A Data Mining Approach to Modeling Customer Preference:
A Case Study of Intel Corporation

by

Sudarshan Venkat Ram

A Thesis Presented in Partial Fulfillment

of the Requirements for the Degree

Master of Science

Approved September 2017 by the

Graduate Supervisory Committee:

Karl Kempf, Co-Chair
Teresa Wu, Co-Chair
Feng Ju

ARIZONA STATE UNIVERSITY

December 2017

ABSTRACT

Understanding customer preference is crucial for new product planning and marketing decisions. This thesis explores how historical data can be leveraged to understand and predict customer preference. This thesis presents a decision support framework that provides a holistic view on customer preference by following a two-phase procedure. Phase-1 uses cluster analysis to create product profiles based on which customer profiles are derived. Phase-2 then delves deep into each of the customer profiles and investigates causality behind their preference using Bayesian networks. This thesis illustrates the working of the framework using the case of Intel Corporation, world's largest semiconductor manufacturing company.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to my advisors Dr. Karl Kempf and Dr Teresa Wu for their motivation, guidance and unstinting support throughout this thesis. I am sincerely thankful to have had this opportunity to associate with them and gain from their immense knowledge and experience. I am also extremely grateful to Evan Rash, Nicholas Mason, Siddhartha Sampath and, Alex Monahan from Intel for their valuable input. Last but not the least, I want to thank my committee member Dr Feng Ju for providing me with his constructive feedback and suggestions.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 New Product Planning:

New product research and development is one of the key areas of business spending that has an enormous impact on a firm's overall business performance [1, 2]. New product introductions allow firms to remain attractive in the market and are crucial for a firm to maintain or expand its market share. A recent benchmarking study [2] suggests a strong connection between new product development and business valuation. According to this study, an average business generates 27.5% of its sales revenue from new products launched in the past three years, whereas the top 20% of businesses realize 38% of sales revenue from new products.

One of the biggest challenges that accompanies new product development is product planning and forecasting. Well before a product is launched into the market, a firm is interested in knowing the important features of the product its customers care about. This would help them to price their product appropriately and forecast its sales accurately. Predicting customer preference is essential for efficient product planning and development, which is key to a new product to succeed in the market. Typically, firms depend on judgment, domain expertise or standard market research techniques such as surveys, field studies, focus groups, etc., to predict customer preference for their new products. However, these approaches can be time-consuming, expensive and sometimes misleading. Today, with the wealth of data that is available from actual sales of previous products, firms are interested in knowing whether this data can be leveraged to extract insightful patterns in customer sales, which can be used to predict customer preference for their future products. Such an approach can be more appealing to the senior management as it is “data-driven” and based on demonstrated customer behavior. The broad objective of this thesis is to explore different data mining techniques that can be used to extract customer preference from historical sales data of previous products. The numerous challenges associated with obtaining customer preference from historical data are briefly discussed in chapter 2.

Semiconductor technology companies face the challenge of new product planning and forecasting on a regular basis, as their product features improve and evolve continuously by Moore’s law [6] (revisited in section 1.3); thus, the semiconductor industry makes a good case for this study. This thesis uses Intel, a pioneer semiconductor manufacturing company, as an example to present and validate its methodology and findings.

The rest of this chapter is organized as follows: section 1.2 gives a brief introduction to Intel and its customers, section 1.3 provides an overview of Moore’s law which sets the pace for new product introductions by semiconductor manufacturing companies in general, including Intel, section 1.4 elaborates the nuances of the business problem related to new product planning that Intel faces as a consequence of Moore’s law [6] and, its complex manufacturing process, section 1.5 introduces the solution approach to this problem proposed by this thesis, and finally, section 1.6 provides an outline of the remaining chapters in this thesis.

1.2 Intel:

Intel was founded by Robert Noyce and Gordon Moore in 1968 and today is the world's largest semiconductor manufacturing company with a 2016 revenue of $59.38 billion [3]. Intel offers a range of products for a variety of markets including desktop, mobile, server, network, and storage [3, 4]. Intel's major customers are not typically consumers but other businesses including, but not limited to [5]: Original equipment manufacturers (OEMs), original design manufacturers (ODMs) who make computer systems, cellular handsets, handheld computing devices, telecommunications and networking communications equipment, and other manufacturers who make a wide range of industrial and communications equipment.

1.3 Moore's law and Manufacturing Process Advancement:

Moore's law is an observation and forecast by Intel co-founder Gordon Moore that the number of transistors per square inch on integrated circuits (IC) doubles approximately every two years, with repeated advances of semiconductor manufacturing processes [4, 6]. The increase in transistor density implies faster, cheaper, more powerful and efficient computing. Intel has embraced Moore's law for decades, which has resulted in new and improved products being launched on a continuous basis across the different markets that it serves.

More specifically, for meeting the evolving demand of its desktop, mobile and server markets, Intel launches a new generation of processors with improved features approximately every one and a half years. Table 1 [7] provides details about past, present and future generations launched by Intel.

Table 1 Intel processor generations: past, present and future (**Source**: [7])

Generation	Release date
Presler, Cedar Mill, Yonah	2006-01-05
Merom	2006-07-27
Penryn	2007-11-11
Nehalem	2008-11-17
Westmere	2010-01-04
Sandy Bridge	2011-01-09
Ivy Bridge	2012-04-29
Haswell	2013-06-02
Haswell Refresh, Devil's Canyon	2014-05-11, 2014-06-02
Broadwell	2014-09-05
Skylake	2015-08-05
Kaby Lake	2017-01-03
Coffee Lake	2H/2017
Cannonlake	2018
Icelake	2018
Tigerlake	2019

1.4 The Business Problem

While this ongoing process and product improvement cycle has been crucial to Intel's success in maintaining its position as the market leader in the desktop, mobile and server verticals, it has also been challenging from a planning point of view. For instance, Intel's primary customers (as noted in section 1.2) include original equipment, design, and other manufacturers who build products for the end consumers or other businesses. Months before launching a new generation of processors, Intel needs to divulge information about the expected performance, specifications and, price range of the new generation processors so that its customers can plan and design their products accordingly. Another interesting aspect of the manufacturing process at Intel is its inherent stochasticity. Unlike many other products, the silicon wafers that go into a manufacturing facility at Intel result in a range of processors with different performance and feature specifications, and Intel must market this distribution efficiently to its customers. Thus, long before a new generation of processors is launched, for efficient planning and marketing decisions, the senior management at Intel needs to predict customer preference for each new type of processors that are going to be offered. However, as processor features improve and change over time, this becomes an increasingly difficult prediction exercise.

Fig. 1b displays a sample list of processors with their key features- Cores (C) and Power (W) and Speed (GHz) across six different generations for the server market's DP (dual processor) segment. As one can observe in the figure, the processor features improve as we move forward from one generation to another:

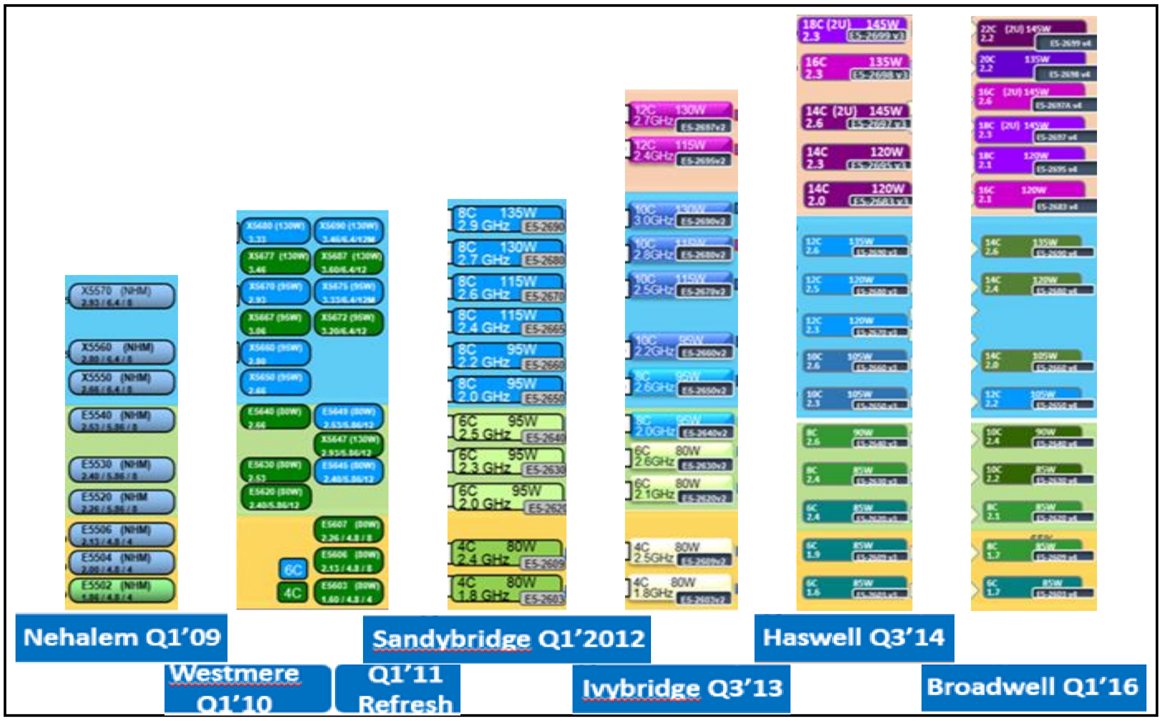


Fig. 1b: Processor feature improvement from one generation to another

1.5 The Solution Approach:

This thesis presents a decision support framework that integrates techniques and concepts from cluster analysis and Bayesian networks to extract useful patterns in customer sales from the historical data that is available from previous generations. The framework provides a holistic view on customer preference by following a two-phase procedure. In phase 1, cluster analysis is used to form customer profiles based on historical processor sales. In phase 2, Bayesian networks are used to perform causal inference on product features that drive sales for the individual customer profiles that are identified in phase 1. The subsequent chapters show how this framework can efficiently handle feature improvement from one generation to another. The potential uses of this framework include:

1. Predicting which customer is going to buy how much of what type of processors in a new generation (for production planning decisions).
2. When a new generation of processors is launched, making better product recommendations for existing customers and new customers who are similar to the existing customers (for sales and marketing decisions).
3. Detailed understanding of customer preference for making informed product development decisions for the future generation of processors.

The Intel server market's DP segment is used as an example to illustrate the framework and its utility.

1.6 Organization of Thesis:

The rest of the chapters are organized as follows:

Chapter 2 presents a brief review of the existing work in customer preference modeling and provides an overview of the techniques used in the framework, namely cluster analysis, and Bayesian networks. Chapter 3 introduces the server market's DP segment data and its features and explains the preprocessing steps that were performed before the two-phase procedure. Chapter 3 then provides a detailed elaboration of phase 1 (customer profiling) and phase 2 (feature preference elicitation) in the procedure. Chapter 4 concludes the thesis with a discussion on the potential uses of the framework.

Chapter 2

LITERATURE REVIEW

2.1 Existing Techniques in Customer Preference Modeling:

One of the most frequently used approaches in customer preference modeling is choice modeling. Choice models can be broadly classified into stated preference and revealed preference models [8–12].

Stated preference analysis extracts customer preference under experimental conditions typically by conducting customer surveys [10–11]. Popular stated preference techniques include self-explicated methods which usually take a “bottoms-up” approach, where potential customers rate individual features which are then used to determine the overall product preferences [13–16]. Another set of techniques such as the MaxDiff [17] and conjoint analysis [18-19] take a “top down” approach where customers are asked to choose between different hypothetical products by giving their relative preference for each of the products under consideration, which is then used to derive preference for individual product features [12, 14]. The major disadvantages of stated preference models is the discrepancy between what customers “state” in a survey and their actual behavior in a real market situation [12, 20–23].

Revealed preference models are based on observational data that “reveals” choices already made by the customer [8, 10-12]. The main advantage of revealed preference analysis is that it reflects the actual behavior of a customer in a real market situation. It’s a convenient form of analysis when the required data is available relieving expensive market research experiments [10, 12, 24]. The major disadvantages of revealed preference models are: its limited applicability to testing new products having attributes that are quite different from existing product attributes and the potential presence of multicollinearity as the data used is observational and not experimental [10, 12, 25]. Recently, with the advent of state of art machine learning techniques, there are numerous methods available to tackle the issue of multicollinearity. For example, [12] shows an effective way to identify key attributes of technology products from actual market data. However, most of the existing methodologies in the literature handle observational inference and little work has been done to draw causal inference on customer behavior from observational data.

2.2 Cluster Analysis:

Cluster analysis involves applying a broad range of techniques for grouping similar observations in a dataset [26–28]. Similarity (or dissimilarity) between observations are measured using a distance metric. There is a plethora of distance metrics available depending on the type of data under consideration. Some of the most common are Euclidean distance and Manhattan distance for numerical data, Jaccard dissimilarity and Hamming distance for Boolean data and, Edit distance and Damerau Levenshtein distance for string data [29]. Once a suitable distance metric is chosen, there are many algorithms that can be used for clustering the observations. The two most popular being the k-means algorithm and the hierarchical clustering algorithm. The main idea of using cluster analysis is to form meaningful clusters that make the most sense in the domain of interest [27]. There are no universally accepted rules (though there are some guidelines) for decisions regarding the choice of the algorithm, the distance metric or the number of clusters and the user typically needs to determine them experimentally [26, 27, 30].

This thesis uses cluster analysis to create processor clusters based on processor features and customer clusters based on customer preference for each of the processor clusters that are created. Here customer preference is inferred from the relative sales quantity bought from each of the processor clusters (a form of revealed prefernce).

Cluster analysis has been used extensively by a wide variety of businesses for marketing research problems to group similar products and similar customers for target marketing. [31] offers great insights on clustering techniques used in marketing research. Although cluster analysis has appeared in various applications, little has been done that caters to specific intricacies of the semiconductor market.

2.3 Bayesian Networks:

Bayesian networks are probabilistic graphical models used to represent the relationship between variables in the domain of interest [32, 33]. There are mainly 2 elements to a Bayesian network, a qualitative element and a quantitative element [34]. The qualitative element is the network structure, which is basically a directed a-cyclical graph (DAG), representing how the variables are related to each other. The quantitative element represents the probabilistic relationship between these variables as represented by the network structure, using conditional probability distributions. In other words, a Bayesian network is a compact representation of the joint probability distribution (JPD) of the variables of interest. Fig 2a depicts a Bayesian network representing the relationship between variables A, B, C and, D.

