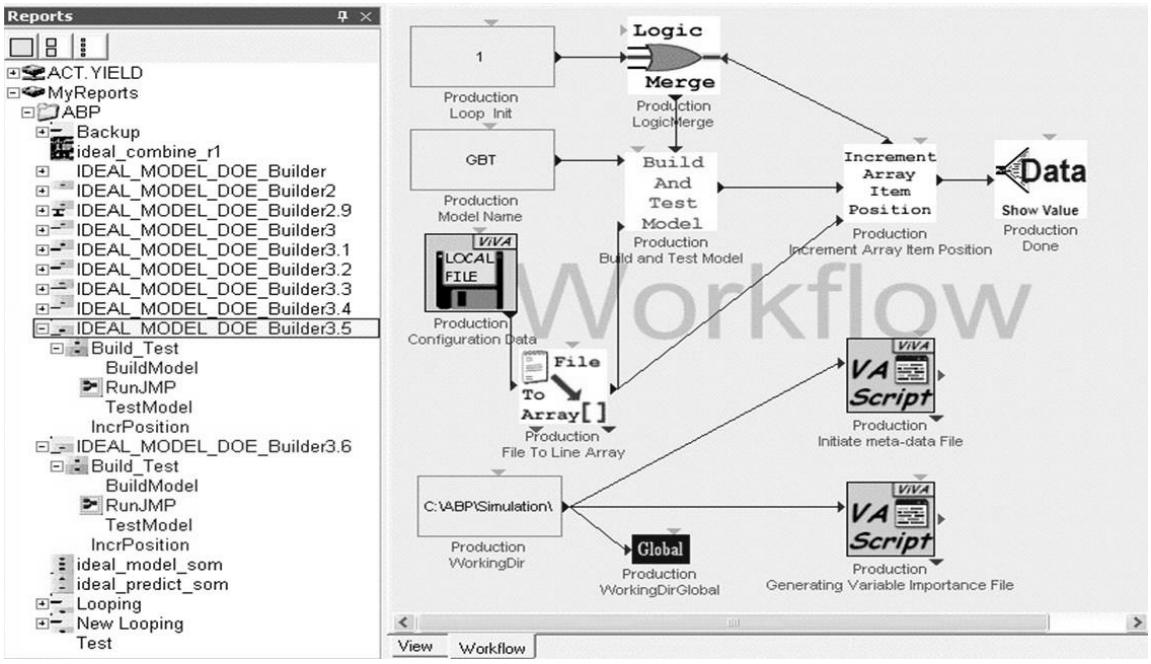


Figure 25. An Example Workflow from the Script-host

3.5 Results and Discussion

Utilizing the infrastructure mentioned in the previous section, several experiments were setup and. The training period was set to two-weeks, prediction period of 11-day and "walk" across 30-days with the temporal-shift set at 0-days. The initial results were promising with the TP/FP ratio close to 2 as shown in Figure 26.

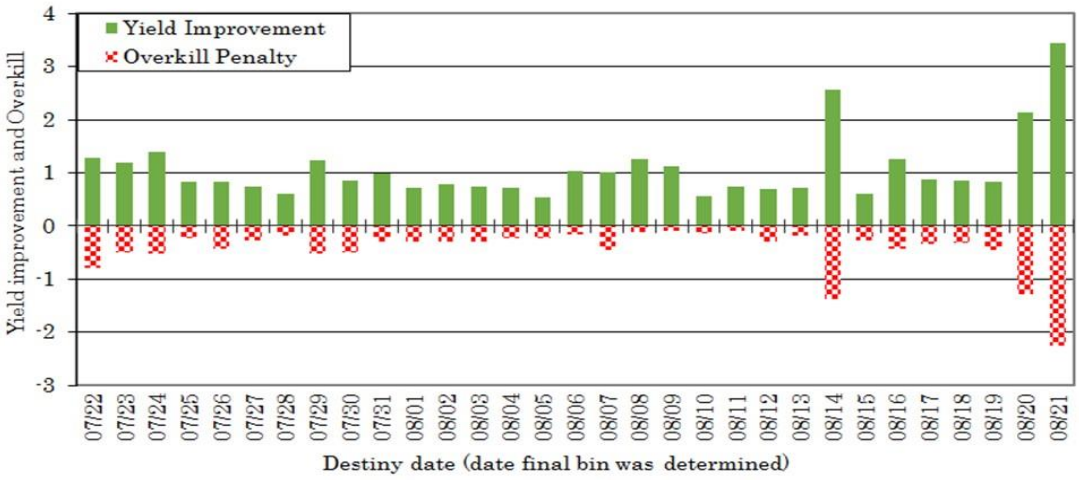


Figure 26. Good Initial Results of TP>FP

However, based on the analysis done in section 3.4.3, it was clear that the temporal-shift was not zero. Thus more experiments were conducted and the temporal shift was set at two different levels. The short 1x corresponding to the option to execute the die-kill at CAM and the longer 4x corresponding to the die-kill if executed at TRDS. Additionally, the period of the simulation was increased to 180 days. The results are shown in Figure 27. As can be seen, the performance ratio is no longer consistent, reversing in places to where the False-Positive is higher than the True-Positive thus contributing negatively to the ROI. Although the result quelled the initial optimism, the learning was still timely. One would much rather know the true performance beforehand than after the ML solution is deployed. Thus avoiding further mistrust in the methods not to mention the financial loss.

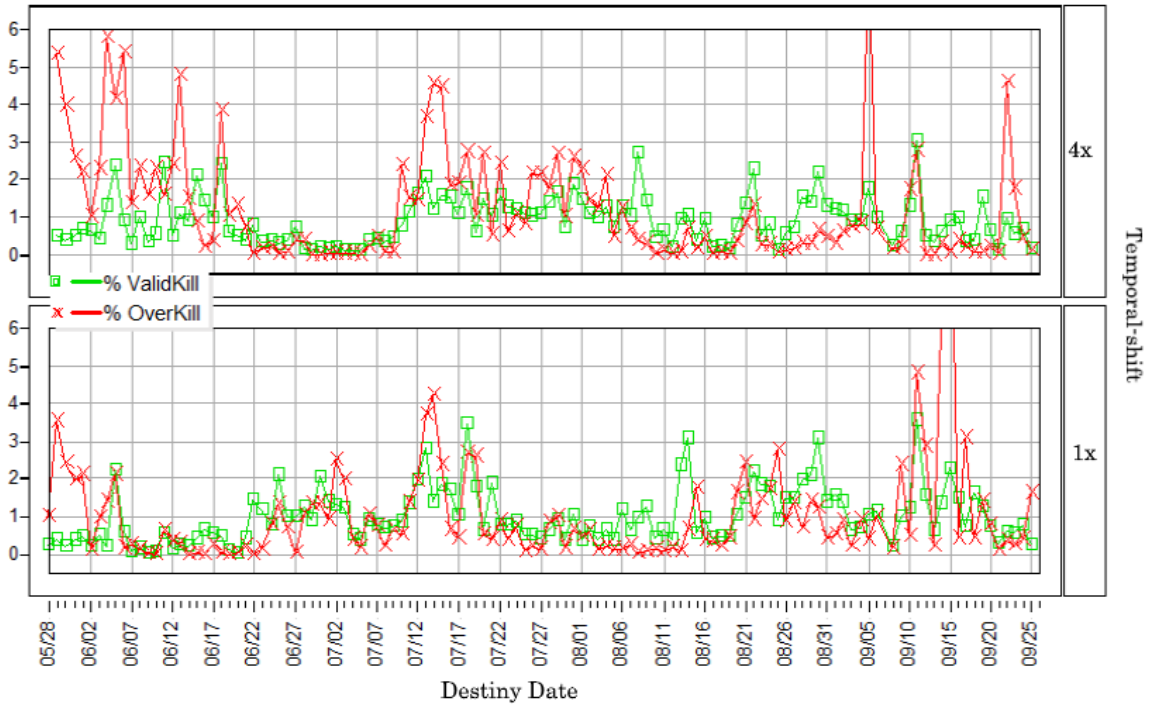


Figure 27. True-Positive and False-Positive for 1x and 4x Temporal-shifts

The inconsistent and undesirable results warranted further analysis to determine the root-cause and what could be done to find a solution. A "good" performance-day (8/30) was chosen to perform reproducibility test. Training duration was fixed to two-weeks, test for 8/30, temporal-shift to 4x. The test was repeated for 36 epochs using random-seeds. The results are shown in Figure 28.

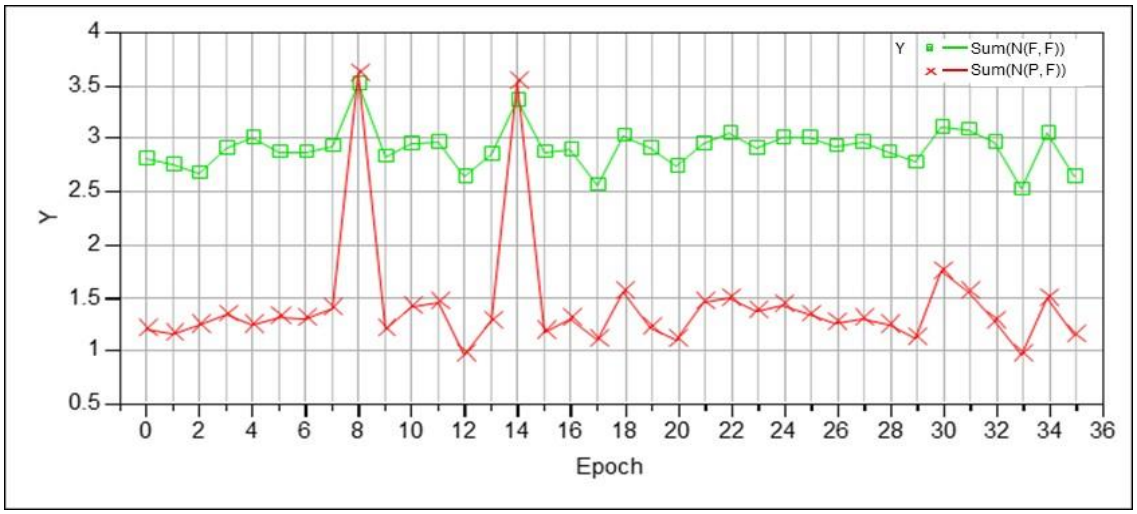


Figure 28. Repeatability on the "Good" Performance Day Data

The results were surprising: although mostly consistent, there were a couple of epochs that had poor performance. This raised a couple of questions: was this behavior due to the stochasticity in the algorithm or the fact that the training set was sampled? Could the other bad performing days be turned around as well if the root-cause was identified? A repeatability test by fixing the training-set revealed that the anomaly was not just due to the algorithm as can be seen from Figure 29.

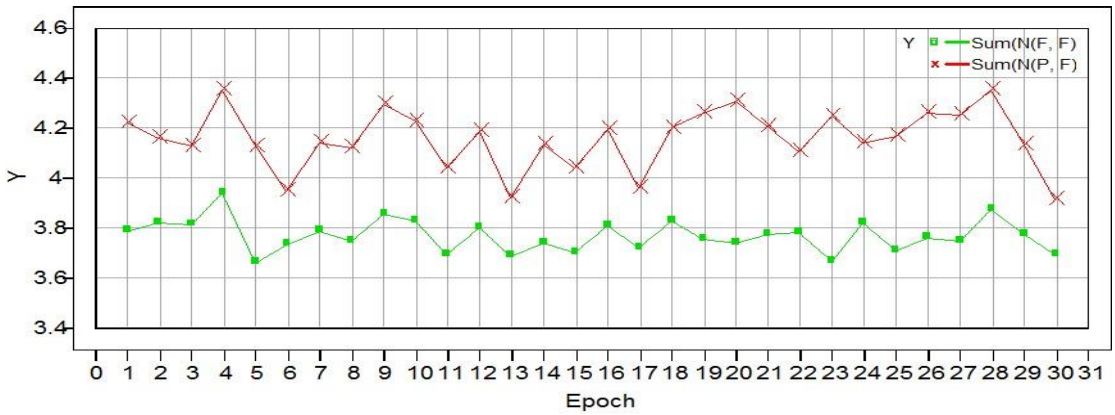
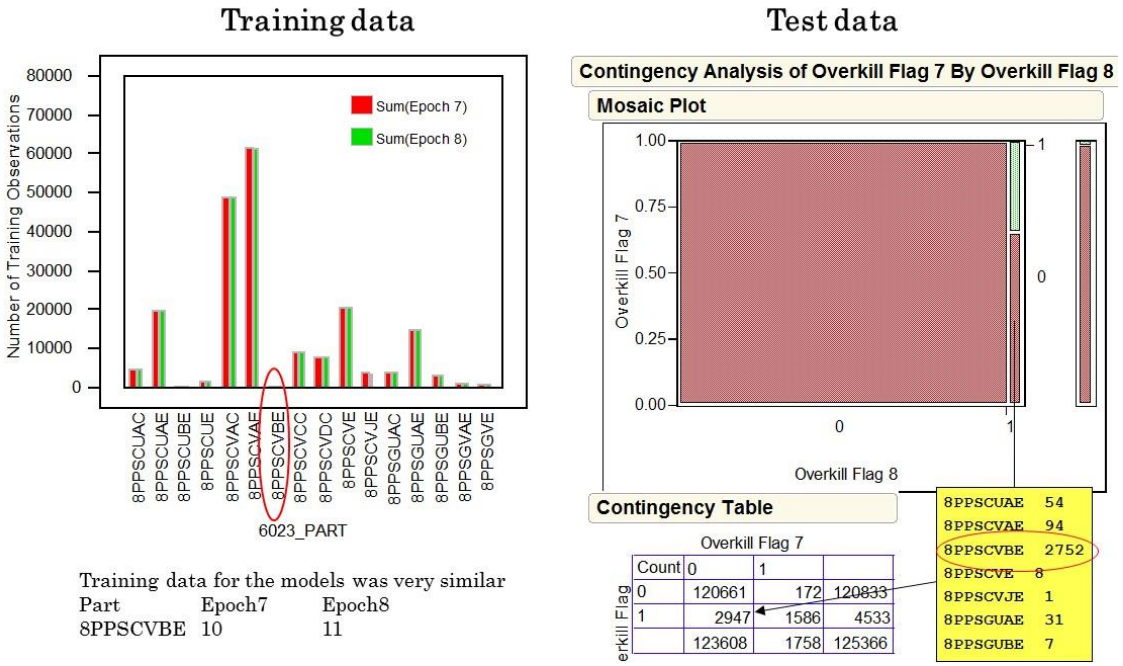


Figure 29. Repeatability Test for the Anomalous Epoch-8

However, the anomaly was not due to the sampling of the training data either. On closer examination of the training sets for epoch 7 and epoch 8, it was discovered that both had about 10 samples (see) of the part that was responsible for 60% of the false-positives. Thus only a handful of samples were available in the training set, whereas, the test-set had a large number of samples of this product flavor. The shortage of training samples of this flavor with the small degree of stochasticity in the algorithm resulted in epoch 7 performing well but epoch 8 being adversely affected. Regardless, it was clear that the instability was due to a part appearing prominently in the test-set that did not have representation in the training-set. The situation as such is unavoidable: the temporal shift is a hard-reality and new-product-flavors will always be introduced.



Training data for the models was very similar

Part	Epoch7	Epoch8
8PPSCVBE	10	11

Figure 30. Unobserved Product Flavor Cause of the Anomalous Behavior

The root cause was confirmed by removing the 6023_PART (part number) as a feature and let the algorithm learn only based on the measured parametric data. When this was done, the anomalous behavior disappeared as seen in Figure 31.

However, removing the part as a feature is not a permanent option. This would cause the performance on other days to deteriorate drastically.