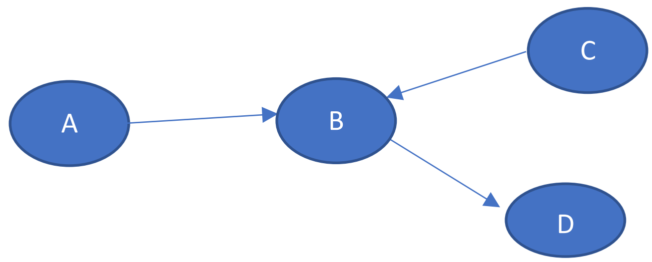


Fig. 2a Bayesian network example

In the above network, A and C do not depend on any other variable. Thus values taken by A and C can be represented by marginal probability distributions P(A) and P(C) respectively. B depends on its parent nodes A and C, and thus is represented by the conditional probability distribution P(B| A, C) while D only depends on B and is represented by the P(D|B). Finally, the JPD of the network is described as:

$$P(A, B, C, D) = P(A)P(C)\, P(B|A, C)\, P(D|B).$$

The structure of a Bayesian network can be modeled by the user entirely using his domain knowledge or can be learned from data. It is also the case that the user can start with an initial structure, and use data to learn the rest of the structure, or the user can modify a data learned structure [34]. The algorithms that are used to learn the structure of a Bayesian network can be broadly classified into two classes [34]. The first is constraint based algorithms, where links (arrows in Fig. 2a) between variables are added or deleted using statistical or correlation based tests. The second is score based algorithms which select a network after comparing scores of candidate networks that typically depends on the fit of a network to data as well as its overall complexity. Once the structure is learned, the next step (also known as parameter learning) is to learn the probabilistic relationships between the variables from data, typically using the maximum likelihood estimation approach.

One of the key features of Bayesian networks is the ability to be used as a tool to draw a causal inference from observational data [32, 34–36]. It is important to note however that Bayesian networks by default offer observational or statistical inference, like many other statistical techniques. However, under a certain set of assumptions and after careful validation by domain experts, Bayesian networks can be used for making causal inference from observational data, something which is usually considered possible only with controlled experiments [34]. There are two stages [34] to drawing causal inference from observational data: stage 1- identification, that requires determining conditions under which causal effects can be identified and listing confounders that need to be adjusted, followed by stage 2- estimation, where the identified effects are estimated after adjusting for the confounders that were listed in stage 1. There are two possible approaches to this [34, 37]:

1) The first approach involves encoding causal understating of the domain of interest in the form of a DAG and using a graphical identification criteria (such as the adjustment criteria [38]) to identify confounders (stage 1) that need to be adjusted, and subsequently using data to estimate the causal effects (stage 2) after adjusting for the identified confounders (typically using techniques such as linear regression).

This approach mandates a complete causal understanding of the domain in the form of a DAG where the link directions represent actual causal directions, as it is a prerequisite for the identification stage. The disadvantage of this approach is that it's sometimes not feasible when the number of variables involved is large or when there is no clear understanding of the entire causal structure, i.e. how all the variables are related to each other.

2) The second approach starts with the identification of confounders (stage 1) using the disjunctive cause criterion [39] and learns a Bayesian network from data. Here the sole purpose of the network is to approximate the JPD of the variables of interest, which is required for estimating the causal effects (stage 2), using techniques such as likelihood matching. The second approach does not mandate a complete understanding of the causal structure of how all the variables are related to each other (thus the arrows in the network need not imply the actual causal directions). However, this approach still requires a basic causal understanding at the variable level for applying the disjunctive cause criterion to identify confounders. This makes the second approach much more practical for real world problems.

Nevertheless, one common assumption for both these approaches is the absence of additional observed or unobserved confounding variables that can influence the domain of interest, other than the variables that are considered. If this assumption turns out to be false, the effect estimate obtained in stage 2 would be biased [34, 37].

This thesis uses Bayesian networks (using the 2nd approach) for estimating the causal effects of the each of the processor features on customer preference as measured by their relative purchase volume. This is necessary as in phase 1 of the procedure, even though customer profiles are identified based on customer preference for different processor clusters (or processor types), it does not give additional insight on the impact of individual processor features.

Bayesian networks have been used in a wide variety of domains [40] including Medicine [41–43], Engineering [44–46] and Computer Science [47–49], among others. However, previous work application of Bayesian networks in marketing research is scarce, especially for making causal inference.

\

Chapter 3
METHODOLOGY

3.1 Data

The data from Intel server market's DP segment consists of a list of processor features along with their sales by each customer. This data covers 4 different Intel generations (Table 1a) namely, Westmere (launched in Q1 2010), Sandy Bridge (launched in Q1 2011), Ivy Bridge (launched in Q2 2012) and, Haswell (launched in Q2 2013).

The data includes a total of 117 processors distributed across the 4 generations as below:

- Westmere (24)
- Sandy Bridge (31)
- Ivy Bridge (30)
- Haswell (32)

And, there is a total of 70 customers from the DP segment who have bought at least one of these processors.

3.1.1 Processor Features

The processor features that were considered can broadly be classified into 6 technical, 1 Marketing and 3 derived features. These are the most critical features that can impact the sales of a processor, as per Intel's senior management:

Technical Features

- Performance: performance value of a processor, as measured by a 3rd party software (passmark software). The performance values of the 117 processor range from 1,797 - 22,520

- Cores: No of computing cores in a processor, range: 2 - 18

- Threads: No of threads that can be processed simultaneously by Intel's hyper-threading technology, range: 4 - 36

- Base Frequency: The clock rate at which a processor performs its internal operations, range: 1.6 GHz - 3.6 GHz

- Turbo Frequency: Max overclock frequency enabled by Intel's turbo boost technology, range: 1.6 GHz - 4 GHz

- Power (TDP): Max amount of heat generated by a processor that the cooling system in a computer is designed to dissipate in a typical operation, range: 40 W-160 W

Marketing

- List Price: Recommended list price available from Intel website range: $191- $4015

Derived

- Performance/Price

- Base Frequency/Price

- Turbo Frequency/Price

The box plots in Fig. 3a below shows the range of processor feature values across the 4 generations (viz Westmere, Sandy Bridge, Ivy Bridge and, Haswell)

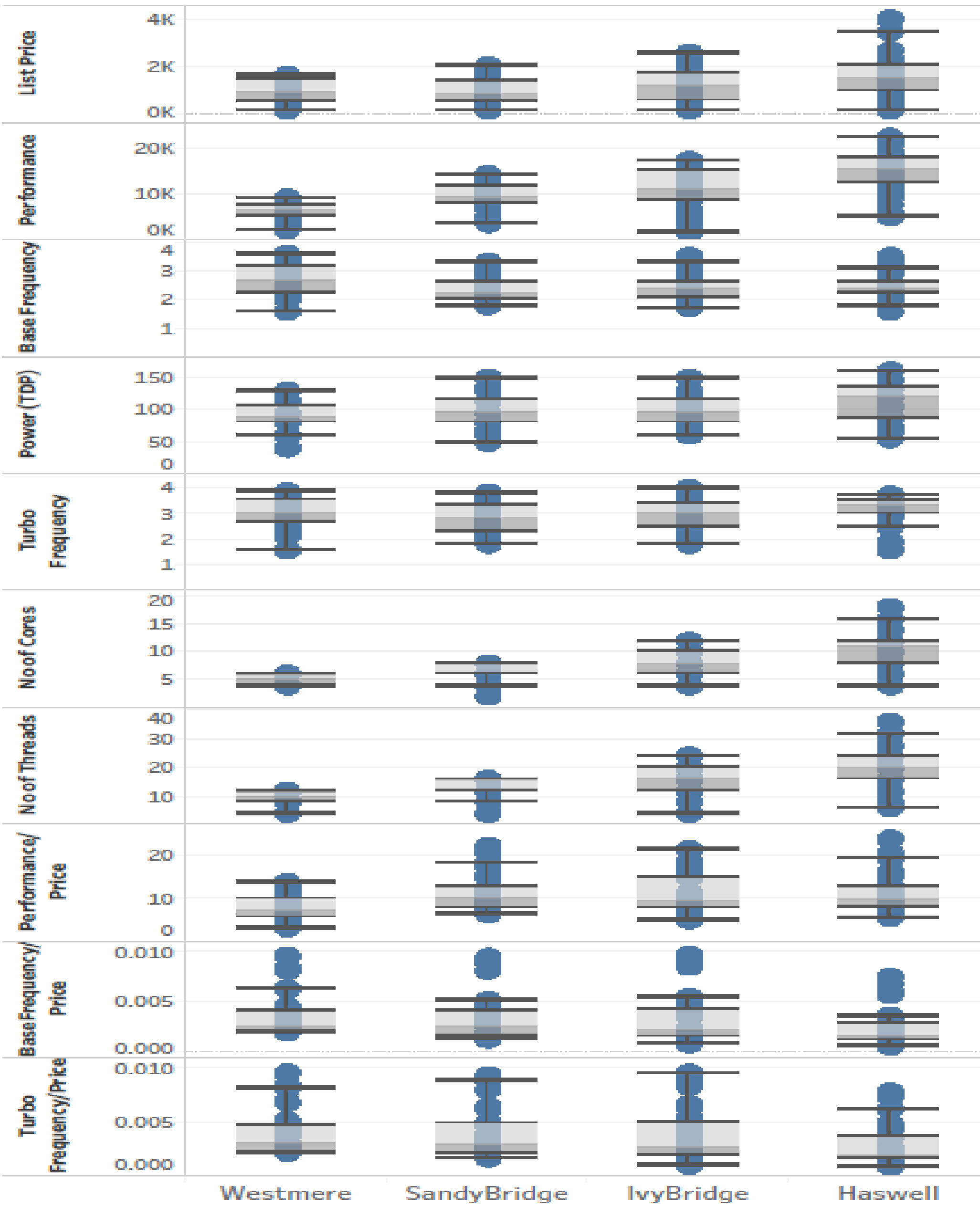


Fig. 3a Processor features

3.1.2 Feature Standardization

As the processor features improve and change from one generation to another, they need to be standardized, for a fairer comparison across generations; for instance, it does not make much sense to compare a processor with a performance value of say, 10000, from Westmere with a processor with similar performance value from Haswell. As one can observe in the “Performance” box plots in Fig. 3a, the maximum performance of processors offered in Westmere is roughly around 10000, which is less than the median performance of processors in Haswell. Moreover, for the same price, processors from Haswell offer better performance than Westmere (note the improvement in “Performance/Price” from Westmere to Haswell in Fig. 3a). Here is a specific example: Processor X5690 from Westmere, with performance value somewhere around 10000 (9171 to be exact) was priced around $1664 whereas processor E5-2623 V3 from Haswell with similar performance (9097 to be exact) was priced at only $ 444! The same argument holds for the rest of the features.

To manage feature improvement across generations, the processor features were standardized within each generation so that their values vary continuously from 0 to1, using the below feature scaling formula (formula 3.1):

$$P_{std}^{x} = \frac{P^{x} - Min_{g_P}^{x}}{Max_{g_P}^{x} - Min_{g_P}^{x}} \tag{3.1}$$

Where,

P^x_{std}: standardized value of feature *x* of processor P

P^x: Actual value of feature *x* of processor P

$Min^x_{g_P}$: Minimum value of feature x within the generation g in which processor P was launched

$Max^x_{g_P}$: Maximum value of feature x within the generation g in which processor P was launched

x: Any of the 10 features elaborated in section 3.1

P: Any of the 117 processors that are considered

g : Any of the four generations- Westmere, Sandy Bridge, Ivy Bridge, Haswell

Feature standardization also takes care of the difference in number and variety of processors from one generation to another, as we now have a single list of processor features that is generation independent, whose values vary continuously from 0-1 as shown in Fig. 3b. (Please note that, from this point onwards, wherever this report mentions "processor features," "product features" or "features" it refers to the standardized processor features, unless specified otherwise).

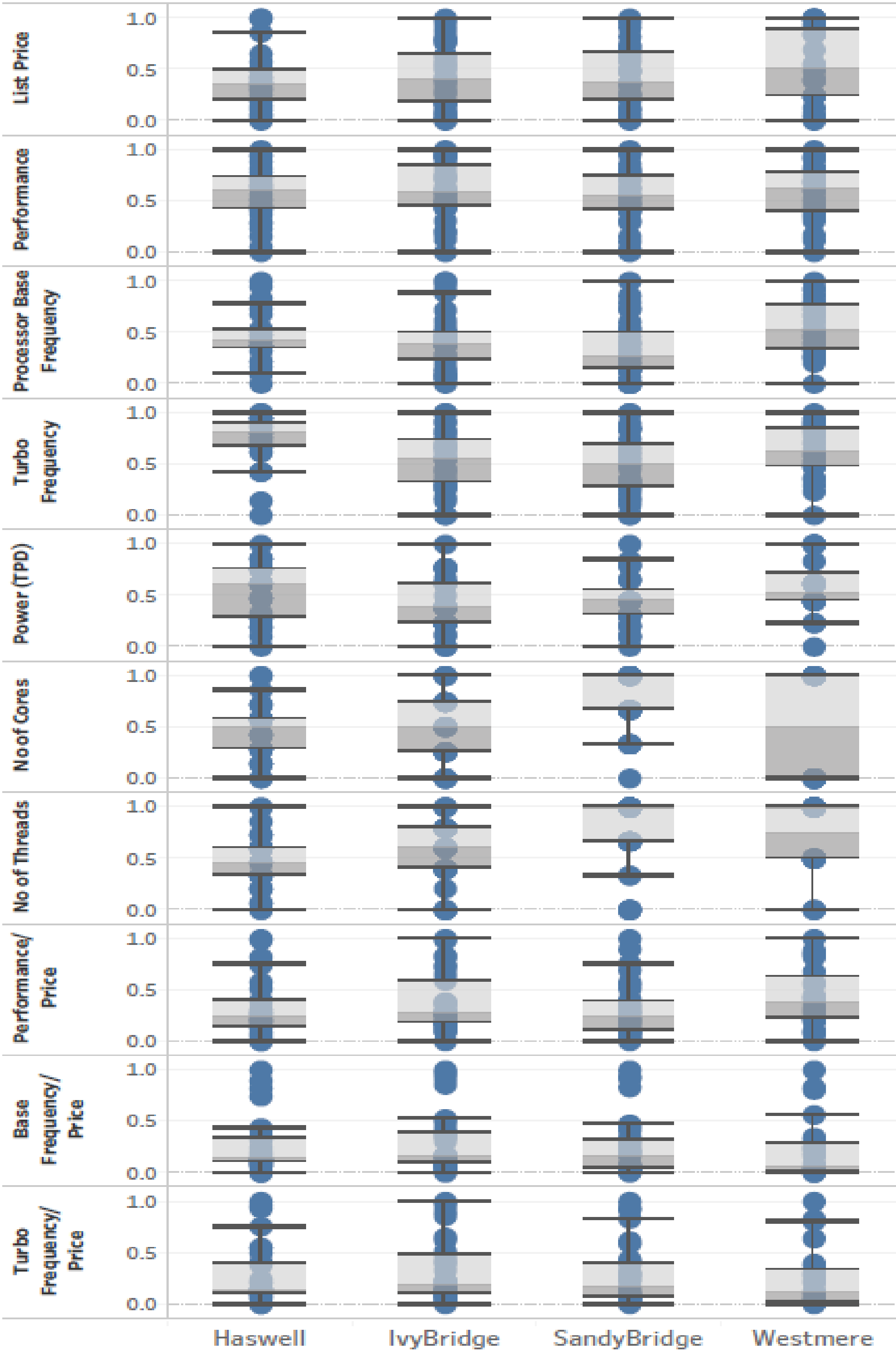


Fig. 3b Processor features after standardization

3.2 Customer Profiling

The first phase in the 2-phase procedure deals with creating customer profiles consisting of customers who have shown interest in purchasing similar processors in the past. Phase 1 consists of 2 steps: step 1-processor clustering and step 2- customer clustering as elaborated in the forthcoming sections.