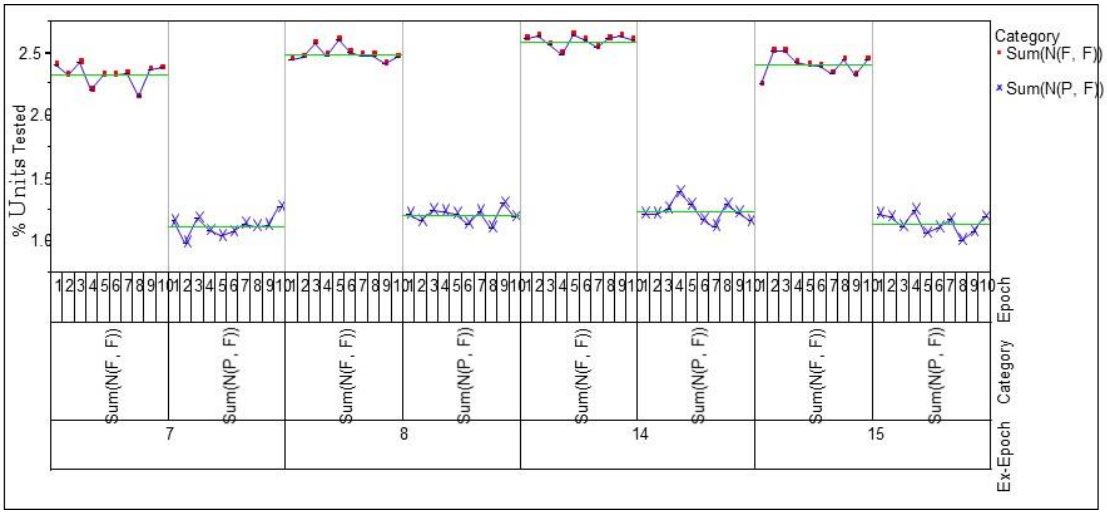


Figure 31. Performance After Removing Part as a Feature

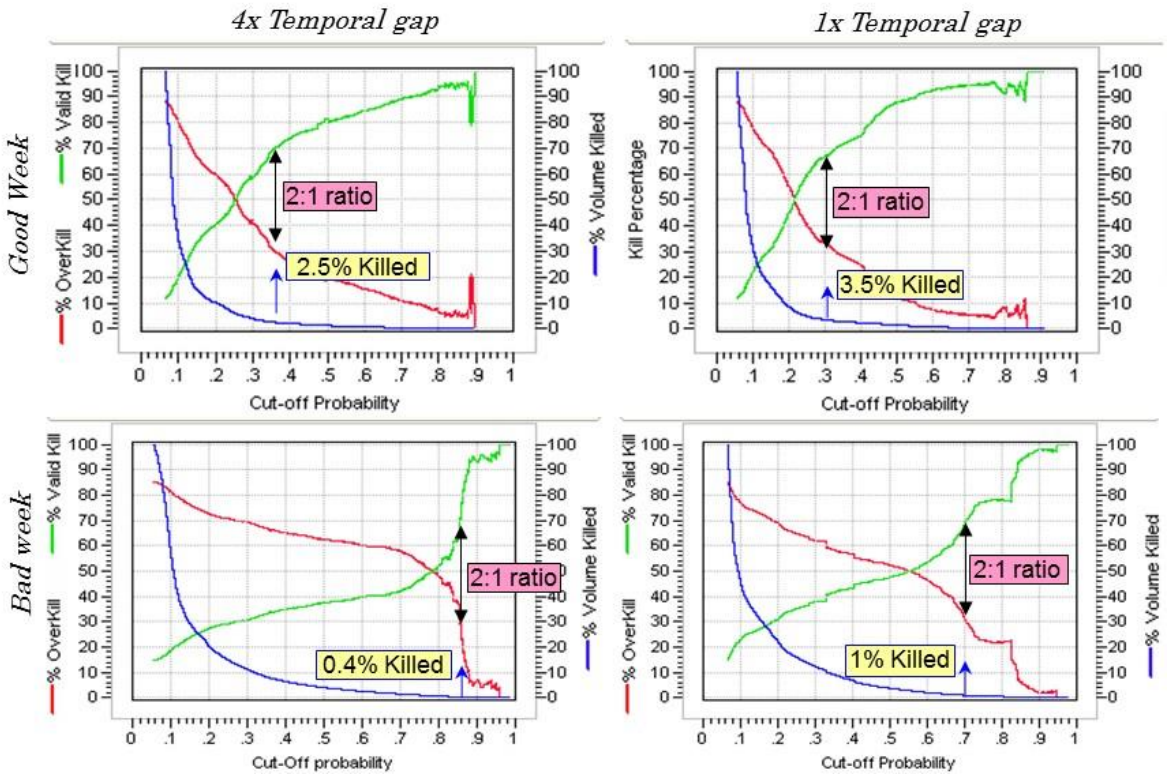


Figure 32. True-positive, False-positive and Kill-rates by Kill-threshold

Several other algorithmic and data-transformation options were tried without much avail. Note a minimum kill-rate *and* a minimum ratio of true-positive to false-positive are required for viability. Figure 32 shows what happens on a "good" week where the curves are convex. During the good week, the training set happens to be highly representative of the test set after taking into account the temporal shift. However, on a bad-week the TP and FP curves become concave – during this week, the challenge is to predict the performance of a test-set that has a new product flavor and who's corresponding temporally-shifted training set does not have the new flavor in sufficiently large quantities yet. Further, the highest ROI for selective-die-kill is when a new flavor is ramping-up and that is when prediction is poorest. Thus a decision was made to move on to a different use-case. As will be seen in the next chapter, the evaluation infrastructure aided a fast pivot.

CHAPTER 4

DEPLOYMENT FOUR-E (DFE)

In practical applications it is often the data and human issues which ultimately dictate success of failure of a project rather than algorithmic and model issues [81]. Even after thorough evaluation that is representative of the field, one needs to convince the decision makers that the solution is both feasible, reliable and sustainable before it can be deployed. Considerations like privacy, security, maintainability have a veto on these decisions. Thus to ensure success it is critically important to focus on these aspects of a data-analytics project. This illustrates the problem, scans the current state of art and proposes a high-level framework to navigate this often ambiguous last mile.

4.1 Problem and Current-state

Since the dawn of data-analytics, the research community has largely focused on inventing powerful and efficient algorithms to extract knowledge from data albeit limited by available data. Limited data has led to techniques around bootstrapping and cross-validation. Around the turn of the century, the focus shifted from data cleaning and pre-processing to the very front-end of understanding the business problem – this allowed much progress. However, several projects have been abandoned either just before deployment or even after 1st deployment due to evaluation and deployment issues. Very little is available in literature in terms of deployment of analytics except in conference presentations [82], and online publications [83]. Today, with abundance of distributed storage, processing and general availability of data, the focus needs to be turned to evaluation and deployment so that the promise of data-science does not remain a promise.

Table 7. Contrasts Between Analytics in Research Versus Industry

LAB / RESEARCH	FACTORY / INDUSTRY
Knowledge extraction	Value extraction
Reproducible results	Reliable performance
Isolate theoretically important ideas	Isolate high impact levers
Novelty highly valued	Diligence highly valued
Generalizable results valued	Precise and consistent results valued
Question driven	Metric-driven
Focus on depth	Focus on transparency
Interactive, flexible	Automated, controllable
Fixed (stationary) data	Fluid (non-stationary) data
Output is a research publication	Output is customer-facing decisions
Constrained by shortage of data	Constrained by, budget, time, legality, privacy, compatibility, brand etc.

The lack of attention to the latter steps of analytics in the research domain could be traced to the differences in nature of analytics in research versus in the industry [23], [84] as shown in Table 7. The goal of research is knowledge extraction to isolate important ideas that can be published whereas the goal of industry is value extraction by discovering high-impact levers that can lead to profitable decisions. Research is driven by open questions that are explored at depth. Hence systems need to be interactive and flexible. Whereas the industry runs by transparent metrics and KPIs. Research places a high value on reproducible results that are novel and generalizable whereas industry desires precise and consistent results achieved through diligence. Lastly, research focusing on stationary data whereas industry has to deal with fluid evolving data in addition to budget, time, legality, privacy and compatibility that have veto-power over projects.

4.2 Real-World Challenges

As mentioned at the start of the chapter, the go/no-go decision to deploy ML in the industry falls upon a decision maker – usually a factory-manager, general-manager or vice-president. Let us examine the key incentives and deterrents for adoption of machine-learning in industrial environments from that decision-maker's perspective. The summary of the factors based on [85] and supplemented by the authors contributions is shown in Figure 33.

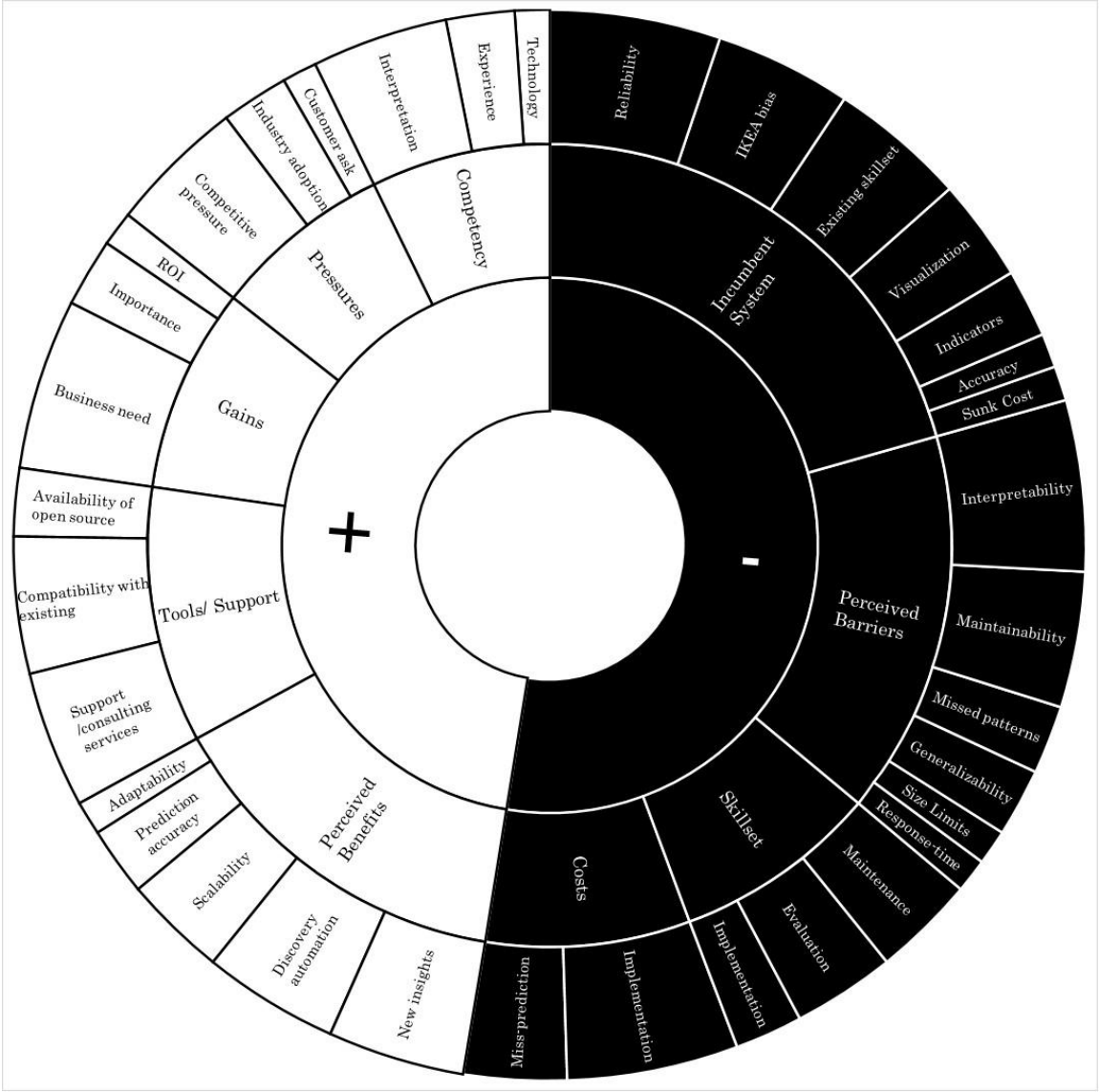


Figure 33. Key Incentives (+) and Deterrents (-) for Adoption of ML

In Figure 33, each of the factors is empirically weighed from 1 through 5. As can be seen, the odds are stacked against adoption of machine-learning analytics in the field unless one of the factors takes on additional weight due to the situation at hand and tips the balance in favor of the adoption as opposed to not. It is important to recognize the perceived barriers to machine-learning are not necessarily the majority of the deterrent, rather primary deterrent is the incumbent system.

Incumbent system is the methodology, infrastructure, teams and terminologies that currently serve the business purpose that the machine-learning system is proposed to replace. Firstly, as the incumbent system is well established, most of the flaws have been either fixed or workarounds are in place making the system effectively reliable irrespective of its actual reliability. However, the new system is expected to be highly reliable out of the box. Secondly, it is important to recognize that incumbent system was built by a team that has been serving the business for a while and hence has earned the trust of the leaders. Hence, leadership seeks their recommendation on adopting anything new. There is a significant cognitive bias called *IKEA Effect* that deters a positive recommendation – the IKEA Effect is when we attach greater value to something we make than the same product built by others. This is also known as the *NIH (Not Invent here)* syndrome. Additionally, adopting machine-learning also means re-skilling the operational teams to be able to sustain and upkeep. The learning-curve is steep and could take months to execute even if there is willingness. Most of the current systems have a set of checks and balances wrapped in indicators and visualizations that bolster the leadership's confidence in the system. The closer the ML system is to being black-box, less it will be trusted.

Perceived barriers to machine learning start with interpretability being on the top. Whereas, rules and correlations are easier to interpret, confusion matrices and F-measures are not. This sword is double-edged: on one hand some measures are hard to interpret, measures like accuracy give a false sense of goodness and result in high expectations which are then not realized in the field. As covered in the previous section, maintainability is a big concern from the skillset perspective but also from a technical perspective where if it was such an involved process to build the models in the first-place, how could they be kept up to date. It is well known that unlike human experts, algorithms do not have peripheral vision. They are limited by the data they are exposed to. Thus, there will be patterns that are missed by the algorithm. Unless the system is designed to have a watchdog, this could case thee solution to lose credibility.

Skillset is a big concern in any machine-learning deployment given that the chief consulting data scientists does not stick around much longer after launch. Several have highlighted how this is a hard-to-find skill to find and retain especially an intersection of analytics, programming and domain knowledge [1]. As seen in the previous sections, skillset is needed to maintain the system and upkeep models. However, the upkeep could actually involve building new models or updating the existing ones. Hence the skillset needs to encompass ability to implement and evaluate machine-learning models as well.

Costs can be bucketed mainly into the actual implementation costs and the cost of misclassification of missed prediction. While the implementation costs tend to be non-recurring, they are less of a concern as opposed to recurring costs of misclassification or missed patterns which need to be controlled.

Fortunately, there are actionable incentives to adopt machine-learning in the industry. By careful planning and design for deployment, one can leverage these incentives to overcome the deterrents and tip the balance in favor of adoption – it goes without mention that the deployment team should ensure that deployment will in fact benefit the organization and people at large.