3.2.1 Phase1- Step 1- Processor Clustering

Step 1 involves grouping similar processors using cluster analysis. The goal of this step is to identify similar processors types that are offered across generations. Any suitable clustering algorithm and distance metric combination can be used for this purpose. Here, the k-means algorithm [50] was used to group the 117 processors with 10 standardized features. As the standardized processor features are numeric in nature, Euclidean distance metric was chosen as the dissimilarity measure.

3.2.1.1 Choosing the Number of Processor Clusters:

In k-means cluster analysis (or any other cluster analysis for that matter), the decision regarding the choice of the number of clusters is usually influenced by two factors- the clustering quality and interpretability. A quality clustering consists of high intra-cluster similarity (observations within a cluster are similar) and/or low inter-cluster similarity (observations from different clusters are dissimilar) [51]. A useful metric that measures clustering quality is the "within cluster sum of squared errors" (also referred to as WCSS in short) shown in the formula below:

$$WCSS\ (K) = \sum_{k=1}^{K} \sum_{n=1}^{N} z_{n,k}\, \|x_n - \mu_k\|^2 \qquad (3.2)$$

Where,

$WCSS\ (K)$ represents within cluster sum of squares for K clusters

K: Number of clusters

N: Number of observations

x_n: Feature vector representing n^{th} observation of N observations

μ_k: Feature vector representing cluster center of the k^{th} cluster of K clusters

$z_{n,k}$: 1 if observation n belongs to the k^{th} cluster, 0 otherwise

$\|\ \|$: represents the Euclidean distance (the dissimilarity measure used here for clustering)

In our example, N =117, (one observation corresponding to each processor) and the length of feature vector x_n is 10 (number of processor features). The less the value of $WCSS$, the more similar the observations are within each cluster and better the clustering quality. The minimum value of $WCSS\ (K)$ is zero, which usually happens when the number of clusters K equals the number of observations N (i.e, each observation represents a unique cluster). The K means algorithm was run for a variety of values of K, ranging from 1-20, and the corresponding $WCSS\ (K)$ was computed for each K. Fig 3c shows a plot of $WCSS\ (K)$ v/s K. As it is apparent in Fig. 3c, $WCSS\ (K)$ usually decreases as the number of clusters, K increases. However, WCSS should not be used as a sole criterion to choose the number of clusters, as a greater number of clusters may not be very interpretable. Having said that, there are no universally agreed upon techniques to determine the "optimal number of clusters", as it often depends upon the problem and the domain under consideration. A commonly used heuristic for choosing the number of clusters is the elbow method [52, 53], wherein K corresponding to the "elbow point" (or the point which produces a noticeable angle in the WCSS v/s K graph- Fig. 3c) is considered to be a good choice for the number of clusters. The rationale behind this method is that there is not much marginal gain after this point (as measured by the reduction in WWCS) with additional number of clusters. Using the elbow method, 4 clusters seems appropriate. However, 5 clusters were finally chosen owing to its interpretability, as it made most sense to Intel's senior management. Section 3.2.1.2 discusses the 5 clusters in more detail.

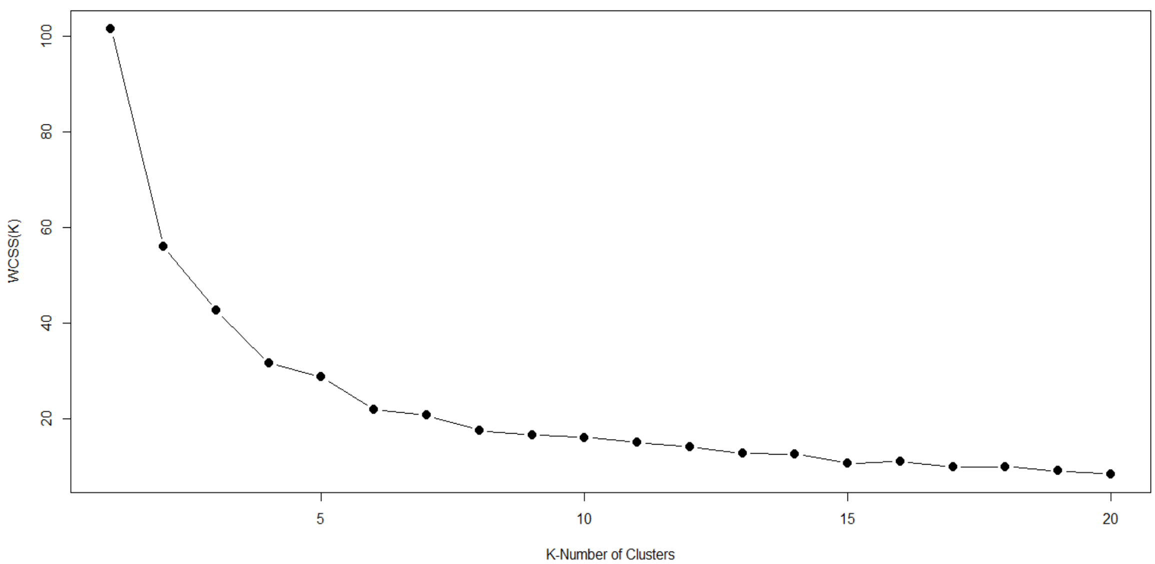


Fig. 3c WCSS(K) v/s K- Step 1

3.2.1.2 Processor Clusters

The 5 cluster centers that result from the k means-algorithm is shown in the Table 2.

Table 2 Processor Cluster Means

Processor Cluster	Cores	Threads	Turbo Frequency	Base Frequency	Performance	List Price	Power (TDP)	Performance// Price	Base Frequency/Price	Turbo Frequency /Price
Top-Bin (26)	0.93	0.94	0.82	0.58	0.9	0.82	0.74	0.14	0.03	0.05
Tier 2-Bin (40	0.73	0.76	0.48	0.25	0.59	0.41	0.36	0.28	0.13	0.16
Value-Bin (16	0.27	0.43	0.5	0.32	0.39	0.15	0.22	0.67	0.4	0.5
Low-end Bin (15)	0.12	0.02	0.23	0.23	0.1	0.03	0.32	0.77	0.91	0.92
Low Core High Power Bin (20)	0.22	0.4	0.88	0.81	0.61	0.55	0.75	0.19	0.16	0.16

The 1st cluster consists of 26 processors with superior features including a high number of cores, threads, a high performance and list price. This cluster is named as the “Top-Bin”. The 2nd cluster consists of 40 processors with advanced features, that are next in the line to the top bin processors and is named “Tier 2 Bin”. The 3rd cluster consists of 16 processors with competitive features that provide good value for money and is named the “Value Bin.” The 4th cluster named the “Lowend Bin” consists of 15 processors with a lower number of cores, threads, low performance and list price. The 5th cluster consists of 20 processors with a lower number of cores and threads. However, these processors provide good speed (base frequency and turbo frequency) and performance but consume high power (TDP) and are named the “Low Core-High Power Bin.” Fig. 3d below shows bar charts that visualizes and compares the 5-processor cluster means

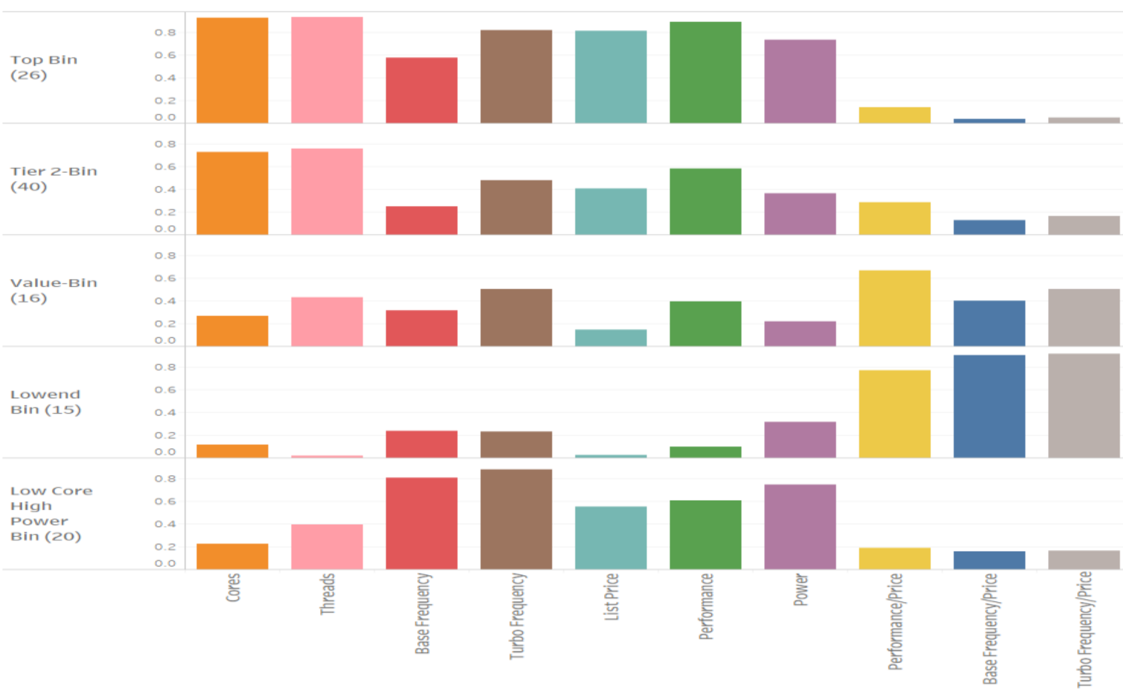


Fig. 3d Processor cluster means- comparison

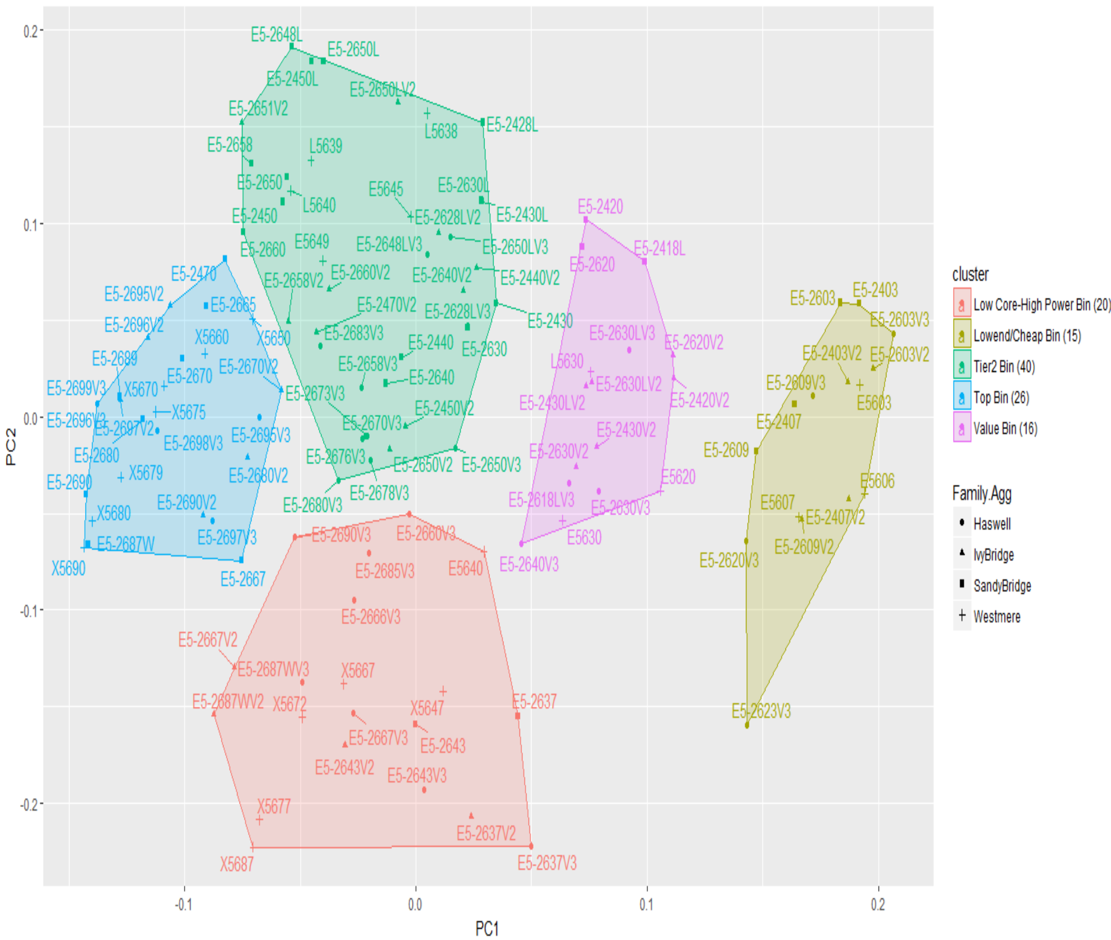


Fig. 3e Processor clusters 2-dimensional visualization using principal components analysis

Fig. 3e visualizes the 117 processors across the four generations (Westmere, Sandy Bridge, Ivy Bridge and Haswell) within their respective clusters using the first two principal components (x-axis PC1, y-axis PC2) of the processor clustering features. More information on principal component analysis can be found here [54, 55].