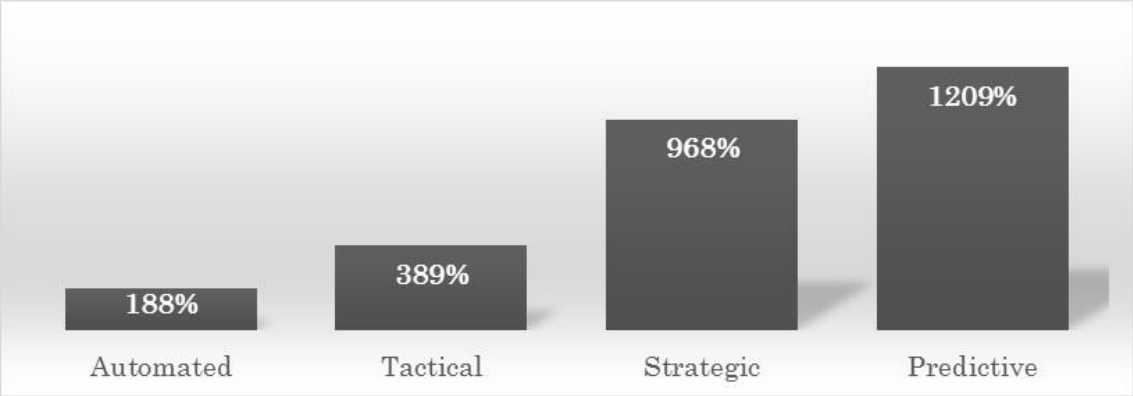


Figure 34. High Return on Investment (ROI) for Predictive Analytics

Pressures and gains of adopting machine-learning and data-analytics at large are increasing. In addition to customer demands and the general industry moving in that direction, many studies have been published on the advantages of adopting analytics by respected consulting firms [86], [87], [4], [9]. Based on a study of 60 deployments over 3 years, an average ROI of 1200% has been claimed for organizations that successfully deploy predictive analytics [86] as shown in Figure 34. Although ROI claims can be imprecise, most agree that the impetus is greatest when predictive analytics provides a clear competitive advantage that is critical to the business in the short-term. That is, it addresses a current urgent business need.

Perceived benefits are the major incentive for adoption of machine-learning unlike the deterrents where other factors take over. Hence it is highly important that the right expectations are set and these benefits are thoroughly evaluated

before being promised. Contrary to popular belief it is not the higher accuracy that attracts businesses to machine-learning systems, it is the promise of new insights that enable new avenues of business and value for example, expanding customer base, offering new products and services or gaining a competitive advantage. Most business leaders are interested in automating the discovery process. They turn to machine-learning when the current system is either too slow or is not scalable to either meet the demands of current business or is not giving the expected growth.

Tools/Support are critical to adoption of machine-learning. As most adoptions are driven by urgent business needs, tools and consulting support facilitate a faster ramp before leadership loses patience. Compatibility with existing systems and processes is also of concern. Thus, standardized tools and processes especially those supported by a wider open-source community as well as a responsible support organization aid integration as well as maintainability.

Competency even if restricted to an in-house core team goes a long way in easing the adoption of machine-learning. The team should be firstly capable of interpreting and translating the results to leadership. Additionally, they should possess experience and be well versed with current technology trends. Lastly, they must be capable of teaching and upskilling their colleagues on data-science concepts. The higher the comfort level among the technology teams around machine-learning the easier deployment becomes.

Note that the veto factors (security, privacy, legality) have not been included in the above discussion as these are non-negotiables that any system will have to meet before it can be deployed in the field. Thus one cannot emphasize enough the importance early of planning for evaluations along these aspects.

4.3 Proposed Deployment Framework

From the above analysis key patterns emerge. There are non-negotiables like security and privacy. Following them are concerns about maintainability and skillsets required to so. Then, doubts around interpretability, reliability and realizing the ROI. Based on these themes, a four step framework shown in Figure 35 is proposed which when followed should result in higher success. The first step is to *ensure* that the "veto" aspects are definitely taken care of with high diligence so the deployment does not trip on them later. The second step systematically *evaluates* and attempts to prove aspects of high importance that ensure that the promised incentives for the machine-learning solutions are met. The third step educates and involves the incumbent teams so that key deterrents are successfully addressed. The fourth and last step *establishes* processes so that the deployment is sustainable even after the implementation team has disbanded. Although the process covers major aspects, the influencing factors from Figure 33 must be incorporated as needed.

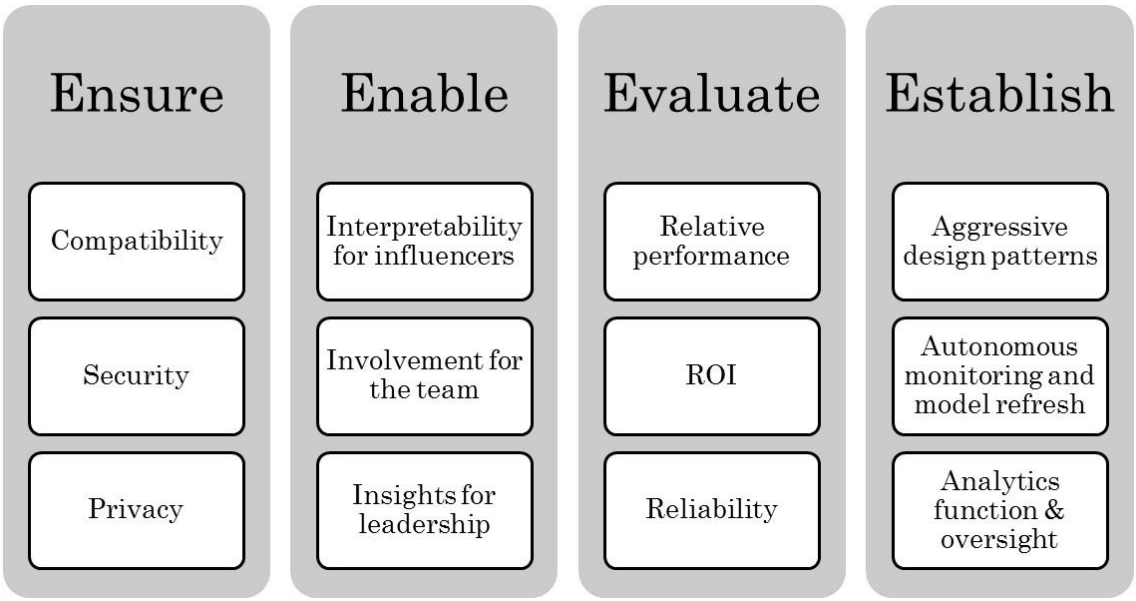


Figure 35. Proposed Steps for Sustainable Deployment

4.3.1 Ensure

There are some data scientists that expect that they are bringing cutting-edge technology to the organization and they will be welcomed with open-arms. However, the reality is quite different. The situation is more attune to a new organ being transplanted into the organization and it is the responsibility of the initiator to ensure it is not rejected. Thus ensuring compatibility security and privacy of data does become the responsibility of the deployment team.

4.3.1.1 Compatibility

Compatibility of the machine-learning system must be ensured along three directions: hardware, software and processes. *Hardware changes* might be needed including additional system resources to store and process data, changes to manufacturing equipment and much more. Some changes are not straightforward and would need significant development effort and ramp time. Furthermore, they could involve capital investment and hence need to be carefully planned sometimes much before the deployment stage. Similarly, *software changes* could involve developing large quantities of glue code. It may be surprising to the academic community that only a tiny fraction of the code in many machine learning systems is actually doing "machine learning" - a mature system might end up being (at most) 5% machine learning code and (at least) 95% glue code [48]. The amount of glue-code could be reduced by use of COTS (commercial of the shelf) systems. However now their compatibility with existing systems needs to be ensured [88]. The ultimate goal of the deployment is to provide interoperability and compatibility between the different software systems and platforms used throughout the process. Integrated

and interoperable models would serve the end user in automating, work with data-analytics systems [18]. More is covered in the maintainability section of this chapter.

4.3.1.2 Security

Security of software systems is another non-negotiable that for all systems and hence machine-learning component is no exception. Most environments into which machine-learning systems are deployed are highly sensitive to breaches. From credit-card fraud detection to manufacturing automation, most of the data is personal, proprietary, highly-confidential or trade-secret. For example, in the simple home HVAC optimization, building occupancy could be a sensitive piece of data that needs to be secured. Recent examples have demonstrated that data breaches can expose not only personal consumer information and confidential corporate information but even national security secrets [4]. With serious breaches on the rise, addressing data security through technological and policy tools is essential. While the technologies available today can safely house information with a variety of security controls in a single system, these policies force special data handling considerations including limited retention periods and data access [6]. In the literature, the following security and privacy requirements are described [89]:

- Resilience to attacks: The system has to avoid single points of failure and should adjust itself to node failures.
- Data authentication: As a principle, retrieved address and object information must be authenticated.
- Access control: Information providers must be able to implement access control on the data provided.

- Client privacy: Measures need to be taken that only the information provider is able to infer from observing the use of the lookup system related to a customer; at least, inference should be very hard to conduct.

Machine learning security is a relatively new field and hence COTS are not readily available. Some high-level views are outlined here. When a learning algorithm succeeds in adversarial conditions, it is an algorithm for secure learning. The crucial task is to evaluate the resilience of learning systems and determine whether they satisfy requirements for secure learning. A framework for analyzing attacks against machine learning systems was proposed in [90]:

1. They may be *Causative* in their influence over the training process, or they may be *Exploratory* and take place post-training
2. They may be attacks on *Integrity* aimed at false negatives (allowing hostile input into a system) or they may be attacks on *Availability* aimed at false positives (preventing benign input from entering a system)
3. They may be *Targeted* at a particular input or they may be *Indiscriminate* in which inputs fail.

Each of these dimensions operates independently, so we have at least eight distinct classes of attacks on machine learning systems [90].

Investigation of security properties of machine learning has to focus on quantitative analysis of attacker's resources needed to subvert a learning process. A four step framework has been recommended that enables one to quantitatively analyze and compare existing algorithms under identical conditions [91]:

1. Axiomatic formalization of the learning and attack processes
2. Specification of attacker's constraints

3. Investigation of an optimal attack policy
4. Bounding of attacker's gain under an optimal policy.

An empirical comparison of basic algorithms can be found in [92].

4.3.1.3 Privacy

Privacy is an issue whose importance, particularly to consumers, is growing as the value of analytics becomes more apparent. However, there is more to privacy than providing security for personal information. Personal data such as health and financial records are often those that can offer the most significant human benefits, such as helping to pinpoint the right medical treatment or the most appropriate financial product. However, consumers also view these categories of data as being the most sensitive [4]. Brands reinforce negative stereotypes when they use data in ways that offend the dignity and privacy of consumers. Recently, a big-name retailer used data analytics to send discount offers for pregnancy-products to a 16-year-old customer even before she had disclosed this to her own family. Intruding on privacy was not the intent, however, the incident caused a huge PR black eye for company's data analytics efforts [9]. Thus we must examine the trade-offs between privacy and utility (or accuracy/precision). Most methods for privacy preservation use some form of transformation on the data resulting in granularity reduction which in turn reduces effectiveness of mining algorithms. This is the natural trade-off between information loss and privacy. Some examples of such techniques are as follows [93]:

- Randomization: add sufficiently large quantities of noise so individual record values cannot be recovered. Algorithms then work on distributions.
- k-anonymity model and l-diversity: reduce the granularity of data representation by generalization and suppression

- Distributed privacy preservation: entities share partial information with the use of a variety of protocols that can still be used to create models
- Downgrading application effectiveness: here precision is deliberately downgraded to avoid situations like one mentioned above.

4.3.2 Enable

This step focuses on three aspects of the organization that are key to acceptance of the machine-learning solution. Most organizations have three layers – those who store the data and generate reports (teams), those who summarize the reports into key takeaways (influencers) and those who decide (leaders). Each group has its apprehensions, needs and priorities. Note that these three function will exist even with a high level of automation because those responsible do not like to fly blind. As noted earlier, leadership looks for new insights, the influencers look for interpretability and teams want to continue to be relevant.

4.3.2.1 Involvement of Teams

Typically, there is a team that keeps the incumbent system running - it is important to involve them in ML deployment due to several reasons. The top one being to avoid resistance for the new machine-learning system. Additionally, with team's involvement, several tasks become smoother and faster. Data extraction and setup of the simulation system could be done faster with teams that know the current "data piping". A lot of tribal-knowledge exists that could be used to clean data and transform it. The team could also help provide assistance for integration of the machine-learning system during deployment and test its reliability with field trials. Involving the team uses the IKEA bias for the deployment of the ML system

rather than against it. Successful deployment and sustenance of analytics system is as much a cultural transformation as it is a technical transformation. Over 62 percent of the managers responding indicated that organizational and cultural factors were the greatest barriers to ROI on analytics. For an analytic capability to really succeed, the entire organization needs to value data-based analysis and decision making [94]. Hence it is a key advantage to have the teams be participants and advocates rather than skeptics.

For the teams to get involved, they first need to be trained on machine-learning to startup or build skills. Although several online and other resources for learning ML have emerged, there is high value in the ML project team conducting training or information sessions. Focused interaction with the team in a learning setting builds relationships, sparks discussions and ideas that can be highly valuable. It is also a great means for the ML team to learn about the domain.

4.3.2.2 Interpretability

Interpretability of machine learning has been debated for as long as the field has existed. Dashboards and visualization are critically important for the acceptance of the model by the business. The influencers, those who summarize the key messages for leadership and have a big say in recommending the go/no-go decision, are less concerned with the debate and more concerned with ensuring their recommendations are based in fact and valid. Interpretability of algorithms becomes much more attainable if one shifts the paradigm from how a result was obtained (algorithm details) to why a certain outcome resulted (data characteristics) [82]. For example, while the actual results of the algorithm could be based on an ensemble like random-forest, one could build a decision tree that is representative of the key

variables in play. Showing the model-level feature importance pareto-chart is sometimes sufficient to establish trust in the algorithm. The training and education mentioned previously also applies to the influencers especially if they are not familiar with machine-learning. In analytical aspiring companies, analytical models often are reported on by a very technical model report, at the birth of the model in a non-repeatable format. In more mature analytical practice, the modeling data is used for insight creation is a repeatable way.

4.3.2.3 Insights

Insights for the leadership need a deeper kind of analysis especially if the expectation is to produce novel perspectives and relationships that have not been explored before. Two types of reporting are desired: analytic reporting and operational reporting. Analytical reporting refers to any reporting on data where the outcome (of the analytical model) has already been observed. This data can then be used to understand the performance of the model and the evolution of performance over time. Creating structural analytic performance reports also pave the way for structural proper testing using control groups.

Operational reporting refers to any reporting on the data where the outcome has not yet been observed. This data can be used to understand what the model predicts for the future in an aggregated sense and is used for monitoring purposes. For both types of reporting, insights are typically created by looking at behavior of subgroups as qualified by the model. By creating a structural reporting facility for the insights, it allows deeper insight in changing patterns that can be used by business users.

This is different from the task of modeling and scoring for an inline automated application. The output is a set of visualizations and dashboards that provide a clear view on the model effectivity and provide business usable insights. To enable the generation of such insights, one needs to establish data-analytics as a function and not just a one-time project. In addition to ROI the recommendation is to propose to establish a group to provide insights. More is covered on this aspect in Section 4.3.4.3.

4.3.3 Evaluate

The more diligence the deployment team applies to evaluation, the smoother the deployment will be. The last explored evaluation comprehensively while covering the temporal and dynamic factors. Establishing a simulation infrastructure was recommended. This infrastructure can now be used to collect data that will be used to convince the decision makers that the machine-learning based system is ready for deployment. The three key vectors for this evaluation are performance relative to incumbent system, ascertaining ROI and assessing the system level reliability of the solution. Thus putting some of the deterrents to rest.

4.3.3.1 Relative Performance

Relative performance assessment with respect to the incumbent system is expected. As can be seen from the perceived benefits of deploying ML solutions, discovery automation and scalability are highly desired. By demonstrating automated model-building, validation and scoring, through the simulation infrastructure mentioned earlier, one can demonstrate high degree of automation. Thus maintainability without a high cost burden is achieved.

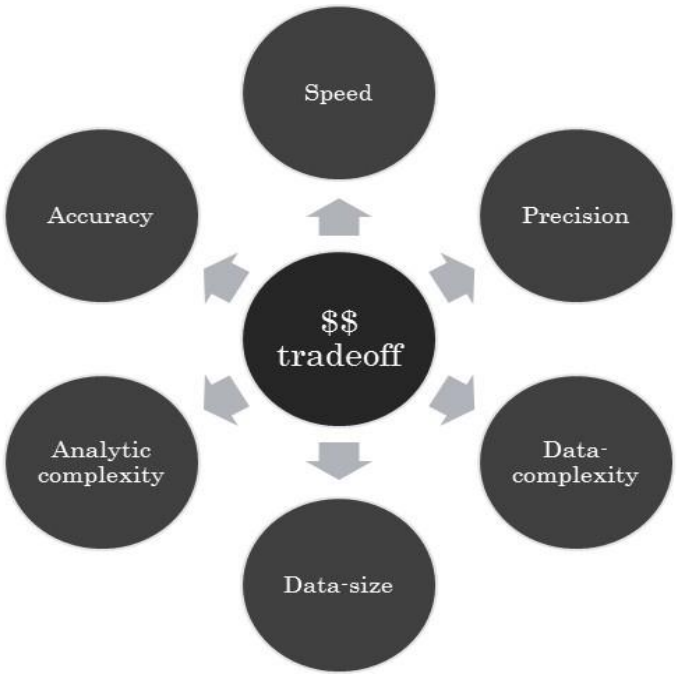


Figure 36. Trade-off in Machine Learning Deployments

The key elements of scalability can be identified as data size, data-complexity, analytic-complexity, speed, accuracy & precision. Balancing these dimensions is usually a zero sum game - an analytic solution is unlikely to simultaneously exhibit all five dimensions, but instead must make trades between them [6]. For example, there might be a trade-off between speed and accuracy – speed of object-detection for avoidance maneuvers in a self-driven vehicle versus identifying what it is. Another example is including 100s of in-home sensor measurements and weather features to predict room-temperature whereas human sensitivity being +/- 2C of that can be achieved by 5 features. In most industrial applications, there are financial implication of this tradeoff; however, all costs are not necessarily known up-front. The best approach here it to identify the knobs and levers in the algorithm to be able to manipulate the behavior so as to tune in the desired operating region using the simulation platform recommended earlier.

4.3.3.2 Return on Investment

Return on investment is the immediate tangible that drives industrial projects. Although not always easy to compute, have a crisp definition up-front of how this will be computed and proved keeps all stakeholders on the same-page. As in Lean Six Sigma Projects, it is advisable to start the project with a projected ROI with aid of a finance analyst and during deployment validate it using the simulation platform. This ensures that the business problem one began with is being addressed.

Table 8. Variable Costs of Machine Learning Algorithms

Cost of	*Due to*
☐ **misclassification errors**	misses & false-positives
☐ **test**	associated with obtaining test-data
☐ **teacher**	associated with labeling data
☐ **intervention / actuation**	applying the prediction back into the process
☐ **unintended consequences**	applying the errors back into the process
☐ **computation**	complexity in terms of time / resources
☐ **human-interaction**	personnel for tuning and oversight

Both the benefits and costs of implementing machine-learning can be categorized as being either one-time (also known as non-recurring) and sustained (recurring). A comprehensive methodology similar to COCOMO or SLIM in software development called DMCoMo (DataMining Cost Model) has been proposed to compute the cost of a data-mining project itself [46]. The model incorporates the trade-offs mentioned above. Whereas the one-time costs are easier to attach

quantities to, the sustained costs and benefits involve some degree of estimation. A taxonomy of classification costs has been proposed that can serve as a checklist to identify these costs and include in the computation [95] – see Table 8.

Once costs are estimated, a proper experiment needs to be set up to measure relative benefits: the analytical model is applied to new data over time in a simulation environment and the outcomes are measured in such a way that the result can be made financial. If the ROI is positive enough, the business will be convinced that they can trust the models; the models are proven to generalize well over time, and a decision can be made if the model should be deployed. Topics of discussion are around the setup of the experiment, control groups, measuring the model effectiveness, computation of the ROI and the success criteria.

4.3.3.3 Reliability

Reliability is a term that is used to mean many things. The common understanding is that of robustness of the algorithm to noise, perturbations, missing values etc. While these are important in the development phase for algorithm selection and tuning, during development, stakeholders are concerned about the system level reliability. What is the propensity for the system to break-down? What happens if predictions are low-confidence? Is there a kill-switch on the predictions?

Reliability is usually a strength of the incumbent systems and without much field-time it is unlikely that stakeholders will trust the system. Hence it falls upon the deployment team to analyze the potential failing points of the system, their impact, probability, detection and/or mitigation. FMEDA (Failure Mode Effect Detection and Analysis) is an industry standard methodology use for this purpose.

Each failure mode is identified through a join brainstorming exercise with key engineering personnel. Then the owner of each area does a deep analysis on the probability of the failure event and the impact (preferably in $$) if it were to realize.

The probability and impact scores are multiplied to obtain a risk-score and sorted in descending order to identify the ones that need most attention. Either mitigations are put into place and/or detection mechanisms identified. Mechanisms need to be designed around the system to put detection and fail-over systems in place. Lastly, these mechanisms need to be field tested and their efficacy demonstrated over sufficient duration. Exhaustive testing is sometime cost-prohibitive so a risk-score based approach could be employed.

One well known apprehension about machine-learning systems is the behavior in case of unknown situations and or edge-cases. To counter this risk, it is highly desirable to determine a confidence for each prediction and use that prediction if and only if it does not exceed the risk threshold of the process in question. If the risk threshold is exceeded, the system simply follows default processing without the prediction or proceeds conservatively. The case is flagged for further offline analysis. The active indicators, human oversight and analytics function covered in next-section elaborate on this aspect.

4.3.4 Establish

In the *Establish* step the deployment is firmly cemented into the organization so that machine-learning and analytics can have long-lasting positive impact on the organization or business. This steps needs large amount of discipline on part of the deployment team as it involves the no so glamorous work that usually happens after

the big go decision has been given. It includes establishing design patterns, monitoring and updating models and sustaining the deployment [96].

4.3.4.1 Active Design Patterns

Active design patterns ensure that machine-learning deployments are maintainable by avoiding superfluous technical debt due to the unique nature of machine-learning algorithms. Paying down technical debt may initially appear less glamorous than research results usually reported in academic ML conferences. But it is critical for long-term system health and enables algorithmic advances and improvements thus making deployments sustainable and lucrative [48].

Erosion of abstraction boundaries makes changes and improvements difficult. By their very nature machine learning systems tend to be tightly coupled with the inputs thus any change in an input or hyper-parameter changes everything. For example, if the inputs are owned by a different engineering group that is making continuous improvements, the machine-learning model will be gravely affected by the changes. Changing the paradigm of from the model being the "logic" to the model being another variable that is allowed to change helps alleviate this situation.

Correction Cascades create model-dependency debt by using a base model and then building a derivative "correction" model to predict something that is only slightly different [83]. When proliferated, the net of models become a nightmare to change or improve. Instead, it might be better to provide a switch inside the model that can be manipulated through input variable, although this is coupling too.

Pipeline jungle can evolve organically in data preparation, as new signals are identified and new information sources added. Pipeline jungles can only be avoided by thinking holistically about data collection and feature extraction. Analysis of

data-dependencies without tools is highly unrealistic. A feature that is important to the model may no longer exist due to cost or time-saving in another engineering group. On teams with many engineers, or if there are multiple interacting teams, not everyone knows the status of every single feature, and it can be difficult for any individual human to know every last place where the feature was used. A tool which enables annotation of data sources and features can be helpful as in [97]. Automated checks can then be run to ensure all dependencies have the appropriate annotations, and dependency trees can be fully resolved.

Underutilized data-dependencies occur when there are variables in the model that are not relevant anymore like old product-codes. These features could be removed from the model with little or no loss in accuracy. But because they are still present, the model will assign them some weight, and the system is therefore vulnerable, sometimes catastrophically so, to changes in these features. A common mitigation strategy is to regularly evaluate the effect of removing individual features from a given model and act on this information whenever possible.

Dead code branches can similarly occur when following easy path to experimentation in an isolated branch that is then abandoned. Any unintended dependencies on these dead path then cause one to maintain backward compatibility which further constrains the flexibility of the code. As machine-learning systems are more prone of experimentation, these dead paths are rampant. It is recommended that experiments are restricted to realistic sandbox environments as far as possible.

Configuration management is of far higher importance in machine-learning systems more-so than other software systems as the core system is primarily defined by configuration. The model is built using configured set of features, algorithms and

hyper-parameters values. All these settings are subject to change asynchronously. As seen earlier, features come and go, models are refreshed and hyper-parameters tuned. Hence it is critical that a solid set of tools and standards be used for model specification with a tight control around production releases. Using PMML to push predictive models into production platforms could mean a less complex overall architecture, lower cost and greater scalability [98].

Glue code is usually needed to get data into and results out of machine-learning packages and can sometimes account for 95% of the codebase. Copious amounts of glue code often make experimentation with other machine learning approaches cost intensive. Real-world machine-learning systems require highly engineered solutions to one large-scale problem; whereas, machine-learning packages provide a one-stop-shop for multiple algorithms. While packages allow, easy interchange of algorithms, the glue code pattern implicitly embeds the problem construction space in supporting code. As a result, any experimentation requires expensive changes. Glue code can be reduced by choosing to re-implement specific algorithms within the broader system architecture. At first, re-implementing a machine learning algorithm in C++ or Java that is already available in R or Matlab may appear as waste of effort. However, the resulting system will require less integration glue code and hence be easier to test, maintain, and allow alternate approaches to be tested. Problem-specific machine learning code can be tweaked with specific domain knowledge that is hard to support in general packages [48].

Unintended loops are another big problem of machine-learning. For example, a model to alter use-behavior indirectly has the user-behavior as the input. Unintended loops are formed that cause the system to drift. The situation becomes

worse if there are undeclared consumers of the model predictions. Hence further obscuring the loop formation making detection much harder.

4.3.4.2 Autonomous Monitoring and Model-refresh

Autonomous monitoring and model refresh address a major concern of machine-learning deployment. Notice that the skillset factor appears more than once in the list of key factors influencing ML deployment. This is due to the apprehension that if building the initial model was such an involved process, would it not be complex (and cost intensive) to upkeep the models? Additionally, it is well known that real-world scenarios are not static; hence, these costs would be recurring. Having mechanisms to detect when a model is out of date and autonomously update or refresh it can be a big confidence building measures for deployment. Additionally, it is often necessary to pick a decision threshold for a given model to perform some action. The threshold is often picked manually to achieve a certain tradeoff between metrics like precision and recall. Thus if a model updates on new data, the old manually set threshold may be invalid and hence also now needs to be updated.

Section 3.2.4 highlighted the various ways in which the system can become dynamic and how this needs to incorporated in validation for real-world scenarios. However, after deployment, the dynamics do not cease. The system needs to be monitored mainly from two conditions: deterioration and runaway. Deterioration in predictive performance is relatively straightforward to detect, the main issues being choosing of a metric. However, even a simplistic metric like prediction bias is a useful to detect change in relationship between the features and predicted variables.

Runaway on the other hand is trickier – it is the end results of the unintended loops that might occur in machine-learning deployments. Let us say a

machine-learning system is designed to predict defective units on a manufacturing line. Whereas the units classified as 'good' might undergo further testing down the line, the units predicted to be defective may never be retested. Thus the false-positive rate might drift undetected. In order to prevent such situations, we need to have control samples on which the prediction is made but the actualization is withheld to ascertain that the algorithm is not creating a self-fulfilling prophecy.

Statistical process control methodologies may be applied to the metric of choice to detect outliers, cycles, drifts, trends in performance using the western electric rules. Based on the severity of the change detected, the model can be autonomously retrained. In systems that are used to take actions in the real world, trip-limits may be set. If the system hits a limit for a given action, automated alerts should fire and trigger manual intervention or investigation. It goes without saying that the whole system and process must be validated before deployment.

4.3.4.3 Analytics function & oversight

From the previous discussion it is clear that machine-learning deployments are not set-it-and-forget-it undertaking, at least not yet. A team is required to monitor the health of the system, respond to alerts and notifications, double-check model configurations and anything that the autonomous system might have missed. Furthermore, there is also the need for continuous improvement as new algorithms and technology becomes available. Thus there is a strong case for human oversight of the analytics deployment.

The glue code and pipeline jungles mentioned earlier are symptomatic of integration issues that may have a root cause in overly separated "research" and "engineering" roles. When machine learning packages are developed in an ivory-

tower setting, the resulting packages may appear to be more like black boxes to the teams that actually employ them in practice. A hybrid research approach where engineers and researchers are embedded together on the same teams (and indeed, are often the same people) will reduce this source of friction significantly.

Most of the major factors influencing successful deployment were addressed in this section except that for new insights. Note that the prospect of new-insights was cited as leadership's main reason for sponsoring analytics in the organization in the hope that it helps capture additional markets, offer new products and services or generates new avenues of business. New insights are possible only with long-term immersion in the domain and intimate knowledge of the data. Achieve that state, is realistic with a permanent team.

This thesis claims that analytics is not a one-time project and should be treated as a permanent function of the organization like finance, IT, marketing etc. Furthermore, it has also been reiterated that data-analytics is more than applying algorithms to data. It is a way of doing science, engineering and business and hence should be practiced by the entire organization not just a small temporary group of people. Having an analytics function with data-scientists on staff that work closely with other functions to continuously derive insights ensures the highest ROI.

Finally, machine-learning deployment should be treated just like software deployment with, design patterns, RESTful API, language bindings, security, privacy, continuous deployment, uptime and other SLAs. In order to make the most of their model assets, enterprises must develop the common processes for communicating and integrating model deployment practices across multiple constituencies in analytics, IT, information security, and other functions [99].

4.4 Demonstration and Results

A potentially lucrative real-world business case for machine-learning deployment was discussed in the previous chapter. And, it was shown that sometimes the domain constraints like system dynamics and temporal-shifts can render the productivity unfeasible. However, the knowledge of the data gained can be salvaged especially if the right infrastructure for dynamic evaluation have been put in place. Additionally, if the uncertain nature of analytics was taken into account and a roadmap of possible use-cases were put in place, pivoting on to the next use-case is straightforward. Thus the venture does not necessarily have to close with failure but just change direction towards avenue for success. Both the elements mentioned above were done in this work: Section 2.6.1 described the roadmap and Section 3.4.4 illustrated the evaluation framework. Multiple subsequent use-cases were picked up by different team-members [54], [56], [57], [100], [101].

4.4.1 Use-case

One of the use-cases is described herein is where the machine-learning solution was successfully taken from concept, development and multiple stages of validation into production. The case is that of die-matching – this case had the additional characteristic that although the ROI was promising, the solution was also critical to ensure that the microprocessor product was competitive is the marketplace (with respect to products offered by the competition). Thus the case had a sense of immediacy of business-need around it. As per the vitamin versus analogy mentioned in 2.5.1, this was most definitely a pain-killer – addressing an immediate business pain-point. The coopetition had introduced a product that was faster than Intel's offering and it would take a process generation (18 months) to catch-up.