3.2.1.3 Validation of Processor Clusters:

Cluster validation is a challenging task since clustering is a form of unsupervised learning. Unlike supervised learning (e.g. a classifier), where we can easily validate a prediction algorithm based on how well it predicts the outcome variable of interest on unseen data, in unsupervised learning, we often don't have any external means to validate an algorithm since we typically don't know the ground truth. (i.e., we don't have an outcome variable of interest). In fact, the whole purpose of unsupervised learning is to learn structures/patterns that are inherent in the data set.

This thesis, however, attempts to "soft validate" the processor clustering step by testing it on Broadwell processors, a generation of processors that were launched after Haswell (Table 1a). The following set of experiments were performed as part of this validation exercise:

1) A list of 30 Broadwell processors that belonged to the server market's DP segment were used as an input in this exercise. The features (section 3.1.1) of the new Broadwell processors were then standardized using the formula 3.1.
2) The processor clusters of the Broadwell processors were predicted using the existing processor clustering (with processors from Westmere, Sandy Bridge, Ivy Bridge and Haswell). I.e. For each of the 30 Broadwell processors, a prediction was performed as to which of the 5 processor types (Top-Bin, Tier 2-Bin, Value-Bin, Lowend-Bin, Low Core High Power Bin) it would belong. There are mainly 2 ways to perform this prediction a) assign the Broadwell processors to their nearest cluster means (as measured by Euclidean distance) from Table 3a, or b) Train a classifier

(such as knn or random forests) on the existing processor clustering data with the processor features of the 117 processors from Westmere, Sandy Bridge, Ivy Bridge and Haswell as input variables and their respective processor clusters as the outcome variable. This trained classifier can then be used to predict the processor clusters of the Broadwell processors. This thesis uses the “cl_predict" function [56] in R, that is based on the latter approach, for predicting the Broadwell processor clusters.

3) The processor features from Broadwell were included in the processor clustering step (phase1-step1).

Observations from 2) and 3) are discussed below:

The WCSS(K) v/s K chart for phase1-step1 after adding the Broadwell processors in the clustering step is shown Fig 3f:

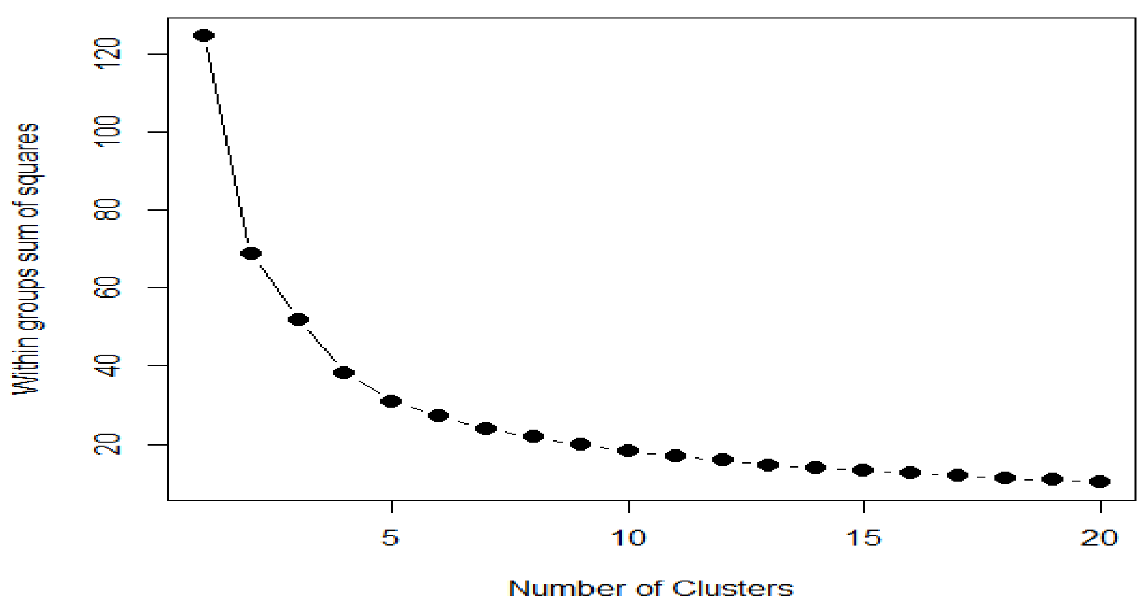


Fig 3f WCSS(K) v/s K- Step 1- after addition of Broadwell processors

Fig 3f looks very similar in structure to Fig 3c, both having comparable WCSS(K) values corresponding to K=5

The 5 cluster means after adding the 30 Broadwell processors in the clustering step are shown in Table 3.

As one can observe in Table 3, the cluster means have barely changed even after addition of the Broadwell processors to the clustering step (compare it with Table 2).

Moreover, the cluster assignments form the previous clustering remain unchanged after addition of the Broadwell processors. This can be verified by comparing Fig. 3g with Fig 3e

Fig. 3g visualizes the 147 processors across the five generations (Westmere, Sandy Bridge, Ivy Bridge, Haswell and Broadwell) within their respective clusters using the first two principal components (x-axis PC1, y-axis PC2) of the processor clustering features

Table 3 Processor cluster means after addition of Broadwell processors

Processor Cluster	Cores	Threads	Base Frequency	Turbo Frequency	Performance	List Price	Power (TDP)	Performance//Price	Base Frequency/Price	Turbo Frequency/Price
Top-Bin (26)	0.92	0.93	0.54	0.84	0.89	0.8	0.75	0.13	0.03	0.05
Tier 2-Bin (40	0.7	0.73	0.25	0.49	0.59	0.39	0.37	0.28	0.12	0.16
Value-Bin (16	0.28	0.41	0.31	0.55	0.4	0.14	0.22	0.68	0.4	0.52
Low-end Bin (15)	0.12	0.03	0.22	0.22	0.09	0.03	0.3	0.76	0.87	0.89
Low Core High Power Bin (20)	0.23	0.38	0.81	0.89	0.6	0.51	0.77	0.19	0.16	0.17

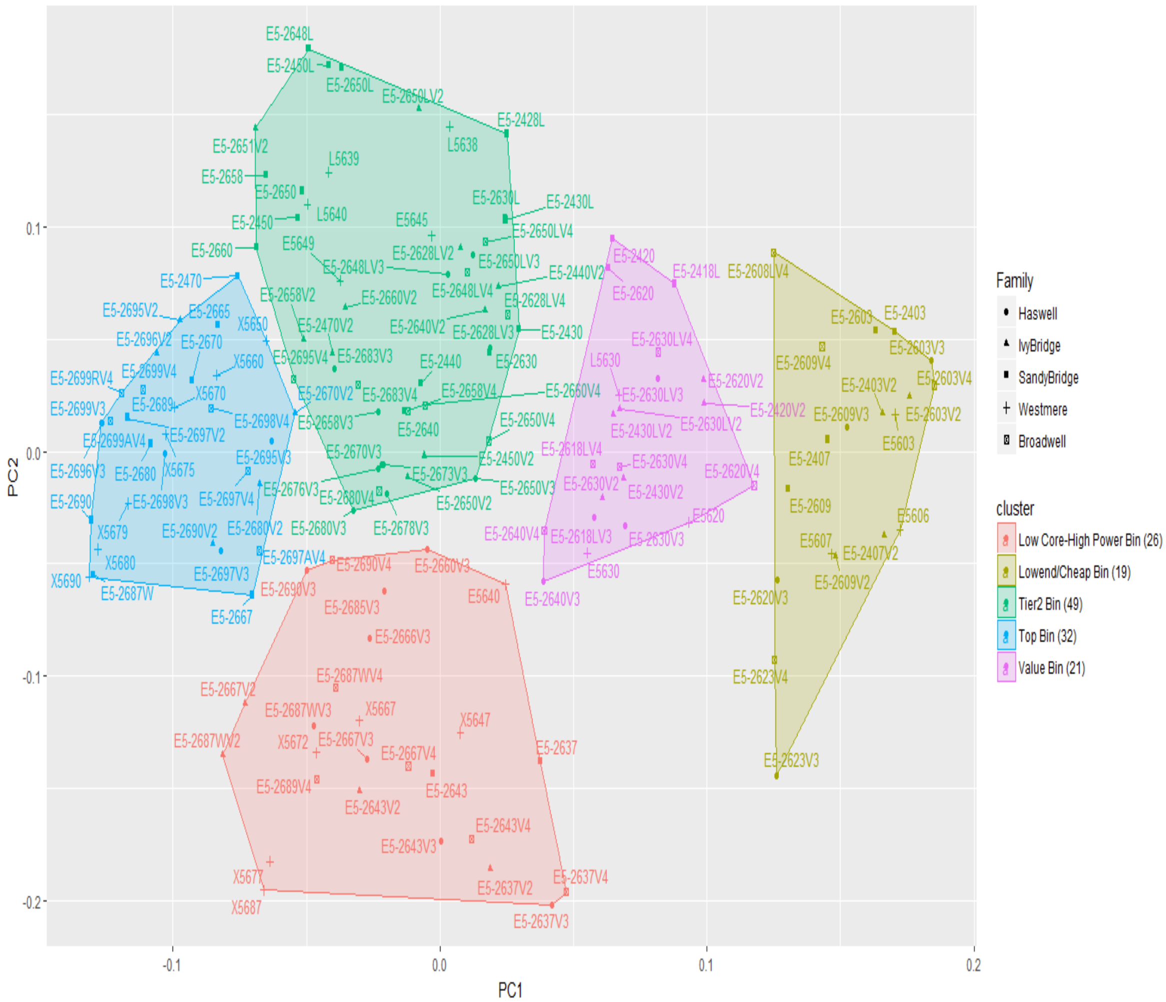


Fig. 3g Processor clusters- after addition of Broadwell processors

Table 4 compares the predicted processor clusters form 2) with the assigned clusters from 3) for the Broadwell processors. From the table, we can see that the assigned and predicted processor clusters are the same for all the Broadwell processors except a single processor- E5-2608LV4

Table 4 Broadwell processors- assigned cluster v/s predicted cluster

Broadwell Processor	**Assigned Cluster**	**Predicted Cluster**
E5-2603V4	Lowend-Bin	Lowend-Bin
E5-2608LV4	Lowend-Bin	Value-Bin
E5-2609V4	Lowend-Bin	Lowend-Bin
E5-2618LV4	Value-Bin	Value-Bin
E5-2620V4	Value-Bin	Value-Bin
E5-2623V4	Lowend-Bin	Lowend-Bin
E5-2628LV4	Tier 2 Bin	Tier 2 Bin
E5-2630LV4	Value-Bin	Value-Bin
E5-2630V4	Value-Bin	Value-Bin
E5-2637V4	Low Core-High Power Bin	Low Core-High Power Bin
E5-2640V4	Value-Bin	Value-Bin
E5-2643V4	Low Core-High Power Bin	Low Core-High Power Bin
E5-2648LV4	Tier 2 Bin	Tier 2 Bin
E5-2650LV4	Tier 2 Bin	Tier 2 Bin
E5-2650V4	Tier 2 Bin	Tier 2 Bin
E5-2658V4	Tier 2 Bin	Tier 2 Bin
E5-2660V4	Tier 2 Bin	Tier 2 Bin
E5-2667V4	Low Core-High Power Bin	Low Core-High Power Bin
E5-2680V4	Tier 2 Bin	Tier 2 Bin
E5-2683V4	Tier 2 Bin	Tier 2 Bin
E5-2687WV4	Low Core-High Power Bin	Low Core-High Power Bin
E5-2689V4	Low Core-High Power Bin	Low Core-High Power Bin
E5-2690V4	Low Core-High Power Bin	Low Core-High Power Bin
E5-2695V4	Tier 2 Bin	Tier 2 Bin
E5-2697AV4	Top-Bin	Top-Bin
E5-2697V4	Top-Bin	Top-Bin
E5-2698V4	Top-Bin	Top-Bin
E5-2699AV4	Top-Bin	Top-Bin
E5-2699RV4	Top-Bin	Top-Bin
E5-2699V4	Top-Bin	Top-Bin

From the above observations, it's reasonable to conclude the following:

I. The processor clustering with 5 cluster centers is very robust and the new generation processors from Broadwell blends nicely with the existing clusters.

II. Previous generation processor clustering can provide quite a reliable prediction of the new generation processor clusters

It should be noted at this point that the Broadwell processor features are used in this thesis only for validating the processor clustering step. Rest of this thesis uses data from the 117 processors spanning 4 generations- Westmere, Sandy Bridge, Ivy Bridge, and Haswell discussed in section 3.1.

3.2.2 Phase 1- Step 2 Creating Customer Clusters

Phase 1- Step 2 groups customers based on their preference levels for the different product clusters identified in step1. This thesis uses the relative purchase quantities of different products that are available from the customer sales data, as an indicator of preference (or revealed preference, to be more precise). However, caution must be exercised before engineering the revealed preference of customers from sales data, and ideally, it should be validated with domain expertise. In the Intel example, the relative purchase quantities (in %) bought by a customer across the five different processor types (Top-Bin, Tier 2-Bin, Value-Bin, Lowend-Bin, Low Core High Power Bin) are used to infer their level of preference for these processor types. For instance, if customer A bought 5000 processors from the Westmere generation, of which 4000 belong to the "Top-Bin", 200 belong to "Tier 2-Bin", 100 belong to the "Value-Bin", 50 belong to the "Lowend-Bin, " and 150 belong to the “Low Core High Power Bin”, then customer A’s preference levels (on a scale of 0-100) for these processor types are inferred as 80, 4, 1 and 3 respectively. Similarly, the preference levels of the same customer for the five processor types are computed from the relative purchase quantity data available from other generations (Sandy Bridge, Ivy Bridge and Haswell) as well. The resulting generation wise preference levels are then averaged to obtain an overall aggregate preference level of Customer A for these processor types (Please note that even though this thesis uses averaging as a method of aggregation, any other suitable aggregation technique can be used for this purpose). Fig. 3h depicts this computation for an anonymized customer X, where the generation-wise relative purchase quantities are averaged to obtain the aggregate preference levels of Customer X for the five different processor types.