To achieve the same speed at comparable power, two-die could be placed on the same substrate and run at lower speeds. However, in order to achieve the top-speed die, both of those die had to be from the same to-speed-bin. To provide context, recollect that the semiconductor manufacturing process has inherent variation and hence even die on the same wafer could have a wide range power (measured by stand-by current drawn, Isb) and speed (measured by the maximum frequency, Fmax). Figure 37 shows a plot of Isb versus Fmax as measured at Sort-test when the die is still on the wafer in its infancy. The bands are the DLCP (Diel level Cherry Picking) categories that the die are divided into as they are picked off in a serpentine path as shown in Figure 38. Each DLCP gets its own tape-reel, thus all die on a reel are from the same DLCP. Notice that all except the bottom band are based purely on Isb. Each DLCP is sold into a specific market segment – the high-power go into servers, medium into desktops and lower-power into mobile segments.

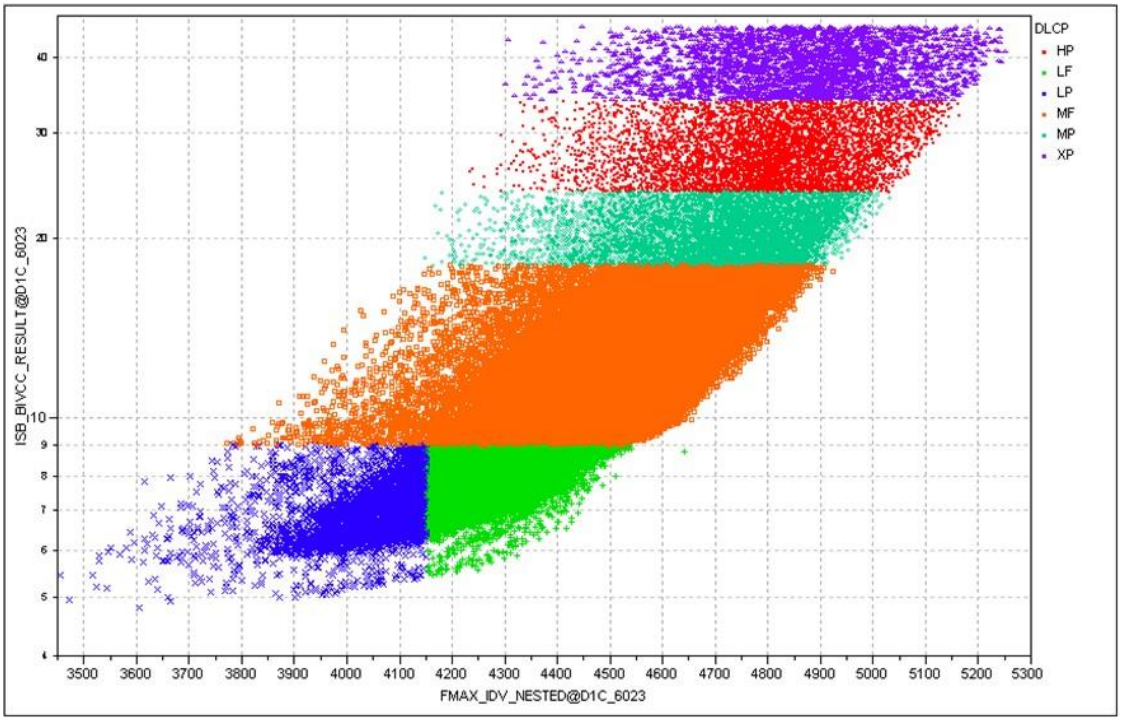


Figure 37. Raw Die Power (shown as Isb) and Speed (shown as Fmax) at Sort

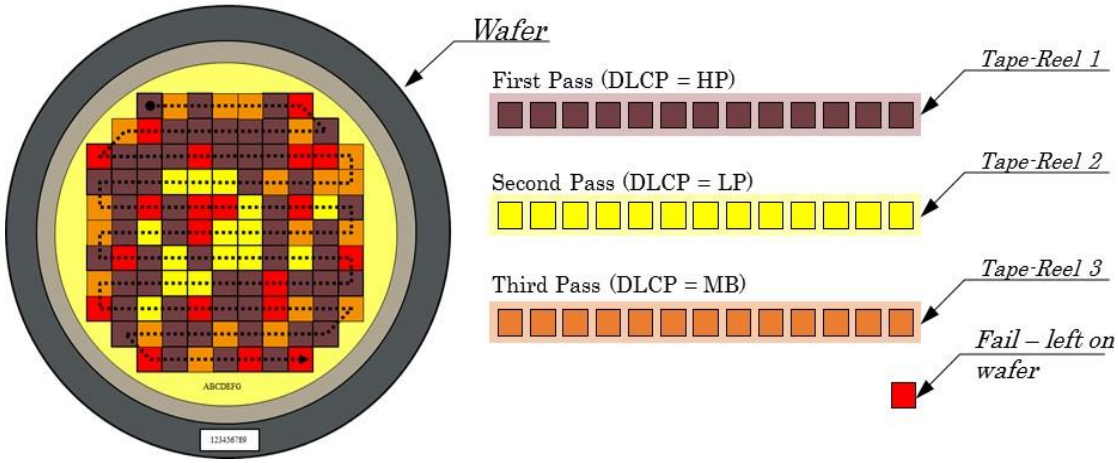


Figure 38. Die from each DLCP on Wafer is Put on a Separate Tape-reel

Observe from Figure 37 that within each DLCP segment i.e., the speed variation can be quite large – for instance, in the MP segment, speed can range from 4200 to 5000MHz. Recollect also that higher-speed die fetches a selling price that is upwards of ten-times that of lower-speed die. Additionally, each speed-level (AKA speed-bin) spans about 200MHz. Thus a single reel could contain die from 4 different speed-bins. In the CAM (Chip Attach Module), die are picked up from the wafer and placed on the reel in serial-order. Hence, when two die are placed on a single substrate, is it by pure chance that a top-bin (top-speed) part will be paired with another top-bin part. Note that the unit has to be sold at the speed of the lowest speed die on the substrate. Given that top bin parts are the rare-class, the analysis showed that early in the process lifecycle, probabilistically, one would not be left with any top-bin units. Thus there was an urgent need to find a better way to achieve top-bin units with two-die packaged onto one substrate. Another challenge is that as the die undergo physical change between sort-test and final class-test, the final speed of the die is only realized later. Hence the final speed and power do not have a simple correlation with speed and power at sort as in Figure 39.

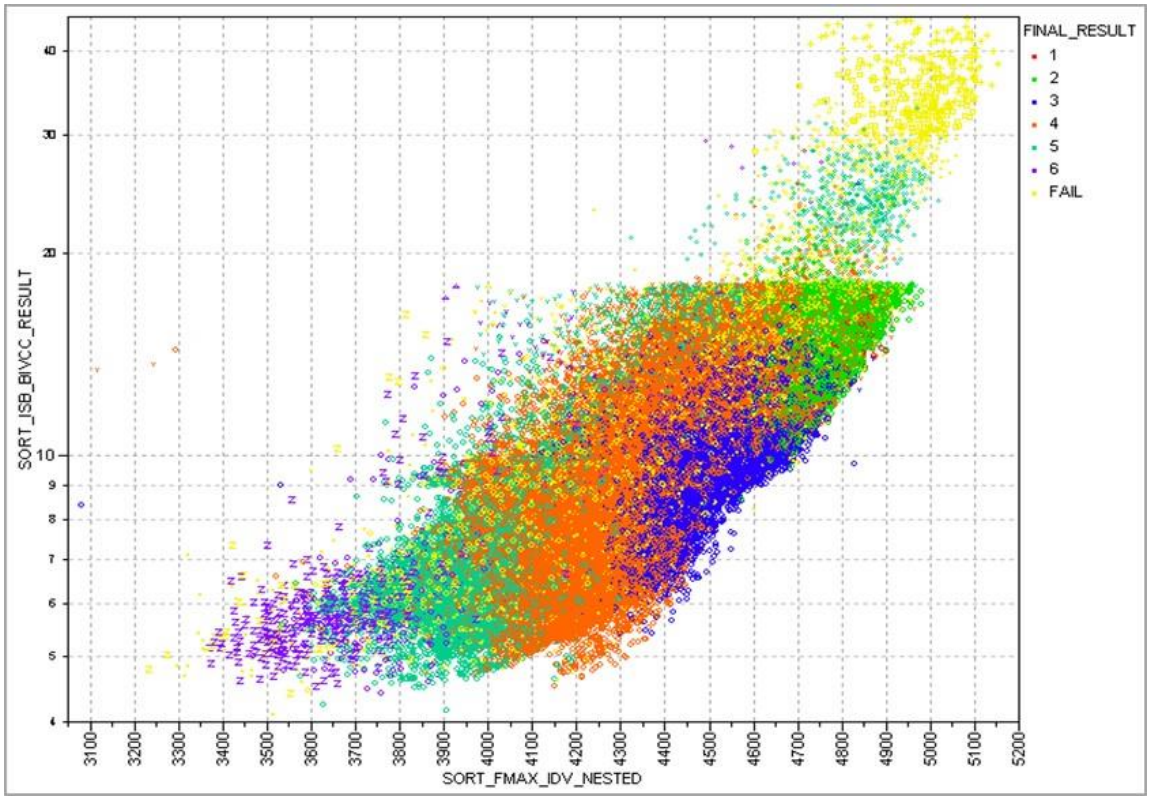


Figure 39. Isb versus Fmax Color-coded by Final Class Speed-bin

There is one more part to the problem. The manufacturing equipment is a large capital-cost hence every minute of the equipment's life has a dollar-value attached to it. The longer it takes to process a unit, the more it costs to manufacture. Hence, there is a constraint on the path-length traversed to find the matching die to been picked next off the wafer. Thus there are three parts to the problem. The first is that the final speed-bin of the die must be predicted based on data collected up-to sort-test in the manufacturing flow. Second, is to find the die on the wafer that is the closest match in terms of predicted-speed so that two neighboring die have a higher likelihood of being from the same speed-bin. Lastly, the total distance traversed to pick the closest matching die needs to be minimized like in the travelling salesman problem.

4.4.2 Solution

A new method was invented to combine the predictive algorithm and path-length optimization to provide the needed balance between speed and accuracy. Appendix A.1 has details of the algorithm. At a high level,

1. Historical data that includes all pertinent information is used to build a supervised machine learning model to predict the final class-bin.
2. Use the predicted class-bin probabilities to construct a pair wise dissimilarity (not necessarily Euclidian) matrix between units.
3. The weighted geometric pair-wise XY distance between die on the wafer is added to the distance matrix to optimize path-length.
4. A MDS (multidimensional scaling) algorithm is applied to the dissimilarity matrix to find a one-dimensional representation of the data such that the sum of distances between neighboring points in the sequence is minimal.

Initial experiments showed promising results with the machine-learning based method achieving higher likelihood of finding closer matches than the incumbent serpentine way of picking the die off the wafer. As seen in Figure 40, for two-die the percentage of matched increases from 40% to 80%. The G1 and G2 are processors from two different generations.

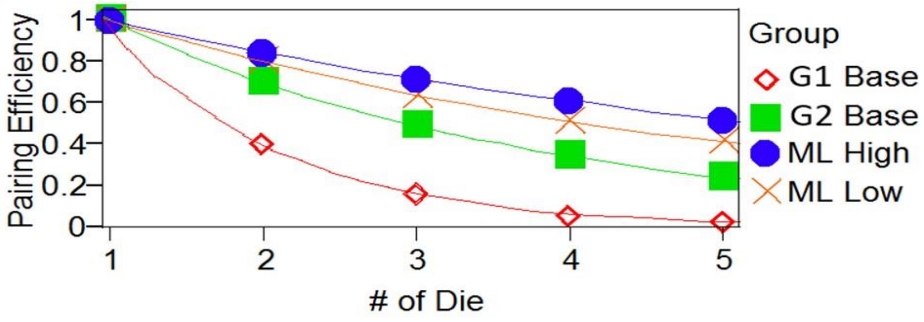


Figure 40. Preliminary Results of ML based Die-matching

4.4.3 Compatibility and Security

Section 4.3.1 covered the importance of ensuring compatibility with existing systems to ensure the deployment is successful. One of biggest compatibility issues for this problem was that the robotic arm that did the pick and place at the TRDS process module was designed to move only in the X and Y directions. Diagonal movements were not expected and hence even if the ML solution came up with an optimized path the arm could not travel that. The team worked the robotic-arm suppliers to drive the changes needed to enable a flexibly programmable path.

The idea was to minimize changes to the existing wafer map interface of both wafer map server and die sort equipment. The interface is standardized according to SEMI E5/Stream 12 (Semiconductor Equipment and Materials International) definitions [102]. Since there are large pieces of existing software in use to do wafer mapping, changes to the defined structure of the wafer map transaction sequences were not recommended. Instead, any solution to provide additional data in the wafer map would need to be encoded in the current structure or at least integrated into the existing message structure such that full compatibility with existing SECS/GEM (SEMI Equipment Communication Standards) software is guaranteed.

Any production deployment of Machine Learning algorithm needs to be several fold better than the existing solution or variants thereof that are less-complex and easier to implement than the ML approach. Note in the Netflix example that a superior algorithm was not implemented as the engineering cost and complexity outweighed the gain from the superior algorithm.

In case of Intel, changes (or lack thereof) to manufacturing are strictly governed by the *'Copy Exactly'* mantra (principle) which is described now. To be

considered for the next generation of the manufacturing process, proposed changes to the previous working process must demonstrate with data that they are statistically better. This is ensured through a change control process organized via a change control board (CCB) drawn from senior management and technical staff as well as a statistician, finance, factory, IT security and legal representatives. Proposed changes are documented and presented to the CCB for approval to start data collection. The preliminary white-paper details the experiments and data that will be collected to prove statistical superiority of proposed change. Once data is collected, a final white-paper is written with the result, analysis and recommendations; based on data, the CCB then decides.

Once a manufacturing process under development is demonstrated to meet the corporate quality and security standards, it is copied exactly to the other sites that will manufacture with the said process. Periodic audits and statistical matching activities ensure that all sites work as one 'virtual factory'.

4.4.4 Relative Performance, ROI and Reliability

The importance of proving relative performance was highlighted before. Once the robotic arm was improved, the question arose whether the cost and complexity of building and maintaining a ML system are justified. There are generally known heuristics that could provide an acceptable (good-enough) solution.

The baseline method is the serpentine TRDS combined with random CAM. The spiral method starts at the center and spirals out to the edge of the wafer. It relies on the process-engineering heuristic that die at the center of the wafer are faster than the edges and, die closest to one another are similar to one another. The heuristic arises from domain knowledge that spatial variation is one of the largest

contributor to variation in the manufacturing process. The other methods are based on raw speed (Fmax) measured at Sort-Test; raw speed and current (ISB); and, another heuristic named MCP DLCP derived from Sort-test.

A set of well-designed experiments helped quell any doubts and establish the superiority of the ML system. The simulation framework described earlier was used to setup, a loop was set up to span the entire 6-month timeframe, moving ahead one day at a time in the following fashion,

1. Extract training data set for a given time-span or 'window'
2. Construct the predictive model based on training data
3. Extract test data set from existing sort/test data
4. Generate the TRDS pick order for the ML and contending methods
5. Generate the die pairings for each method to construct a virtual unit
6. For every virtual unit calculate die-to-die variances for Fmax, Isb, and speed-bin for POR and ML methods
7. Move time window ahead by one day and repeat

The two criteria used are the top-bin percentage and overall match percentage across bins. Several stages of validation were performed to obtain final approval of the solution: the first two stages were in the lab followed by a factory pilot. The first stage (DOE 0) was a preliminary *simulation* (Section 2.5.6) to ensure results warranted further work. DOE 1 was the *feasibility validation* (Section 2.5.5) which includes 120k samples using the dynamic simulation environment as described above. DOE 2 is the *field validation* (Section 2.5.3) that include a factory pilot with 4.5 Million units to ensure reliability (Section 4.3.3.3). The results are summarized in **Error! Reference source not found.**.

Table 9. Results from Multiple Validation of ML Die-matching Solution

DOE	Method	TRDS Pick Method	CAM Placement Method	# of DLCP Categories	TOP Bin %	Overall Bin-match %	Sample Size
DOE 0	Baseline	Serpentine	Random	1		39.2	117474
	Sequential	Serpentine	Sequential	1		69.3	1421952
	FMAX	Predefined	Sequential	1		69.5	1363231
	ISB/FMAX	Predefined	Sequential	1		70.32	1370685
	MCP DLCP	Serpentine	Sequential	5		68.7	1508513
DOE 1	Sequential	Serpentine	Sequential	4	4.7	77	120000
	Spiral	Spiral	Sequential	4	5.1	77	120000
	Predictive	Predefined	Sequential	4	5.97	85	120000
DOE 2	Sequential	Serpentine	Sequential	6	16.4	65	4500000
	Spiral	Spiral	Sequential	6	16.5	66	4500000
	Predictive	Predefined	Sequential	6	17.4	68	4500000

Note in DOE 2 the algorithms were tweaked to favor top-bin over die-matching at every bin level as top-bin was the ROI driver for this particular product, factory and market mix; a change in which would necessitate re-tuning. The trade-off between path-length and die closeness can be seen in Figure 41. As seen the ML method is not just superior to alternatives but also provide ability to tune by weight.

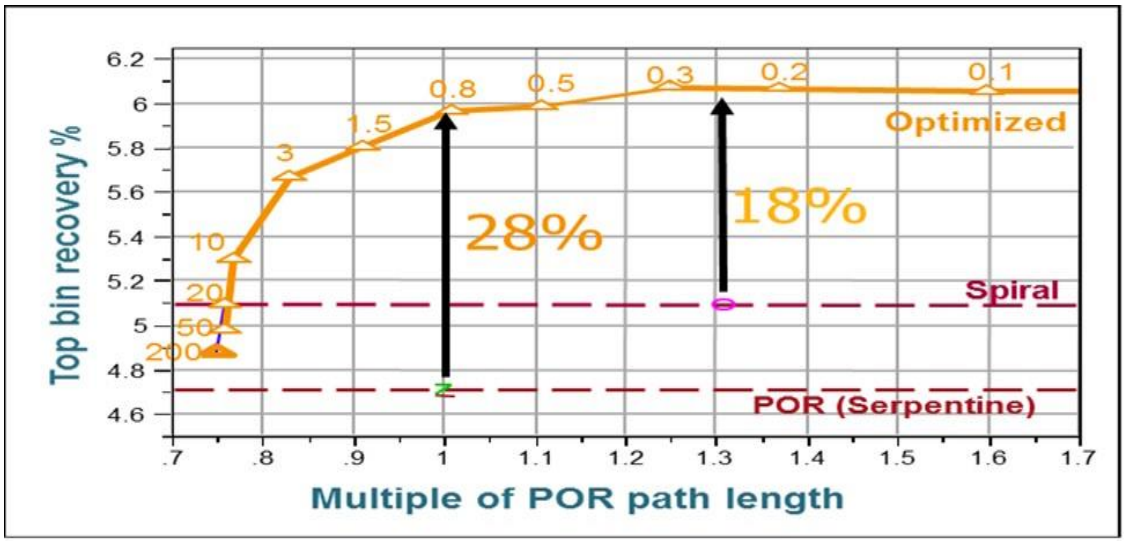


Figure 41. Trade-off Between Path-length and Die Bin-matching

An example of a path traversed based on weight is shown in the Figure 42. As seen, the higher the weight, the more emphasis is given to path-length optimization and smaller weights cause the matching to be better but then the path traversed is long.

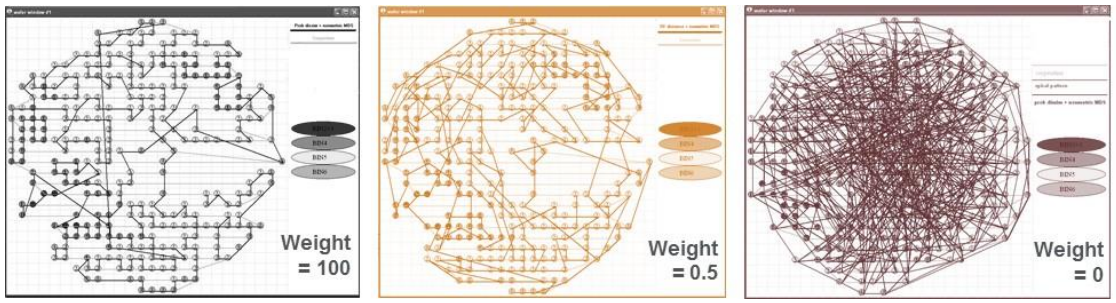


Figure 42. Paths Lengths Traversed Based on Weight

The machine-learning based ordering resulted in the speed-bin distribution shift towards the lucrative side as shown in Figure 43. The speed-bin yield of bin2 and 3 went up from 4.7% to 6.1% - this might not seem like a large jump but given the difference in average selling price and volumes manufactured, it adds up to a lot of value. Based on the simulations and the factory pilot, the 5-year Net-Present-Value ROI computed on this project was to the tune of $33 Million. Furthermore, it helped the sponsors deliver a market response to the competitive challenge. The resulting US patent filed for this work [103] was cited by 10 Google patents.

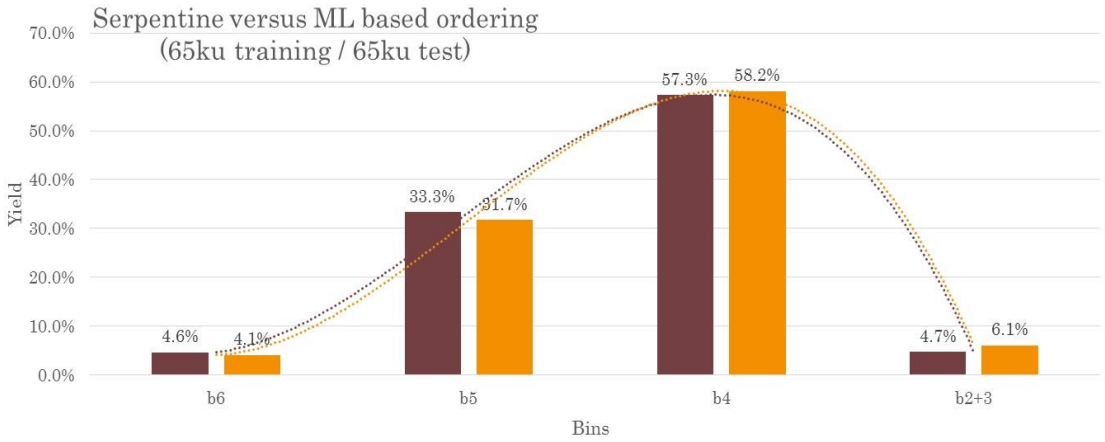


Figure 43. Yield Improvement Across Speed-bins with ML Ordering

CHAPTER 5

CONCLUSION AND FUTURE WORK

The promise of data science has been around under various names and forms for several decades now, statistics, design of experiments, KDD, data mining and now data-science. The field has had a large impact on humanity just as agriculture had centuries ago on forming early civilizations. The next turning point in agriculture that led to human growth came with the exercise of *yield improvement* through modern seeding, growth and harvesting techniques. Instead of scattering seeds and let nature take its course, humans started looking at each seed as an investment and determined what was needed to improve the yield of that investment. Similarly, it is time that data scientists look at what is needed to ensure the success of each data-analytics project so it bears fruit. This work is an attempt to distill the knowledge so far and extend it to improve the yield of deploying machine-learning in the real-world.

5.1 Summary

This work started with identifying the problem that less than 50% of the data-analytics projects today culminate if failure. Lack of a solid framework was identified as one of the key causes. The need for frameworks justified and shortcoming of existing frameworks illustrated. An analysis of the evolution of data-analytic frameworks with the objective of extracting unique and valuable contributions from each of the prior proposed frameworks followed. Based on the analysis a novel *Design for Deployment* framework was proposed. The proposed framework emphasizes the role of continuous validation throughout the exploring, development and implementation phases of data-analytics to ensure each step is

taking the project closer to deployment and any issues are identified early. Further, the work focused on the evaluation and deployment functions within the analytics.

The state of the art and problem with prevalent methods and metrics used by default for evaluating algorithms in machine-learning research and development we examined. Next, some challenges that are unique to real-world scenarios, namely, system-dynamics and temporal-shifts were identified and their role on impeding success of machine-learning deployments was highlighted. Recommendations were made to setup a versatile simulation framework. The framework is capable of evaluating algorithms in a dynamic environment by applying temporal-shifts and walking through time. The proposed methodology and simulation framework were demonstrated on a real-world use-case in semiconductor manufacturing – one of the most complex industrial environments that exists. Through the demonstration it was clear that although each machine-learning project need not end up in a useful model, following the recommendation set-forth allows thee team to pivot on to the next idea for application of machine learning in that enterprise.

Moving to the deployment function, the problem and current state were examined and real-world challenges explained. Whereas there are a multitude of challenges, they were shown to be dominated by reliability, interpretability, and maintainability in addition to the issue of trust and change. A four-pronged approach to counter these challenges was proposed with three aspects in each of the prongs. The next use-case in semiconductor manufacturing was used to demonstrate firstly how the foundations recommended earlier aided a fast pivot and reuse of knowledge gained. Additionally, the relevant aspects of deployment challenges were successfully overcome using recommendations made earlier.

5.2 Significant Contributions

This thesis is what can be termed as a position-work: highlighting a key area for the field that is in need of much research in order for the community to benefit as a whole. The area being that of increasing the yield of machine-learning algorithms by improving the rigor, framework and ultimately standards for building machine-learning analytics applications. It has been mentioned before that just as the software world lacked software engineering, the analytics world lacks '*analytics engineering*'. I hope this work is the leads to the creation of that field. It is already drawing attention by the fact that 11 patents by Google around the platform, creation, selection, assessment, storage, refreshing and application of models cite the patent from this work as shown below.

Table 10. Google Patents that Cite the Patent from this Work

Patent Number	**Title**	**Publication Date**
US9406019B2	Normalization of predictive model scores [104]	8/2/2016
US9239986B2	Assessing accuracy of trained predictive models [105]	1/19/2016
US9189747B2	Predictive analytic modeling platform [106]	11/17/2015
US9070089B1	Predictive model importation [107]	6/30/2015
US9020861B2	Predictive model application programming interface [108]	4/28/2015
US8694540B1	Predictive analytical model selection [109]	4/8/2014
US8595154B2	Dynamic predictive modeling platform [110]	11/26/2013
US8533222B2	Updateable predictive analytical modeling [111]	9/10/2013
US8521664B1	Predictive analytical model matching [109]	8/27/2013
US8443013B1	Predictive analytical modeling for databases [112]	5/14/2013
US8364613B1	Hosting predictive models [113]	1/29/2013

The key contributions of this work can be listed in terms of the claims made at the beginning of the document:

1. A comprehensive *Design for Deployment (DFD)* framework was proposed to conceptualize, build, validate, deploy and sustain machine-learning analytics in the real-world. The framework highlighted the role of validation and showed how each development step has a higher likelihood of success when paired with a validation step in a 'V' model. This would help channel analytics aspirations of enterprises world-over into tangible ROI and thus not have the analytics dram remain a dream.
2. Attention was drawn to the underserved latter third of analytics projects namely, evaluation, deployment and sustenance where much research is needs if analytics deployments are to success in the industry.
3. The fallacy of using the ubiquitous method of accuracy with cross-validation to compare and select algorithms was expose. Some alternatives were charted, with recommendation to use hold-out and kappa as the new default.
4. Two critical realities of the real-world autonomous analytics were identified: dynamics and temporal-shifts. Dynamics in the relationship between features and predicted values as well as the temporal-shifts between model-building and scoring/prediction. The detrimental effect these have on viability of machine-learning algorithms was demonstrated.
5. A *Dynamic Evaluation Framework* (DEF) was recommended that would incorporate simulation of the dynamics as well as the temporal-shift to give representative results from running Design of Experiments.

that could be trusted.

6. The need for updating all-models in the field was highlighted and techniques were recommended to detect model deterioration as well as parameters that need to be considered for refresh like training-window.
7. A four-pronged approach called *Deployment Four-E (DFE)* was recommended to deploy and sustain machine-learning analytic systems in the field. The approach addressed how to overcome common deterrents around deploying machine-learning analytics into production systems by leveraging the available incentives. Interpretability, maintainability, reliability, compatibility, security, privacy, viability, and many-more were addressed using the approach.
8. A "clutch" architecture was recommended that would allow a non 24/7 uptime SLA machine-learning analytics module to interact and govern decisions in a mission-critical 24/7/365 SLA system.
9. The recommended *Design for Deployment (DFD)* framework and *Dynamic Evaluation Framework* were used to demonstrate how strategic ROI can be maximized by pivoting to other items on the roadmap when faced with failure. Thus recommending '*analytics as a function*' paradigm.
10. Lastly, the *Design for Deployment (DFD)* analytics framework, the *Dynamic Evaluation Framework (DEF) and the Deployment by Four Es (DFE)* were demonstrated in semiconductor manufacturing with a learning experience from a dead-end and pivot towards the successful deployment with $33 Million ROI.

To summarize, the *Design for Deployment (DFD)* framework allows treating analytics as a function of the enterprise rather than a one-time project. Hence when faced with a stubborn roadblock, the team can always pivot on the chosen solution path or the direction of exploration thus allowing for agility in the analytics. Moreover, with each discovery and development phase married to a corresponding validation step, deployment thinking is encouraged in the same spirit as test driven development (TDD).

The *Dynamic Evaluation Framework (DEF)* provides a simulation environment that is capable of "walking" through time to ensure the stability and sustainability of the machine-learning solution. A good simulation setup closely represents reality while providing expeditious answers to crucial questions by allowing the toggling of variables that have the highest influence on the chosen metric. The framework allows experimentation with the various parameters affecting the model at different phases in the data analytics process. Furthermore, decisions can be based on experiments and data instead of intuition and rules of thumb.

The *Deployment Four-E (DFE)* approach helped addressed many deterrents of deploying machine-learning analytics in the field. At the end of the day, a machine-learning system is also software and must be subject to the same rigor of deployment and maintenance as any other software system. The proposed approach allows for this and hopefully draws more interest for further research.

Note in the Netflix example at the beginning of the document, that a superior algorithm was not implemented as the engineering cost and complexity outweighed the gain from the algorithm. Thus it is time we focused on 'analytics engineering'.

5.3 Future Work

As pointed out earlier, this work is considered a positional-thesis – the hope is that it will spark further research in analytics engineering – a field that does not yet exist. Just as software engineering brought direction to programming languages and shifted focus from the excitement of creating new languages and constructs towards value generation for the user, analytics engineering should draw attention to the evaluation and deployment of machine-learning so they can bring us the promised value. There are several avenues for future research:

1. The methodologies proposed here were demonstrated on two use-cases, both is semiconductor manufacturing. The author also used the methodology in an energy sustainability startup under the auspices of ASU center for entrepreneurship. The results were not included in this thesis due to drive focus. As many of the analytics venture today are startups, the unique set of challenges created by a startup environment deserve more attention.
2. In general industry shies from using machine-learning in mission critical systems due to general mistrust in black-box methods and that even researchers believe they can be unpredictable at boundaries. As use of analytics becomes pervasive in industry, self-driven cars and such, this excuse would not be acceptable any longer. We demonstrated the clutch architecture to use a ML system in a 24/7 application. Thus there is lot more progress that can be made on using ML in mission-critical systems.