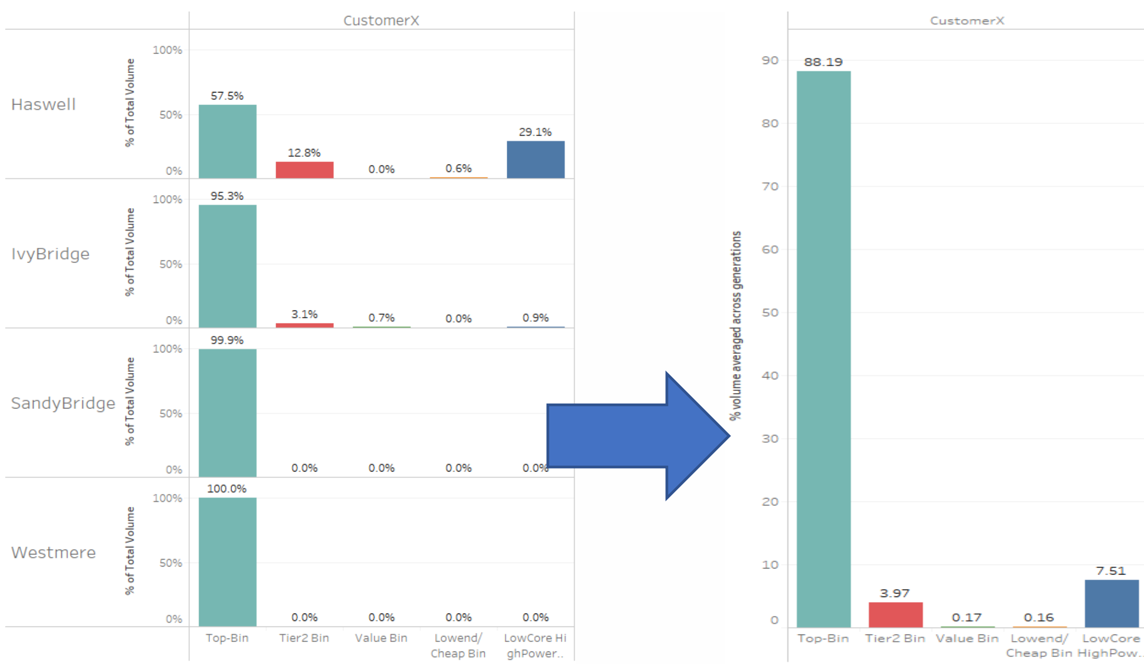


Fig. 3h Aggregate preference level computation

Once the aggregate preference levels are computed for each of the 70 customers, cluster analysis can again be used to identify customers with similar aggregate preference levels for the five processor types. The data used for clustering in step 2 of phase 1 consists of the 70 customers as observations and their respective aggregate preference levels for the five processor types as clustering variables. Again, k means algorithm with Euclidean distance metric was used for clustering the customers.

3.2.2.1 Choosing the Number of Customer Clusters

Fig. 3i shows WCSS(K) v/s K for the customer clustering step. Here the elbow point is not that obvious. Nevertheless, 7 clusters were chosen to owe to its high quality (relatively small WCSS) and interpretability (as it made most domain sense to Intel's senior management). Section 3.2.2.2 discusses the customer clusters in more detail.

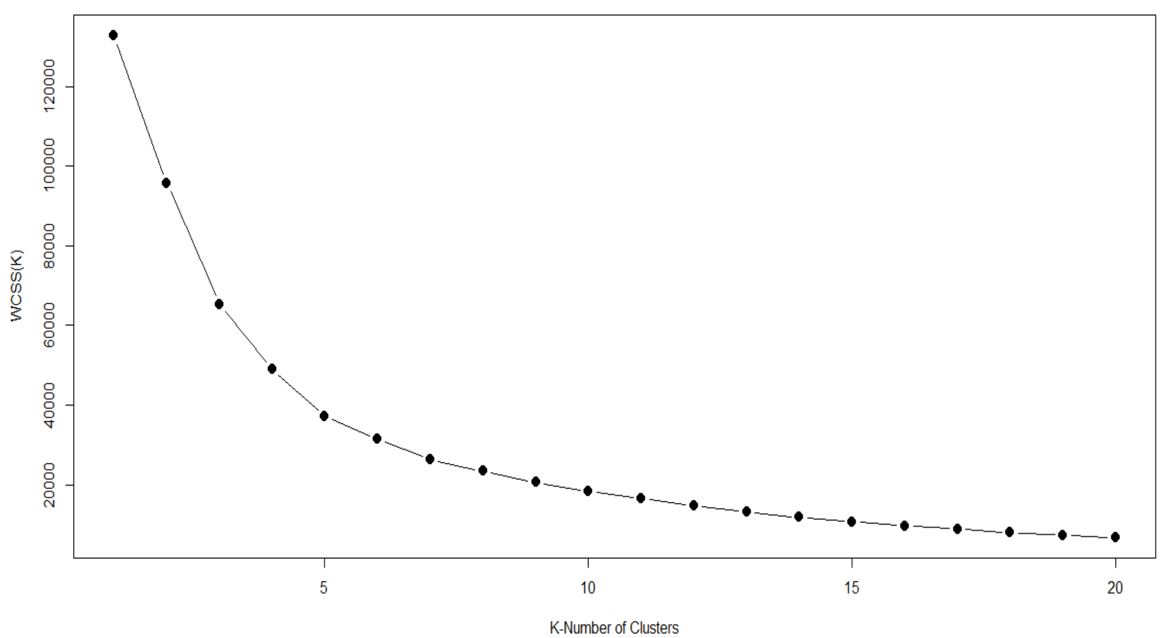


Fig. 3i WCSS(K) v/s K -Step 2

3.2.2.2 Customer Clusters

Table 5 below shows the 7-customer cluster means (rounded to the nearest integer). The first cluster consists of 4 customers who have shown a high preference towards purchasing processors belonging to the “Top-Bin” processor type, with an aggregate preference level mean of 81 for the “Top-Bin” processors, across the 4 generations that were considered (Westmere, Sandy Bridge, Ivy Bridge and Haswell). This cluster is named as the “Top-Bin Cluster.” Similarly, the “Tier 2 Cluster” consists of 9 customers who have shown very high preference towards purchasing the “Tier 2 Bin” processors (with an aggregate preference level mean of 81), the “Value-Cluster” has 6 customers who have shown a high propensity towards purchasing the “Value Bin” processors (with an aggregate preference level mean of 75). The “Value and Tier 2 Cluster” consists of 12 customers who have shown preference towards purchasing the “Value-Bin” and the “Tier 2-Bin” processors, with an aggregate preference level mean of 46 and 36 respectively, similarly the “Top Bin and Tier 2 Cluster” consists of 10 customers who have shown interest in purchasing the “Top Bin” (aggregate preference level mean of 46) and “Tier 2-Bin” (aggregate preference level mean of 33) processors. The “Lowend Cluster” consists of 5 customers who have shown a high preference (aggregate preference level mean of 66) towards purchasing processors from the “Lowend Bin”.

Table 5 Customer cluster means

Customer Cluster	Top-Bin	Value-Bin	Lowend-Bin	Tier 2-Bin	Low Core High Power Bin
Top Bin Cluster	81	13	0	3	2
Tier 2 cluster	4	4	6	81	6
Value Cluster	15	75	4	5	1
Top Bin and Tier 2 Cluster	46	10	5	33	5
Value and Tier 2 Cluster	7	46	10	36	1
Lowend Cluster	3	16	66	16	0
Everything Cluster	19	28	20	22	11

Finally, the “Everything Cluster” consists of 24 customers who haven’t shown a strong preference towards purchasing any one of the five processor types, and tend to purchase processors evenly across the five processor types. An interesting takeaway here is that none of the customer clusters have shown a strong preference towards purchasing the “Low-Core High Power Bin” processors. Fig 3.j below shows pie charts that compares the 7-customer cluster means.

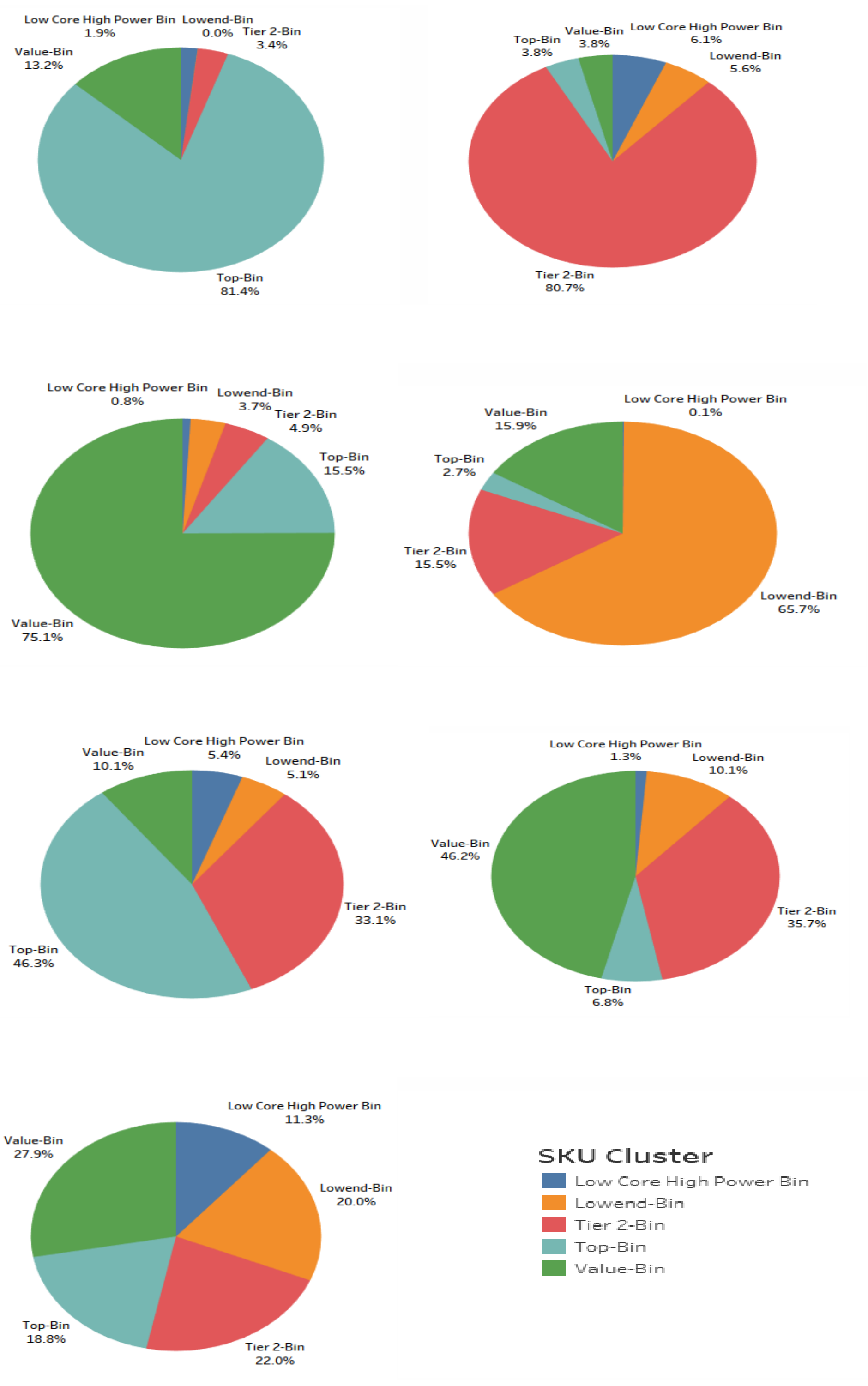
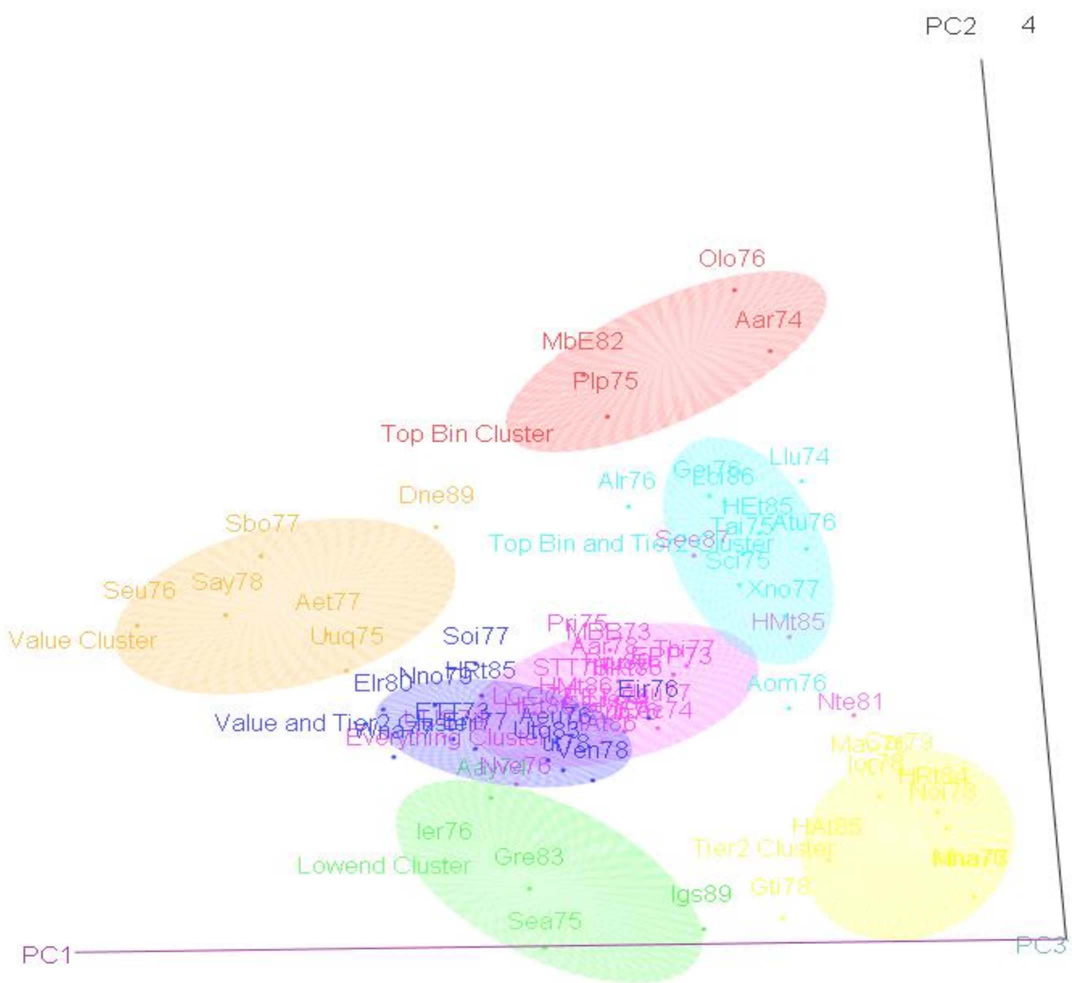
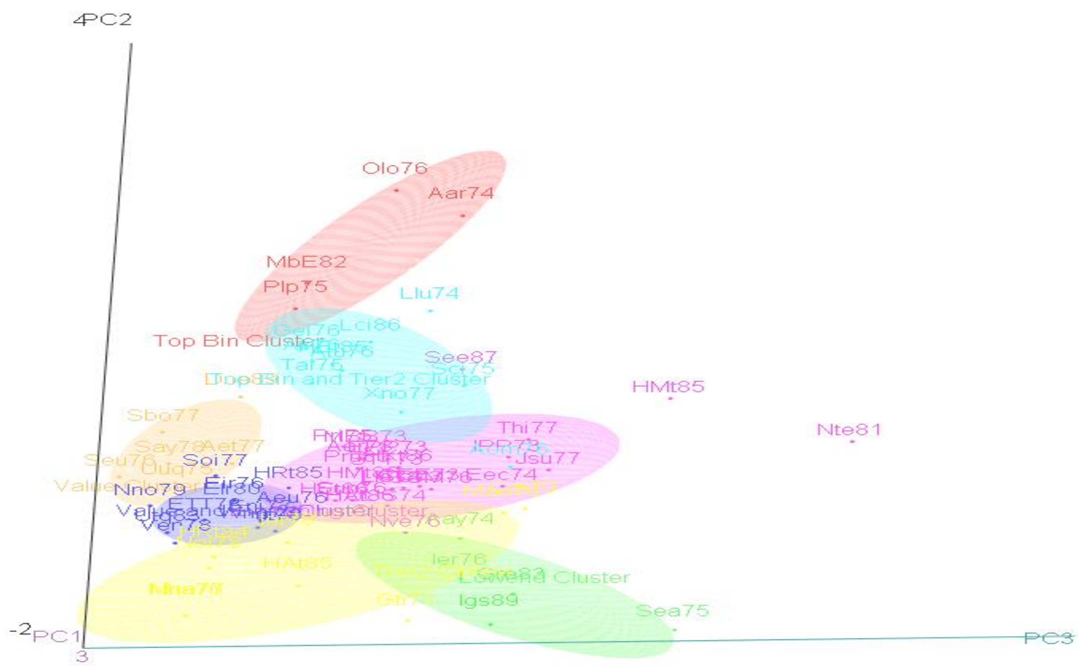


Fig. 3j Customer cluster means-comparison

Fig. 3k below visualizes the 70 customers, within their respective clusters using the first three principal components (x-axis PC2, y-axis PC3, z-axis PC1) of the step 2 clustering features.

Fig. 3k Customer clusters 3-dimensional visualizations using principal components analysis




PC2
PC1
PC3
Olo76
Aar74
MbE82
Plp75
Llu74
Top Bin Cluster
See87
HMt85
Nte81
Xno77
Sbo77
Thi77
Soi77
HRt85
Jsu77
Eec74
Nno79
Aeu76
Nve76
Ver78
Ier76
Lowend Cluster
Igs89
Sea75

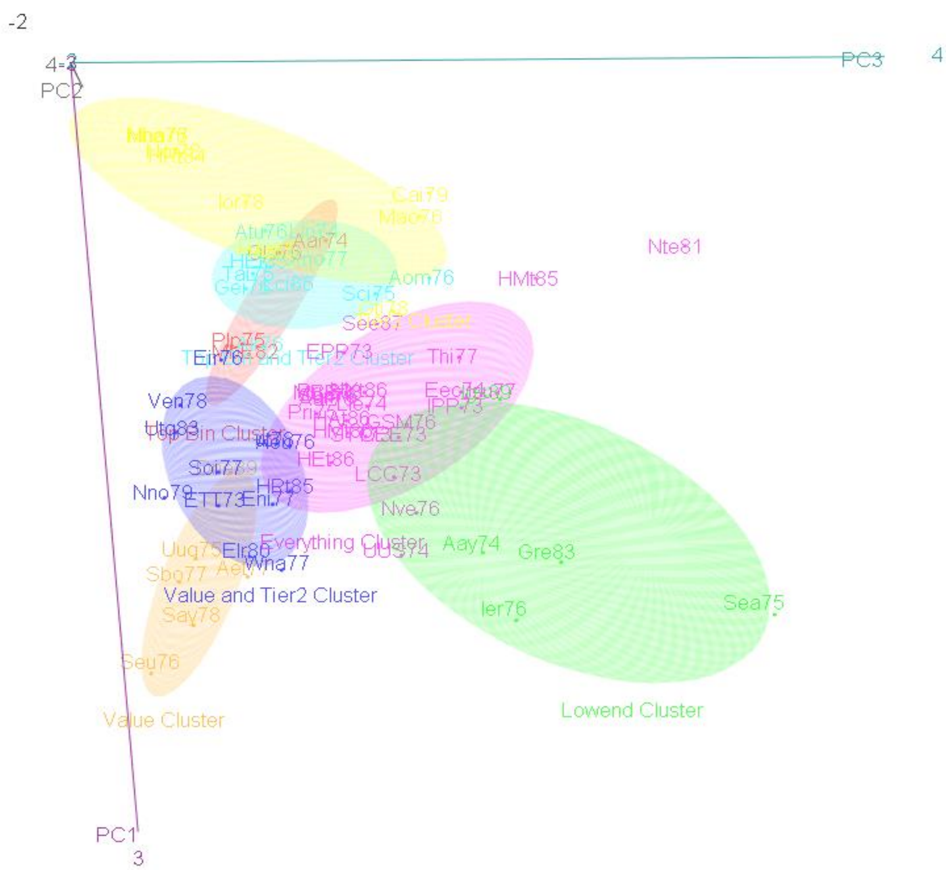

PC3
PC2
PC1
Nte81
HMt85
Aom76
Sci75
EPP73
Thi77
Ven78
Top Bin Cluster
Soi77
HRt85
HEt86
LCG73
Nno79
ETT73
Nve76
Everything Cluster
Aay74
Gre83
Elr80
Wna77
Value and Tier2 Cluster
Say78
Ier76
Sea75
Seu76
Value Cluster
Lowend Cluster

Fig 3l below shows the sales and revenue contributions by each of the 7 customer clusters across the four generations- Westmere, SandyBridge, Ivy Bridge and Haswell. As apparent in the figure, a major share (>50%) of both sales and revenue is contributed by customers belonging to "Everything Cluster".

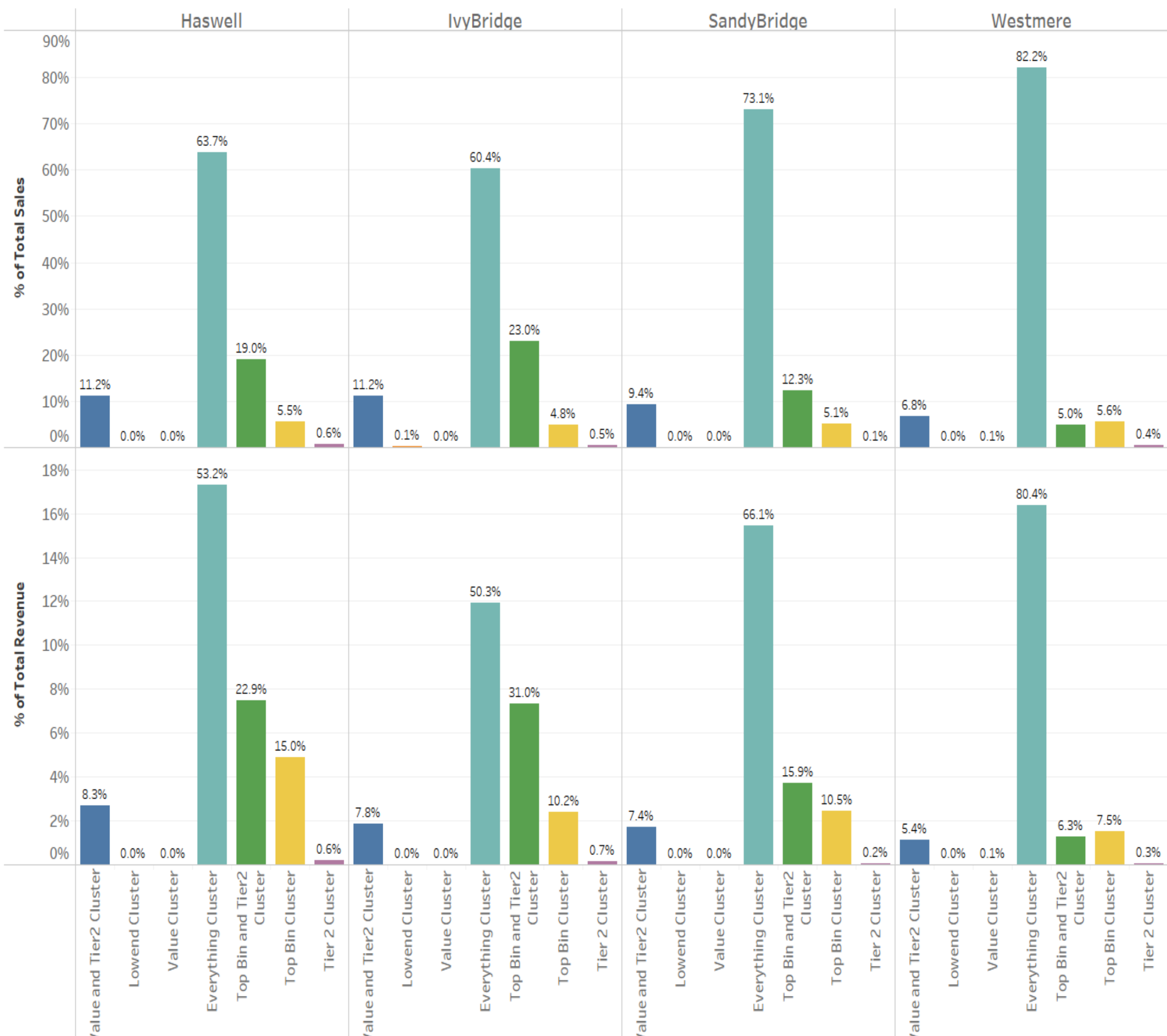


Fig. 3l Customer Clusters – Sales and Revenue % Contributions

3.2.3 Customer Profiles

Clustering customers based on their preference levels provide an insightful way to analyze their purchasing behavior across generations. For example, Fig. 3m below shows customers belonging to the "Top-Bin Cluster".

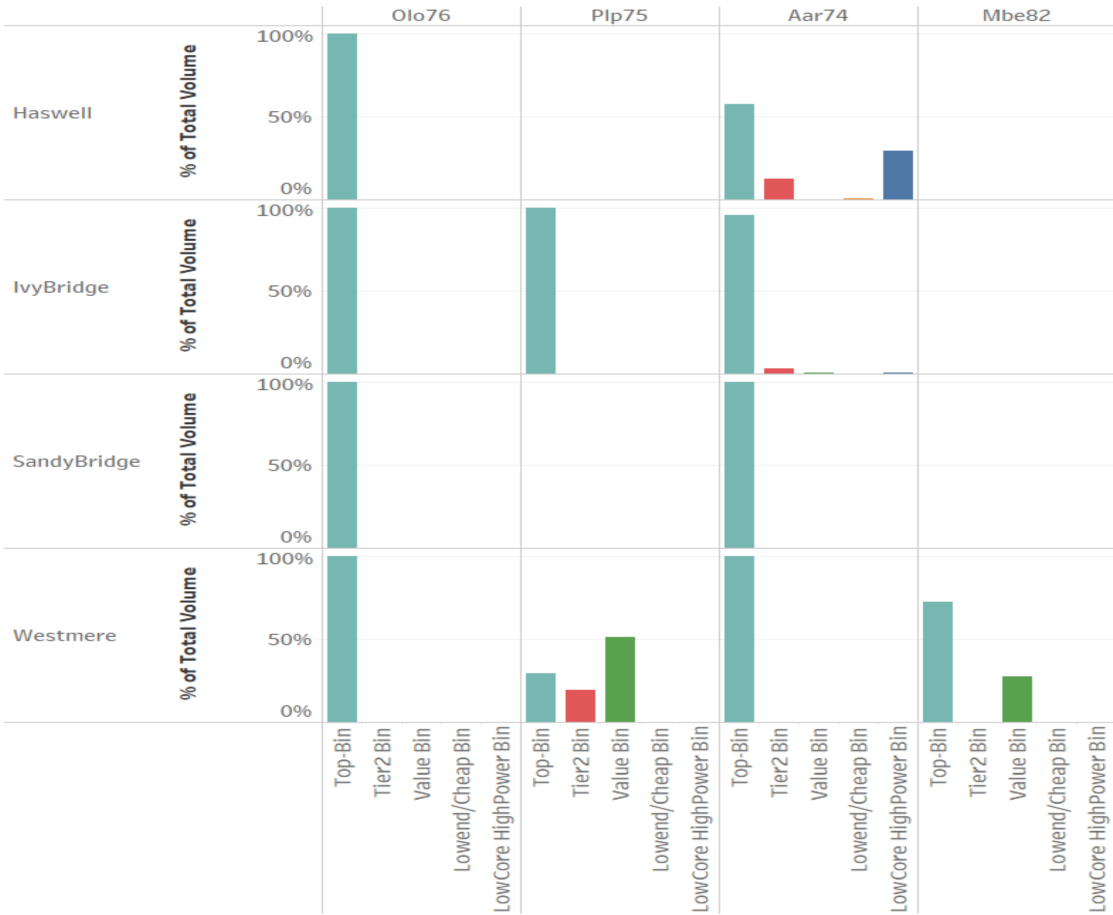


Fig. 3m Customer Profile- Top-Bin Cluster

The figure represents the generation wise relative distribution of the volume (in %) bought by the "Top-Bin Cluster" customers across the five different processor types. As one can observe in the figure, all the 4 customers belonging to this cluster have historically shown a strong preference towards purchasing the "Top-Bin" processors. Another interesting pattern that we can observe among customers within the "Top-Bin Cluster" is that their relative distribution of purchase volumes across the 5 processor types stays more or less the same in each of the 4 generations, even as the number, variety and, features of these processors change from one generation to another. This is a useful insight for Intel, as it better informs their planning and marketing decisions concerning the "Top-Bin Cluster" customers for future generations. For instance, Fig. 3n below provides a summarized version of Fig. 3m, with the relative % volumes averaged across the 4 generations.