3. The clutch architecture proposed was specifically to address the uptime problem among others – what are common elements of an architecture suitable for analytics?
4. Design patterns for data-analytics are another ripe area of research. Currently they are next to nonexistent with several industry influencers trying to grapple with the issue. Research in this area would greatly help the industry.
5. Analytics process-models started appearing only at the turn of the century. It has come a long way from waterfall to agile and TDD. Similarly, it is would be a worthy topic to experiment vis more case studies and research what suites analytics projects what are the boundary conditions and caveats.
6. The CMMI standards exist for software development, some have been proposed for physics models as well [114] – what would the corresponding standards for analytics look like? Although PMML has been around for a few years, it is to analytics like UML is to software-engineering. It is not an end-all elixir and there are process questions.
7. Earlier, a reference was made to DMCoMo – like CoCoMo it is a proposed method to estimate costs of a data-analytics project. There is scope for further research on the robustness of this model and adoption.
8. Computer science is aided by the quality assurance (QA) field that has its own tools, techniques and expertise in addition to software engineering, for efficient value-generation. There is a similar opportunity for analytics – how do we address the unique demands of QA for analytics [115]?

9. The issues of security and privacy are quite important in software-development. In analytics, this is further aggravated as information can be used in ways that have the potential to unintentionally expose vulnerabilities or sensitive information like in the Target example – much research is needed on how to put safety mechanisms in place to forecast, detect and prevent or contain these effects.
10. Computer science has seen a recent surge in research around social-media that intersects heavily with sociology, and even psychology. There are similar opportunities for machine-learning with business [116], finance and energy [117] which can be explored with more zeal.
11. Lastly, the establishment of standards has always benefits growth of any industry with great benefit to consumers. Although many tools, packages and approaches exist, the analytics industry could use standards that govern these aspects.

It is time machine-learning got out of the lab and started providing value to humankind for the years of research that has gone into it. This would be possible when we start thinking outside the algorithm and onto the broader aspects of *analytics engineering*.

REFERENCES

[1] D. J. Patil and M. Loukides, "Building Data Science Teams: The Skills, Tools, and Perspectives Behind Great Data Science Groups," O'Reilly, Sebastopol, CA, 2011.

[2] S. Higginbotham, "Want to ditch your data scientists? Here are 7 startups that can help," Jul-2012. [Online]. Available: https://gigaom.com/2012/07/05/want-to-ditch-your-data-scientists-heres-are-7-startups-that-can-help/. [Accessed: 18-May-2015].

[3] J. Joseph, "Using Big Data for Machine Learning Analytics in Manufacturing," Tata Consultancy Services, 2014.

[4] J. Manyika *et al.*, "Big data: The next frontier for innovation, competition, and productivity," McKinsey & Company, May 2011.

[5] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms*, 1st ed. Cambridge University Press, 2011.

[6] M. Herman *et al.*, *The Field Guide to Data Science*. Booz Allen Hamilton, 2013.

[7] X. Amatriain and J. Basilico, "Netflix Recommendations: Beyond the 5 stars (Part 1)," *The Netflix Tech Blog*, Apr-2012. [Online]. Available: http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html. [Accessed: 14-May-2015].

[8] M. Fania and J. D. Miller, "Mining Big Data in the Enterprise for Better Business Intelligence," Intel IT, Jul. 2012.

[9] R. Salkowitz, "From Big Data to Smart Data: Using data to drive personalized brand experiences," Microsoft, 2014.

[10] D. Gillblad, "On practical machine learning and data analysis," Doctoral Dissertation, KTH School of Computer Science and Communication, Stockholm, Sweden, 2008.

[11] Intel, "Transistors to Transformations - From Sand to Silicon - How Intel Makes Chips," 2012. [Online]. Available: http://www.intel.com/content/www/us/en/history/museum-transistors-to-transformations-brochure.html.

[12] K. B. Irani, J. Cheng, U. M. Fayyad, and Z. Qian, "Applying machine learning to semiconductor manufacturing," *IEEE Expert*, vol. 8, no. 1, pp. 41–47, Feb. 1993.

[13] J. T. Pfingsten, "Machine Learning for Mass Production and Industrial Engineering," Doctoral Dissertation, Mathematics and Physics, Eberhard Karls Universität Tübingen, Germany, 2007.

[14] Y. Zhu and J. He, "Co-Clustering Structural Temporal Data with Applications to Semiconductor Manufacturing," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 4, pp. 1–18, May 2016.

[15] C.-H. Lee, H.-C. Yang, S.-C. Cheng, and S.-W. Tsai, "A Hybrid Big Data Analytics Method for Yield Improvement in Semiconductor Manufacturing," in *Proceedings of the ASE BigData & SocialInformatics*, 2015, p. 9.

[16] R. J. Baseman, J. He, E. Yashchin, and Y. Zhu, "Run-to-run control utilizing virtual metrology in semiconductor manufacturing," US Patent 9 299 623, 29-Mar-2016.

[17] C. Giraud-Carrier and O. Povel, "Characterising Data Mining Software," *Intell. Data Anal.*, vol. 7, no. 3, pp. 181–192, 2003.

[18] K. Cios, R. Swiniarski, W. Pedrycz, and L. Kurgan, "The knowledge discovery process," in *Data Mining: A Knowledge Discovery Approch*, Springer, 2007, pp. 9–24.

[19] R. Wirth and J. Hipp, "CRISP-DM : Towards a Standard Process Model for Data Mining," in *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 2000, pp. 29–39.

[20] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[21] Ó. Marbán, G. Mariscal, and J. Segovia, "A Data Mining & Knowledge Discovery Process Model," in *Data Mining and Knowledge Discovery in Real Life Applications*, J. Ponce and A. Karahoc, Eds. I-Tech Education and Publishing, 2009, pp. 1–16.

[22] Ó. Marbán, G. Mariscal, E. Menasalvas, and J. Segovia, "An Engineering Approach to Data Mining Projects," *Intell. Data Eng. Autom. Learn. - IDEAL 2007*, vol. 4881, pp. 578–588, 2007.

[23] J. Wills, “From the Lab to the Factory: Building a Production Machine Learning Infrastructure,” 2014. [Online]. Available: http://www.infoq.com/presentations/machine-learning-infrastructure.

[24] G. Piatetsky, “CRISP-DM, still the top methodology for analytics, data mining, or data science projects,” 2014. [Online]. Available: http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html.

[25] L. A. Kurgan and P. Musilek, “A survey of Knowledge Discovery and Data Mining process models,” *Knowl. Eng. Rev.*, vol. 21, no. 1, pp. 1–24, Jul. 2006.

[26] G. Mariscal, Ó. Marbán, and C. Fernández, “A survey of data mining and knowledge discovery process models and methodologies,” *Knowl. Eng. Rev.*, vol. 25, no. 2, pp. 137–166, 2010.

[27] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, “Advances in knowledge discovery and data mining,” Feb. 1996.

[28] S. Ahangama and C. D. Poo, “Designing a Process Model for Health Analytic Projects,” *PACIS 2015 Proc.*, 2015.

[29] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “Knowledge Discovery and Data Mining: Towards a Unifying Framework.,” in *Int Conf on Knowledge Discovery and Data Mining*, 1996, pp. 82–88.

[30] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From Data Mining to Knowledge Discovery in Databases,” *AI Mag.*, vol. 17, no. 3, p. 37, 1996.

[31] P. Cabena, P. Hadjinian, R. Stadler, J. Verhees, and A. Zanasi, “Discovering data mining: from concept to implementation,” Jan. 1998.

[32] R. J. Brachman and T. Anand, “The Process of Knowledge Discovery in Databases: A First Sketch,” in *KDD-94 AAAI-94 Workshop on Knowledge Discovery in Databases*, 1996, pp. 37–57.

[33] C. Gertosio and A. Dussauchoy, “Knowledge discovery from industrial databases,” *J. Intell. Manuf.*, vol. 15, no. 1, pp. 29–37, 2004.

[34] A. H. Milley, J. D. Seabolt, and J. S. Williams, “Data Mining and the Case for Sampling,” SAS Institute Inc, Cary, NC, 1998.

[35] A. Azevedo and M. F. Santos, "KDD, SEMMA and CRISP-DM: a parallel overview," *IADIS Eur. Conf. Data Min.*, no. January, pp. 182–185, 2008.

[36] A. G. Buchner, M. D. Mulvenna, S. S. Anand, and J. G. Hughes, "An Internet-Enabled Knowledge Discovery Process," *Proc. 9th Int. Database Conf. Hong Kong*, vol. 1999, pp. 13–27, 1999.

[37] Two Crows Corporation, *Introduction to Data Mining and Knowledge Disocvery*, 3rd ed. 2005.

[38] S. Sharma and K.-M. Osei-Bryson, "Toward an integrated knowledge discovery and data mining process model," *Knowl. Eng. Rev.*, vol. 25, pp. 49–67, 2010.

[39] C. Shearer, "The CRISP-DM model: the new blueprint for data mining," *J. data Warehous.*, vol. 5, no. 4, pp. 13–22, 2000.

[40] Y. Li, M. A. Thomas, and K. Osei-Bryson, "A snail shell process model for knowledge discovery via data analytics," *Decis. Support Syst.*, 2016.

[41] G. S. Nascimento and A. A. Oliveira, "An Agile Knowledge Discovery in Databases Software Process," *ICDKE*, 2012. .

[42] K. J. Cios and G. W. Moore, "Uniqueness of medical data mining," *Artif. Intell. Med.*, vol. 26, no. 1–2, pp. 1–24, 2002.

[43] S. Moyle and A. Jorge, "RAMSYS - A methodology for supporting rapid remote collaborative data mining projects," *ECML/PKDD01 Work. Integr. Asp. Data Mining, Decis. Support Meta-learning*, no. 1, 2001.

[44] J. Solarte, "A proposed data mining methodology and its application to industrial procedures," MS Thesis, Industrial Engineering, University of Tennessee, Knoxville, 2002.

[45] O. Marbán, J. Segovia, E. Menasalvas, and C. Fernández-Baizán, "Toward data mining engineering: A software engineering approach," *Inf. Syst.*, vol. 34, no. 1, pp. 87–107, 2009.

[46] O. Marbán, E. Menasalvas, and C. Fernández-Baizán, "A cost model to estimate the effort of data mining projects (DMCoMo)," *Inf. Syst.*, vol. 33, no. 1, pp. 133–150, 2008.

[47] J. Misiti, “Awesome Machine Learning,” *GitHub*. [Online]. Available: https://github.com/josephmisiti/awesome-machine-learning#awesome-machine-learning-.

[48] D. Sculley *et al.*, “Machine Learning : The High-Interest Credit Card of Technical Debt,” *NIPS 2014 Work. Softw. Eng. Mach. Learn.*, pp. 1–9, 2014.

[49] S. R. Covey, *The 7 Habits of Highly Effective People*. Free Press, 2004.

[50] O. Laudy, “Standard methodology for analytical models,” *LinkedIn*, 2015. [Online]. Available: http://olavlaudy.com/MediaWiki/index.php?title=Standard_methodology_for_analytical_models. [Accessed: 06-Aug-2016].

[51] D. Wetherill, “Broken links Why analytics investments have yet to pay off,” The Economist, Intelligence Unit, 2016.

[52] O. Laudy, “Data Science Data Architecture,” *LinkedIn*, 2015. [Online]. Available: http://olavlaudy.com/MediaWiki/index.php?title=Data_Science_Data_Architecture. [Accessed: 06-Aug-2016].

[53] N. Kupp and Y. Makris, “Integrated Optimization of Semiconductor Manufacturing : A Machine Learning Approach,” *IEEE Int. Test Conf.*, pp. 1–10, 2012.

[54] A. Borisov, I. Chikalov, E. St. Pierre, and E. Tuv, “Rule Induction for Identifying Multilayer Tool Commonalities,” *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 2, pp. 197–201, May 2011.

[55] H. Jing, R. George, and T. Eugene, “Contributors to a Signal from an Artificial Contrast,” in *Informatics in Control, Automation and Robotics II*, J. Filipe, J.-L. Ferrier, J. A. Cetto, and M. Carvalho, Eds. Springer Netherlands, 2007, pp. 71–78.

[56] W. Hwang, G. Runger, and E. Tuv, “Multivariate statistical process control with artificial contrasts,” *IIE Trans.*, vol. 39, no. 6, pp. 659–669, Mar. 2007.

[57] E. R. S. Pierre, E. Tuv, and A. Borisov, “Spatial Patterns in Sort Wafer Maps and Identifying Fab Tool Commonalities,” in *ASMC (Advanced Semiconductor Manufacturing Conference) Proceedings*, 2008, pp. 268–272.

[58] E. R. S. Pierre, E. Tuv, and A. Borisov, “Classification of spatial patterns on wafer maps,” US Patent US7 937 234 B2, 2008.

[59] R. Goodwin, R. Miller, E. Tuv, A. Borisov, M. Janakiram, and S. Louchheim, "Advancements and Applications of Statistical Learning/Data Mining in Semiconductor Manufacturing.," *Intel Technol. J.*, vol. 8, no. 4, 2004.

[60] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[61] F. Provost, T. Fawcett, and R. Kohavi, "The Case Against Accuracy Estimation for Comparing Induction Algorithms," *Proc. Fifteenth Int. Conf. Mach. Learn.*, pp. 445–453, 1998.

[62] A. Ben-David, "A lot of randomness is hiding in accuracy," *Eng. Appl. Artif. Intell.*, vol. 20, no. 7, pp. 875–885, 2007.

[63] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[64] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.

[65] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, 2006, pp. 233–240.

[66] D. M. Powers, "Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, Dec. 2011.

[67] L. R. Hope and K. B. Korb, "A Bayesian metric for evaluating machine learning algorithms," *AI 2004 Adv. Artif. Intell. Proc.*, vol. 3339, pp. 991–997, 2004.

[68] I. Kononenko and I. Bratko, "Information-Based Evaluation Criterion for Classifier ' s Performance," *Mach. Learn.*, vol. 6, pp. 67–80, 1991.

[69] J. Gama, R. Sebastiao, and P. P. Rodrigues, "On evaluating stream learning algorithms," *Mach. Learn.*, vol. 90, no. 3, pp. 317–346, 2013.

[70] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Int. Jt. Conf. Artif. Intell.*, vol. 14, no. 12, pp. 1137–1143, 1995.

[71] NIST/SEMATECH, "8.1.3.1 Censoring," *e-Handbook of Statistical Methods*, 2012. [Online]. Available: http://www.itl.nist.gov/div898/handbook/apr/section1/apr131.htm.

[72] N. Japkowicz, "Why Question Machine Learning Evaluation Methods? (An illustrative review of the shortcomings of current methods)," *AAAI-2006 Work. Eval. Methods Mach. Learn.*, p. 6, 2006.

[73] O. Laudy, "Data Science Data Logic," *LinkedIn*, 2015. [Online]. Available: http://olavlaudy.com/MediaWiki/index.php?title=Data_Science_Data_Logic. [Accessed: 06-Aug-2016].

[74] F. Zamora-Martinez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy Build.*, vol. 83, pp. 162–172, Nov. 2014.

[75] T. R. Hoens and N. V. Chawla, "Learning in non-stationary environments with class imbalance," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, 2012, p. 168.

[76] S. Mitchell, "The application of machine learning techniques to time-series data," 1995.

[77] E. Alpaydin, *Introduction to Machine Learning, second edition*. 2010.

[78] P. S. Science, "Introduction to Design of Experiments," *Penn State Science*. [Online]. Available: https://onlinecourses.science.psu.edu/stat503/node/5.

[79] D. C. Montgomery, *Design and Analysis of Experiments*, 8th ed. Wiley, 2013.

[80] A. Borisov, I. Chikalov, V. Eruhimov, and E. Tuv, "Performance and Scalability Analysis of Tree-Based Models in Large-Scale Data-Mining Problems," *Intel Technol. J.*, vol. 9, no. 2, pp. 143–150, May 2005.

[81] P. Smyth, "Applying Classification Algorithms in Practice," *Stat. Comput.*, vol. 7, pp. 45–56, 1997.