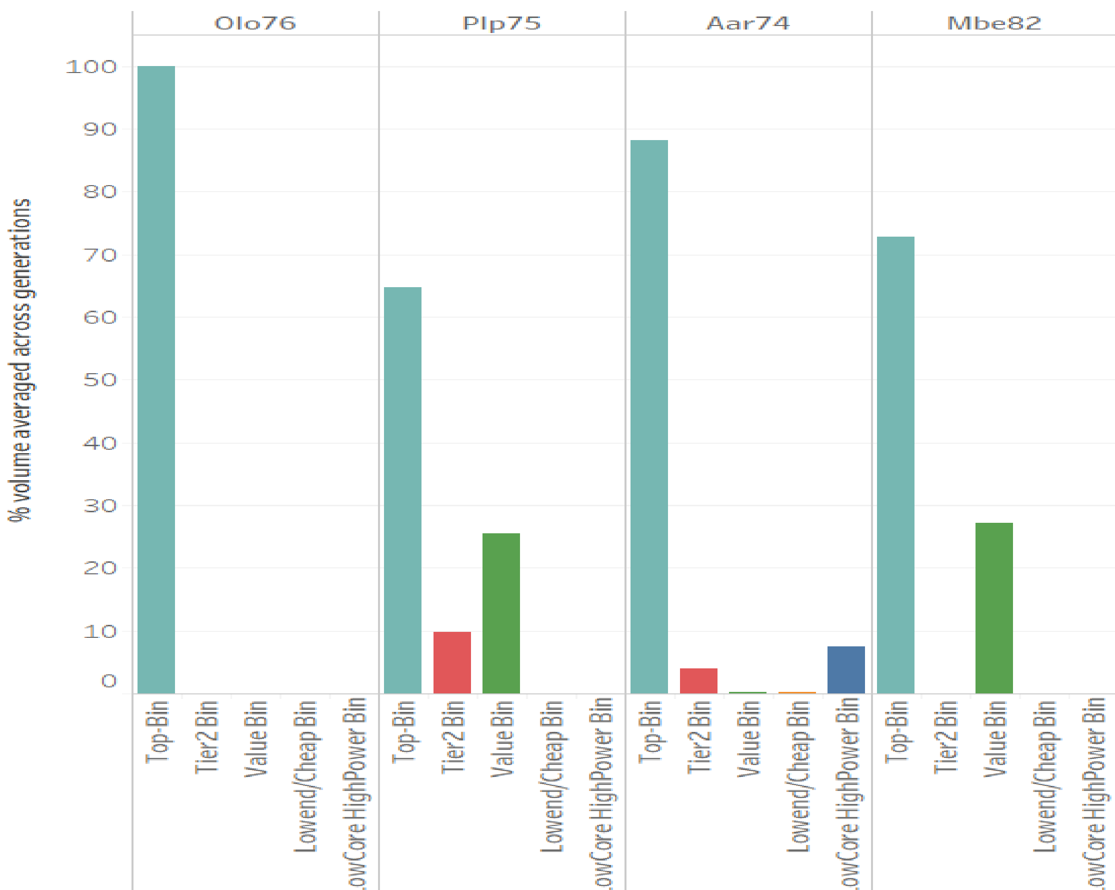


Fig. 3n Customer Profile Summary- Top-Bin Cluster

Consider customer "Mbe82" (anonymized) as an example. From Fig. 3m, we learn that Mbe82, historically, has purchased an average of ~ 70% of their total volume from the "Top-Bin" and the rest (~ 30%) from the "Value-Bin". Now when Intel launches a new generation of processors, and we expect Mbe82 to purchase a total of say, 5000 units from this new generation, this knowledge can be used to forecast how much Mbe82 is going to purchase from each of the five different processor types in this new generation of processors. Thus a good forecast based on the knowledge we have on Mbe82 would be 3500 units (70% of 5000) from the "Top-Bin" and 1500 units (30% of 5000) from the "Value-Bin" processors and 0 units from the rest. Another use of customer profiles is as a recommender system. For instance, the knowledge that the "Top-Bin Cluster" customers have historically shown a high propensity towards purchasing the "Top-Bin" processors can be used to recommend them specifically the processors that belong to the "Top-Bin" category in a new generation of processors (content-based filtering). Now when Intel attracts a new customer in future, and with subsequent market research studies, if it can be shown that they are similar to an existing customer(s) who belongs to the "Top-Bin Cluster"; this information can still be used to recommend processors to this new customer (collaborative filtering), even if Intel does not have any prior transaction history with this customer. Fig. 3o and Fig. 3p presents similar charts for customers belonging to "The Tier 2 Cluster". The customer profile charts of the remaining 5 customer clusters are included in the appendix.

Fig. 3o Customer Profile- Tier2-Bin Cluster

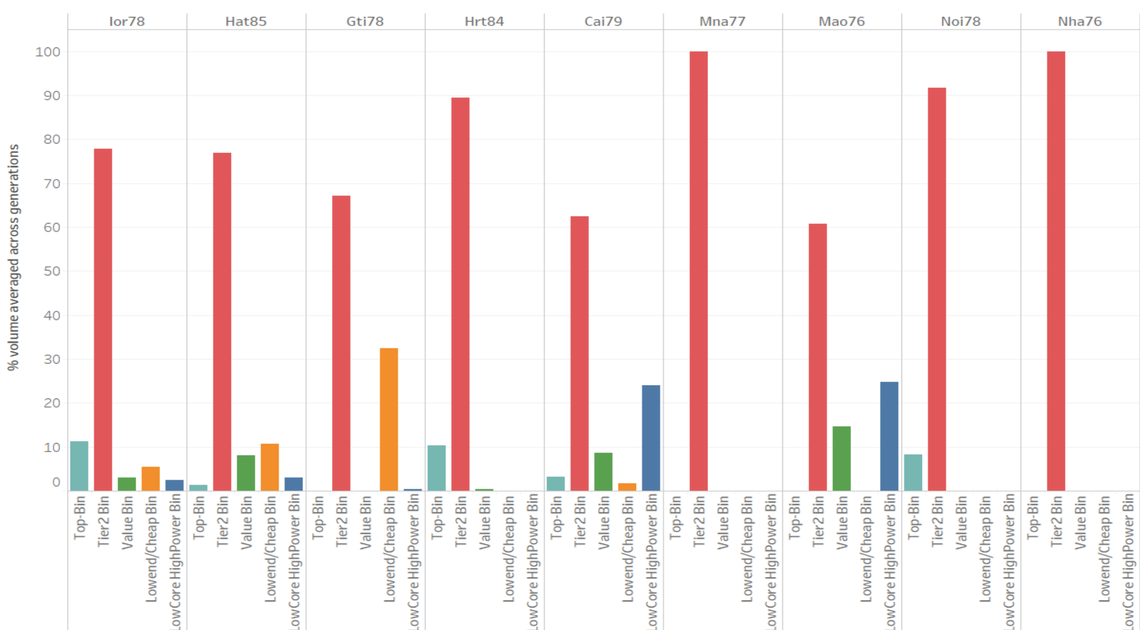


Fig. 3p Customer Profile Summary- Tier2-Bin Cluster

3.3 Phase 2- Feature Preference Elicitation

Even though phase-1 (customer profiling) in the 2-phase procedure helped us discover groups of customers who are interested in purchasing similar types of processors, it doesn't provide us with any additional insight on why they are interested in purchasing these processor types. For instance, the "Top-Bin" processor type consists of processors with a high number of cores and threads, high performance, high turbo frequency etc. From the customer profiling results (phase 1), we know that the "Top-Bin Cluster" customers have shown high preference towards purchasing the "Top-Bin" processors. Phase 1 however, does not tell us anything about the impact of individual processor features on their preference; like do they value "Cores" more than "Performance"? and if yes, by how much?

Phase 2 in the procedure attempts to answer these questions by quantifying the causal effects of individual features on preference of each of the 7 customer clusters identified in phase 1. As discussed in Chapter 2, phase 2 uses Bayesian networks for making causal inference from observational data that is available in the form of processor sales by each customer.

3.3.1 Phase 2 Illustration

Perhaps, phase 2 and its purpose can be best explained with the case of the "Everything Cluster" customers. The "Everything Cluster" consists of 24 customers who haven't shown a strong preference towards purchasing any one of the 5 processor types and, the majority of both sales and revenue form the DP segment is contributed by the customers belonging to the “Everything Cluster” (Fig 3l). Thus, it’s both important and interesting to learn which of 10 processors features they value the most, by quantifying the impact of each of the these features on their preference. Again, the data considered consists of 117 processors across the 4 generations (Westmere, Sandy Bridge, Ivy Bridge, Haswell). Here, the 10 processor features are considered as potential treatment variables that can have an impact on the response variable of interest- the “Preference Score”. The “Preference Score” for a processor *P* by a customer *C* is essentially its volume purchased by customer *C*, standardized within the generation in which *P* was launched (Once again, a function of the relative purchase quantity of a processor bought by a customer is assumed as an indicator of their preference for that processor). The “Preference Score” is computed for each of the customer-processor combination in the “Everything Cluster” using formula 3.3 below, where $SV_P^c > 0$. Note here that formula 3.3 is same in structure as formula 3.1 except for a few notational changes.

$$PS_P^c = \frac{SV_P^c - MinSV_{gP}^c}{MaxSV_{gP}^c - MinSV_{gP}^c} \tag{3.3}$$

Where,

PS_P^c: Preference Score for processor *P* by customer *C*

SV_P^c: Volume of processor *P* purchased by customer *C*

$MinSV_{g_P}^{c}$: Minimum individual processor purchase volume by customer C in the generation g in which processor P was launched

$MaxSV_{g_P}^{c}$: Maximum individual processor purchase volume by customer C in the generation g in which processor P was launched

C: Any of the 24 customers in the "Everything Cluster"

P: Any of the 117 processors that were considered

g: Any of the four generations- Westmere, Sandy Bridge, Ivy Bridge, Haswell

Thus, to summarize, the observational data from the "Everything Cluster" customers used in phase 2 consists of 10 processor features that can potentially impact the variable of interest- the "Preference Score", PS_P^c. As discussed in chapter 2, phase 2 uses the 2nd approach to causal inference from observational data using Bayesian networks. It starts with identification of confounders using the disjunctive cause criterion [39] in stage 1, and then learns a Bayesian network from the data, for estimating causal effects using Jouffe's likelihood matching algorithm [34, 37] in stage 2. This thesis uses BayesiaLab, a commercial software, for implementing phase 2.

Before moving forward, it's important to list the main assumptions for estimating these causal effects. It is critical to note at this point that these assumptions are domain specific and must be backed by domain expertise [34, 37]:

1) The “Preference Score”, PS_P^C is an accurate reflection of preference of customer C for processor P.
2) There are no other observed or unobserved confounders that affects this domain (10 processor features + the response variable: “Preference Score”). As discussed earlier in Chapter 2, this is an important assumption and if it turns out to be false, the effect estimates obtained would be biased.
3) In stage 1, using the disjunctive cause criterion for confounder selection, when estimating the effect of each of the 10 processor features on “Preference Score”, the other 9 processor features are listed as confounders as all the 10 processor features here can potentially impact the “Preference Score”.

Now, stage 2 involves learning a Bayesian network form the observational data we have, for estimating the causal effects. Since we are dealing with continuous variables (both processor features and the Preference Score, varying continuously from 0-1), and since we don’t have any prior assumptions regarding any functional forms that control these variable distributions, it makes sense to learn a non-parametric Bayesian network from data, where the continuous variables are discretized (binned) as a preliminary step before a network is learned from data.

The discretization of variables can be done both manually as well as using a suitable algorithm. However, it should be validated with domain expertise and one should also take into consideration aspects regarding the computation complexity, the number of observations available, etc. before the discretization step. For instance, the score based structural learning algorithms in BayesiaLab [34] are based on the Minimum Description

Length score (MDL), which depends on the strength of the relationships between variables as well as the overall network complexity. Here, Increasing the number of discrete states per variable increases the complexity of the resulting conditional probability distributions between variables, thus requiring more data to find relationships that can compensate for the additional complexity.

For our purpose, the 10 processor features and the response variable- “Preference Score” were discretized into 5 bins, using the “K-Means Discretization” module in BayesiaLab (which runs a k-means clustering algorithm underneath for splitting each of the variables into 5 bins). The variable distributions after the discretization step is as shown in Fig. 3q below.