[82] J. Bloom and H. Brink, "Overcoming the Barriers to Production-Ready Machine Learning Workflows," *Strata Conference - Making Data Work*, 2014. [Online]. Available: http://conferences.oreilly.com/strata/strata2014/public/schedule/detail/32314.

[83] X. Amatriain, “Ten Lessons Learned from Building (real-life impactful) Machine Learning Systems,” *Machine Learning Conference (MLconf)*, 2014. [Online]. Available: http://www.slideshare.net/xamat/10-lessons-learned-from-building-machine-learning-systems.

[84] T. Dunning, “Which Algorithms Really Matter?,” *ACM international conference on Information & Knowledge Management*. slideshare.net, San Francisco, 2013.

[85] R. Rana, M. Staron, C. Berger, J. Hansson, M. Nilsson, and W. Meding, “The Adoption of Machine Learning Techniques for Software Defect Prediction: An Initial Industrial Validation,” *Commun. Comput. Inf. Sci.*, vol. 466 CCIS, pp. 270–285, 2014.

[86] Nucleus Research, “The Stages of an Analytic Enterprise,” *Nucl. Res.*, no. March, pp. 1–6, 2012.

[87] J. Holdowsky, M. Mahto, M. E. Raynor, and M. Cotteleer, “Inside the Internet of Things (IoT),” Deloitte University Press, 2015.

[88] N. Lavesson, V. Boeva, E. Tsiporkova, and P. Davidsson, “A method for evaluation of learning components,” *Autom. Softw. Eng.*, vol. 21, no. 1, pp. 41–63, 2013.

[89] R. H. Weber, “Internet of Things – New security and privacy challenges,” *Comput. Law Secur.*, vol. 26, no. 1, pp. 23–30, Jan. 2010.

[90] M. A. Barreno, “Evaluating the Security of Machine Learning Algorithms,” PhD Dissertation, Computer Science, University of California, Berkley, 2008.

[91] P. Laskov and M. Kloft, “A Framework for Quantitative Security Analysis of Machine Learning,” *Proc. 2nd ACM Work. Secur. Artif. Intell.*, pp. 1–4, 2009.

[92] T. Woods, M. Evans, D. Rust, and B. Podoll, “Security in Machine Learning : Measuring the relative sensitivity of classifiers to adversary-selected training data.” pp. 1–11.

[93] C. C. Aggarwal and P. S. Yu, “Chapter 2 A General Survey of Privacy-Preserving Data Mining Models and Algorithms,” *Privacy-preserving data Min.*, pp. 11–52, 2008.

[94] T. H. Davenport, J. G. Harris, D. W. De Long, and A. L. Jacobson, “Data to Knowledge to Results : Building an Analytic Capability,” *Calif. Manage. Rev.*, vol. 43, no. 2, pp. 117–138, 2001.

[95] P. Turney, P. Turney, and N. R. C. Ca, "Types of Cost in Inductive Concept Learning," in *Cost-Sensitive Learning Workshop at the 17th International Conference on Machine Learning (ICML)*, 2000, vol. 6, pp. 1–7.

[96] M. Gasner, "Design Challenges for Real Predictive Platforms," *Strata Conference - Making Data Work*, Feb-2014. [Online]. Available: http://conferences.oreilly.com/strata/strata2014/public/schedule/detail/31779.

[97] H. B. Mcmahan *et al.*, "Ad Click Prediction : a View from the Trenches Categories and Subject Descriptors," in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and DataMining, KDD 2013*, 2013.

[98] J. Taylor, "Standards-based Deployment of Predictive Analytics," Decision Management Solution, 2016.

[99] R. Way, "Model Deployment: The Moment of Truth - Analytic Model Deployment Best Practices & Case Studies," CORIOS, Portland, OR, 2013.

[100] M. G. Baydogan, G. Runger, and E. Tuv, "A Bag-of-Features Framework to Classify Time Series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2796–2802, Nov. 2013.

[101] H. Deng, G. Runger, and E. Tuv, "Bias of Importance Measures for Multi-valued Attributes and Solutions," in *ICANN 2011, Part II, Lecture Notes in Computer Science*, 2011, vol. 6792, pp. 293–300.

[102] Semiconductor Equipment and Materials International, "SEMI E5-0813 - SEMI Equipment Communications Standard 2 (SECS-II)," *SEMI E5*, 2012. [Online]. Available: http://ams.semi.org/ebusiness/standards/SEMIStandardDetail.aspx?ProductID=1948&DownloadID=3110.

[103] E. Tuv, S. Shahapurkar, and A. Borisov, "Method for selecting a rank ordered sequence based on probabilistic dissimilarity matrix," US Patent 0 239 652 A1, 2007.

[104] W.-H. Lin, T. H. K. Green, R. Kaplow, G. Fu, and G. S. Mann, "Normalization of predictive model scores," US Patent 8 370 279 B1, 2013.

[105] W.-H. Lin, T. Green, R. Kaplow, G. Fu, and G. S. Mann, "Assessing accuracy of trained predictive models," US Patent 8 533 224 B2, 2013.

[106] G. S. Mann, J. M. Breckenridge, and W.-H. Lin, "Predictive analytic modeling platform," US Patent 8 706 659 B1, 2014.

[107] W.-H. Lin, T. H. K. Green, R. Kaplow, G. Fu, and G. S. Mann, "Predictive model importation," US Patent 8 583 576 B1, 2013.

[108] W.-H. Lin, T. H. K. Green, R. Kaplow, G. Fu, and G. S. Mann, "Predictive model application programming interface," US Patent 8 229 864 B1, 2012.

[109] W.-H. Lin, T. H. K. Green, R. Kaplow, G. Fu, and G. S. Mann, "Predictive analytical model selection," US Patent 8 694 540 B1, 2014.

[110] J. M. Breckenridge, T. Green, R. Kaplow, W.-H. Lin, and G. S. Mann, "Dynamic predictive modeling platform," US Patent 8 595 154 B2, 2013.

[111] J. M. Breckenridge, T. H. K. Green, R. Kaplow, W.-H. Lin, and G. S. Mann, "Updateable predictive analytical modeling," US Patent 8 250 009 B1, 2012.

[112] W.-H. Lin, T. H. K. Green, R. Kaplow, G. Fu, and G. S. Mann, "Predictive analytical modeling for databases," US Patent 8 443 013 B1, 2013.

[113] W.-H. Lin, T. H. Green, R. Kaplow, G. Fu, and G. S. Mann, "Hosting predictive models," US Patent 8 364 613 B1, 2013.

[114] R. G. Hills, W. R. Witkowski, A. Urbina, W. J. Rider, and T. G. Trucano, "Development of a Fourth Generation Predictive Capability Maturity Model.," Sandia National Laboratories (SNL-NM), Albuquerque, NM, Sep. 2013.

[115] X. Xie, J. W. K. Ho, C. Murphy, G. Kaiser, B. Xu, and T. Y. Chen, "Testing and validating machine learning classifiers by metamorphic testing," in *Journal of Systems and Software*, 2011, vol. 84, no. 4, pp. 544–558.

[116] A. I. Dimitras, S. H. Zanakis, and C. Zopounidis, "A survey of business failures with an emphasis on prediction methods and industrial applications," *Eur. J. Oper. Res.*, vol. 90, no. 3, pp. 487–513, May 1996.

[117] C. Cui, "Building Energy Modeling: A Data-Driven Approach," PhD Dissertation, Industrial Engineering, Arizona State University, Tempe, 2016.

[118] E. Tuv, "Ensemble Learning," in *Feature Extraction, Foundations and Applications*, vol. 207, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 187–204.

[119] K. Torkkola and E. Tuv, "Ensemble Learning with Supervised Kernels," in *Machine Learning: ECML 2005*, 2005, vol. 3720, pp. 400–411.

APPENDIX A

PREDICTIVE OPTIMIZATION ALGORITHM

Sorting entities in the order of a known variable (example: 5-year old's current height) is a trivial problem for which several efficient algorithms are available. However, sometimes (in manufacturing and other applications) it is desirable to rank-order entities by a variable whose present value is unknown. For example, arranging 5-year old's by how tall they will be when they are 17. In addition, it might be desirable to optimize for a cost function associated with building the ordered sequence. Say the 5-year old's need to be arranged along the shortest path such that daily pickup to sports coaching is feasible. The Intel application that inspired the invention of this algorithm is that of die-paring on multi-chip products as described in Section 4.4.1.

The solution consists of two components: prediction and optimization. The prediction approach is covered extensively in [118] and [119]. Any algorithm that provides prediction probabilities can be readily adopted. For optimization, a pair-wise distance matrix is then constructed using explicit partition of the predictor space by the learning machine as well as the distance representing the sequence cost. A flavor of non-parametric unidimensional scaling is then used to compute an optimized sequence. Neighboring members in the sequence have high likelihood of being matched on the predicted variable. Moreover, as the similarity matrix incorporates distance-cost, the resulting sequence minimizes "path length".

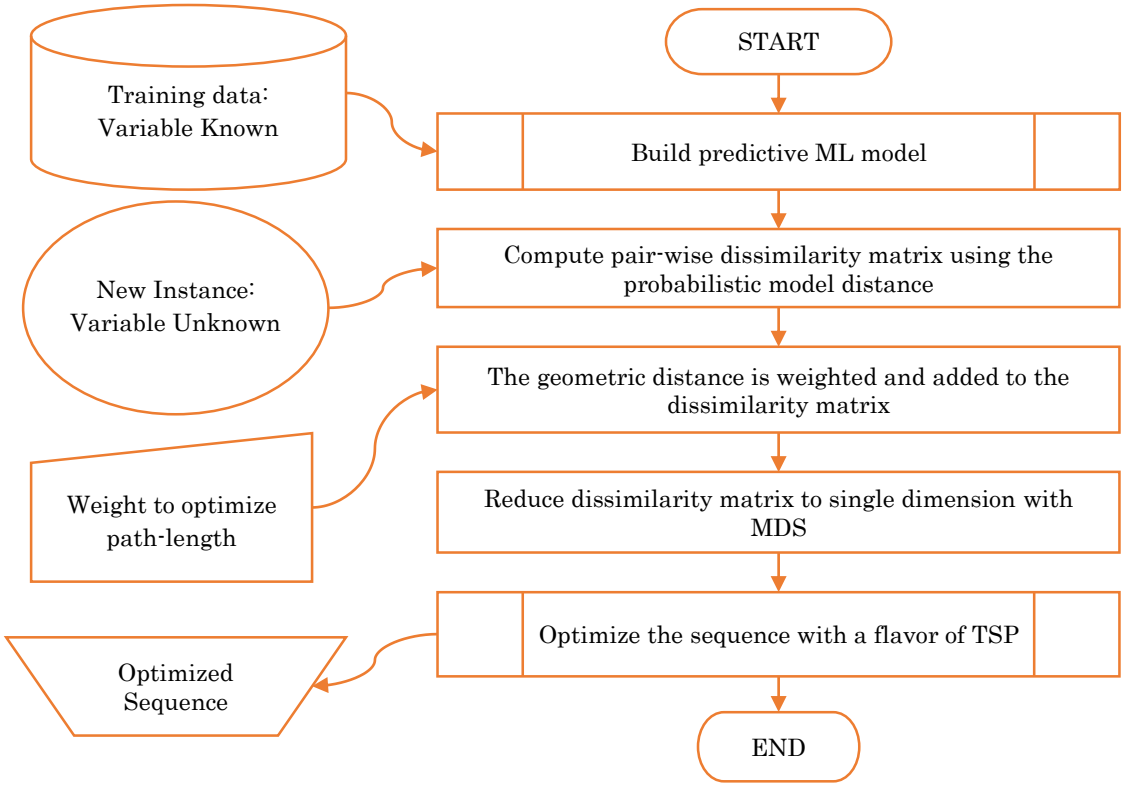


Figure 44. Predictive Optimization Algorithm

Figure 44 shows the predictive optimization algorithm - it is described below:

1. An enhanced random subspace algorithm model is built on the most recent historical data to learn unknown variable (r classes) using relevant measurable attributes.
2. The model applied to an observation results in a probability vector p of length r. We define the similarities between two entities k and m using a probabilistic distance $d_{km} = 1 - \sum_{i=1}^{r} p_{ki} p_{mi}$ that measures likelihood of 2 units belonging to the same class defined by probabilities predicted by RF classifier.
3. To optimize the cost *(path-length)* and overall matching error, the cost (*spatial distance*) is weighted and added to each element of distance

matrix $\mathbf{d_{ik}}$: *i,k=1,..,N*. The weight determines the importance given to the cost parameter: when the weight is set to zero, the path length is ignored and when weight is large, it amounts to pure cost optimization.

4. Given the resulting distance matrix $\mathbf{d_{ik}}$: *i, k=1,…N* between *N* entities we search for a one-dimensional ranking (multidimensional scaling along one axes) of all points $\mathbf{r_{xi}}$: $r_{i1} < r_{i2} < \ldots < r_{iN}$ to minimize total sequence length, $L = \sum_{j=1}^{N} d_{i_j, i_{j+1}}$. The simulated annealing algorithm may be used to solve this optimization problem:
    a. Build initial sequence S:
        i. Take point 1 as starting point
        ii. Select point closest to previous, add to sequence and remove from eligibility list.
        iii. Repeat (i until all points are organized in a sequence
        iv. Set initial temperature *T = 10 * d(S)*
    b. Repeat 10*N times
        i. Take random transposition *(i, j)*
        ii. Calculate decrease in goal function delta *(i, j) = d(S')-d(S)*
        iii. If *δ(i, j)<0* accept *(i, j)* else, accept with probability *exp(-δ(i, j)/T)*
        iv. Reduce *T: T = T * (1-0.1/N)* and continue.

All parameters, such as temperature decrease rate, number of steps and permutations per step are chosen empirically, and the result is not sensitive to a choice of the parameters. Moreover, often just the initial step (4 gives the desirable sequence, and a very little is gained after running next two steps.

BIOGRAPHICAL SKETCH

Som Shahapurkar was born Somnath (Sachidanand) Suhas Shahpurkar in April 1975. He was raised in Bangalore, India and completed high-school from St. Joseph's Indian High School in Bangalore. He went on to pursue his dream of becoming an engineer at Bangalore University and earned his Bachelor of Engineering degree in Instrumentation & Electronics from Bangalore University in 1997, with a senior project in application of Fast Fourier Transforms for predictive control carries out at ABB Ltd. He joined Tata Infotech Ltd. (now merged with Tata Consultancy Services) as an Oracle Applications Developer. He obtained his MS in Electrical and Computer Engineering from University of Arizona with a thesis that proposed a hybrid clustering algorithm with application to gene microarray analysis. Soon after he joined Intel in Chandler, AZ as product engineer writing inline test programs for microprocessors. He earned a patent in machine-learning for die-pairing. He subsequently held diverse set of positions as a data-warehouse architect, statistician and hardware validation manager. While at Intel he earned his *Lean Six Sigma Black Belt* on a project with multi-million-dollar ROI. He also authored and taught statistics, data-science and lean product-development courses to his colleagues. Som launched a startup in home energy-management using machine-learning techniques from his research with the help of the Center for Innovation at ASU and went on to become a semi-finalist at CleanTech Open. During his later years at Intel, he diversified into customer and business development for smart-home and buildings, retail, self-driven automotive and analytics IoT business verticals. Som started pursuing his dream of earning a doctoral degree while at Intel and has been publishing mainly in Intel journals as well as internal conferences. In 2015, Som joined Verizon Telematics as the Director of Software Development and in 2016 October he moved back from into research with FICO as *Principal Analytic Scientist – Go to Market*. In addition to spending time with his family, Som likes to ride his bicycle for charitable causes and facilitate *7-Habits of Highly Effective People program*.