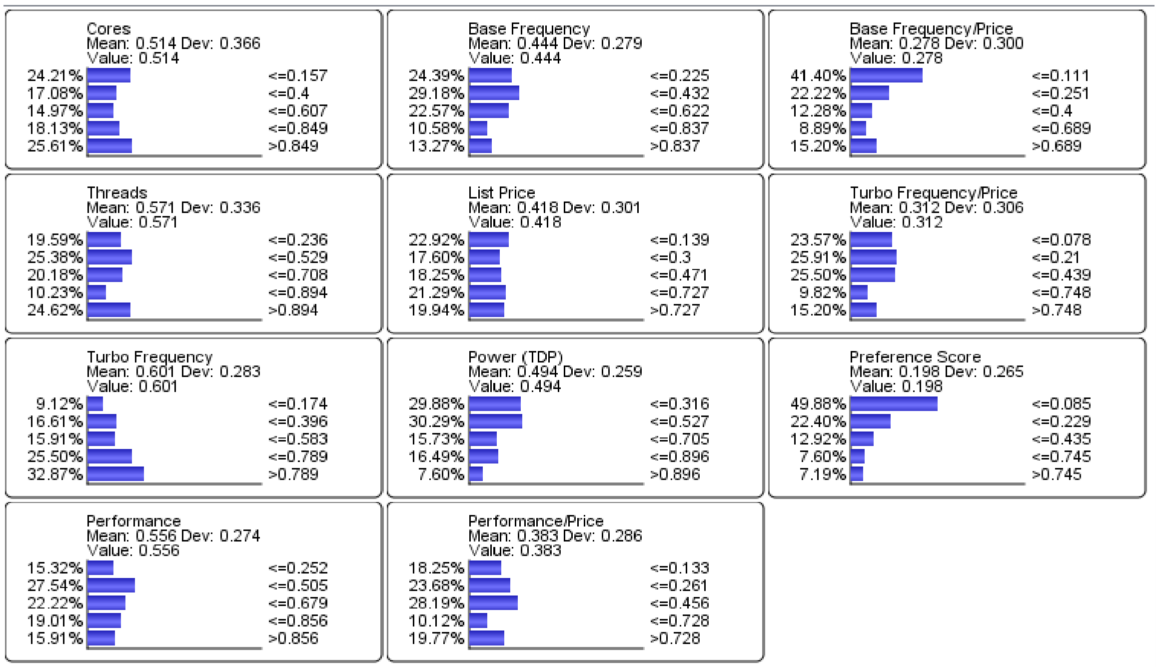


Fig. 3q Phase 2 variables after discretization

After all the variables were discretized, a Bayesian network was learned from data. As discussed chapter 2, the sole purpose of this network is to represent the joint probability distribution (JPD) connecting the variables in the network, and the link directions does not necessarily mean causality. An augmented naïve Bayes structure is learned from data, and the conditional probability distributions relating these variables are estimated using maximum likelihood estimation. Fig. 3r below shows the network structure.

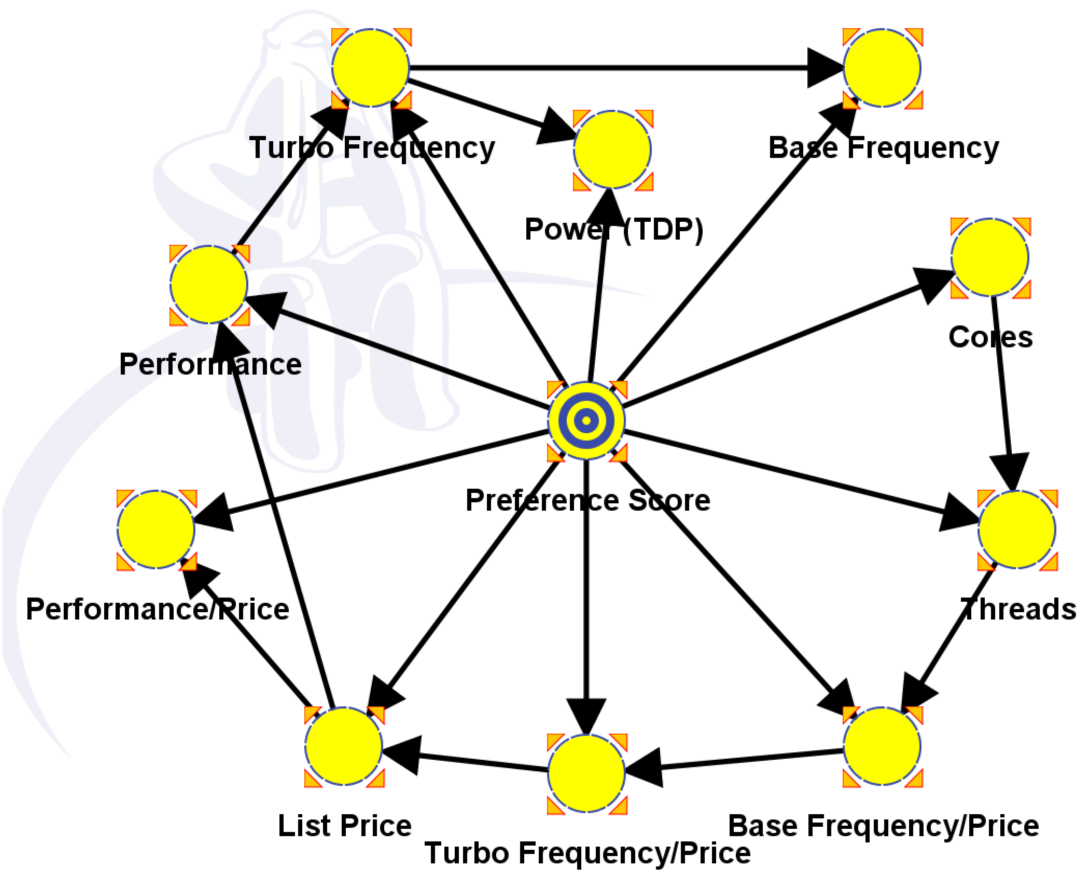


Fig. 3r The Bayesian network structure

Before moving ahead and estimating the effects of the 10 processor features on the "Preference Score", its perhaps useful to gather some insight on the working of Jouffe's likelihood matching algorithm (or LM algorithm) [34, 37], which runs on the background of BayesiaLab, while estimating these effects. As [57] explains "*The LM algorithm searches for a set of likelihood distributions, which, when applied on the Joint Probability Distribution (JPD) encoded by the Bayesian network, allows obtaining the posterior probability distributions defined (as constraints) by the user*." In our case, suppose we want to estimate the causal effect of the feature "Cores" on "Preference Score", we would want to make sure that there are no confounding effects from the 9 other features that we have listed as potential confounders in assumption 3 (and we have already assumed that there are no other variables that affects this domain in assumption 2). In other words, we need to adjust for the effect of these 9 features, before we can estimate the actual (causal) effect of "Cores" on "Preference Score". One way to do this is to fix the probability distributions of the 9 other features (Threads, Performance, Base Frequency, Turbo Frequency, Power, List Price, Performance/Price, Base Frequency/Price and Turbo Frequency/Price) in the JPD represented in Fig.3r, while we change the states (or levels) of the probability distribution of "Cores". This is made possible by the LM algorithm. The LM algorithm searches for prior probability distributions for all the 9 confounding features that when applied on the JPD represented in Fig.3r, matches their posterior distributions, upon changing the levels of "Cores". Thus, the probability distributions of the 9 other features are maintained while we change the levels of "Cores"; which gives us the actual effect (or as BayesiaLab terms it -the "Direct Effect") of "Cores" on the "Preference

Score”. This is illustrated in Fig. 3s. In the figure, when we fix the interval (level) of cores to 0-0.157 (with a mean value of 0.025), its results in a distribution of “Preference Score” with a mean value of 0.129. Now when we change the level of “Cores” to 0.157-0.4 (with a mean value of 0.228), the distribution of “Preference Score” changes, as shown in Fig. 3t to a mean value of 0.149, while the distribution of the 9 confounding features remain unchanged.

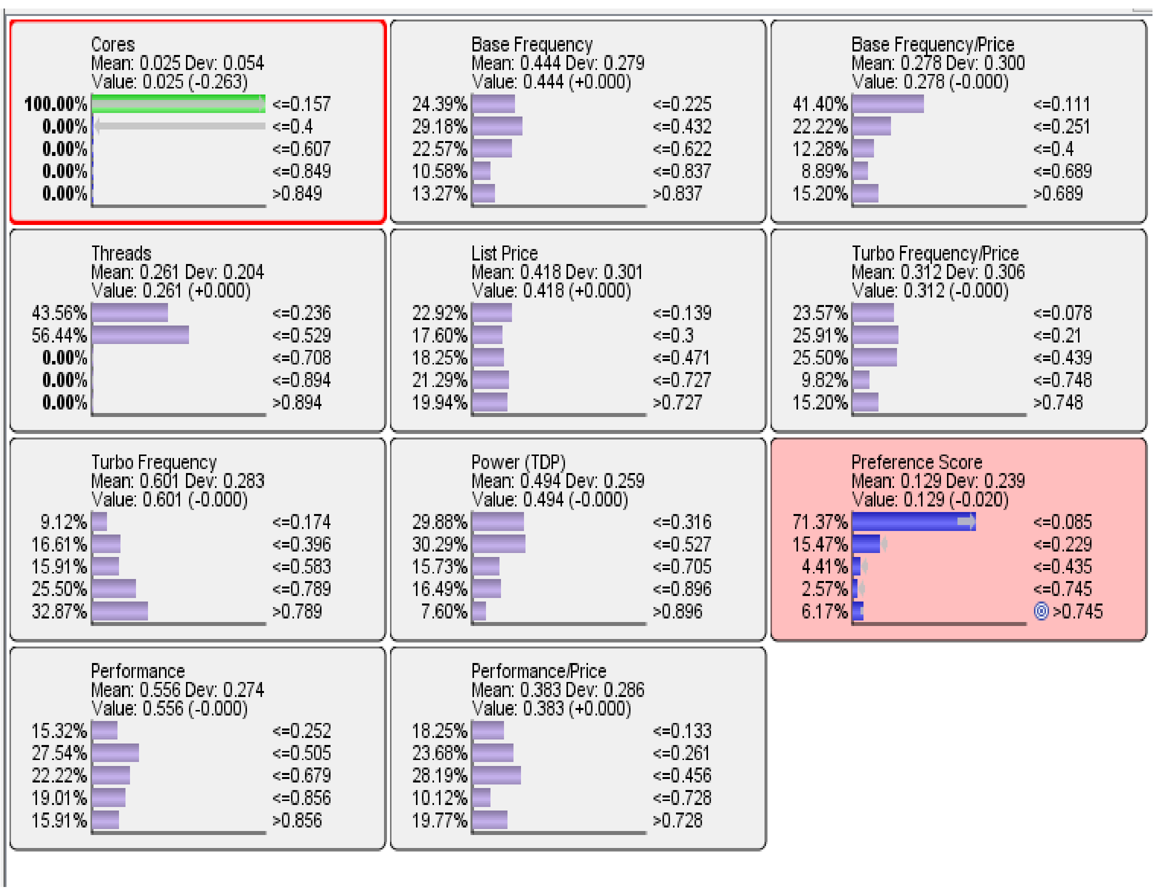


Fig. 3s LM algorithm illustration part 1

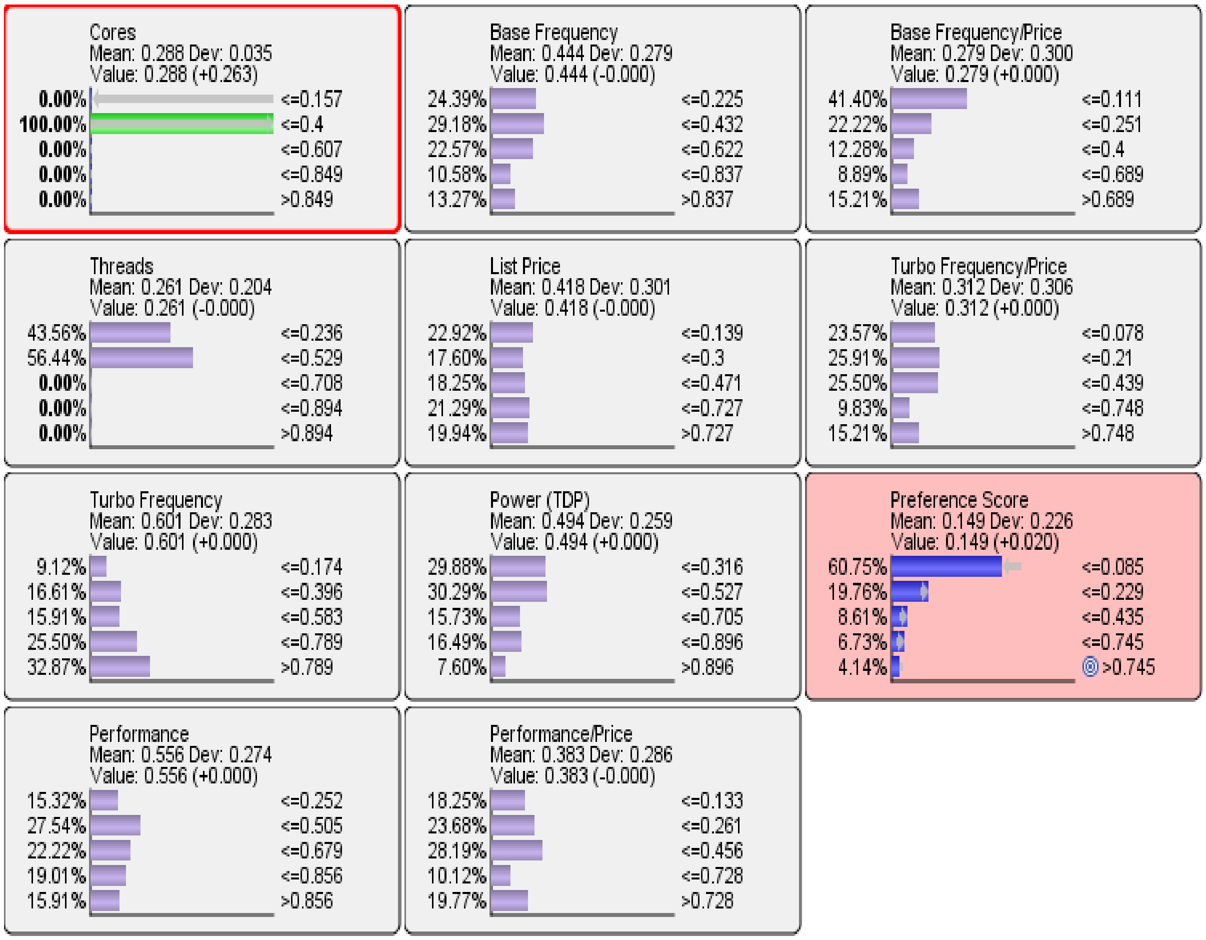


Fig. 3t LM algorithm illustration part 2

Fig. 3u and Fig. 3v contrasts this with the observed effect of "Cores" on "Preference Score" (i.e., without fixing the distributions of the 9 confounding features), where the "Preference Score" distribution changes from a mean value of 0.214 to 0.244 when the variable "Cores" changes from 0-0.157 (with a mean value of 0.025) to 0.157-0.4 (with a mean value of 0.228)

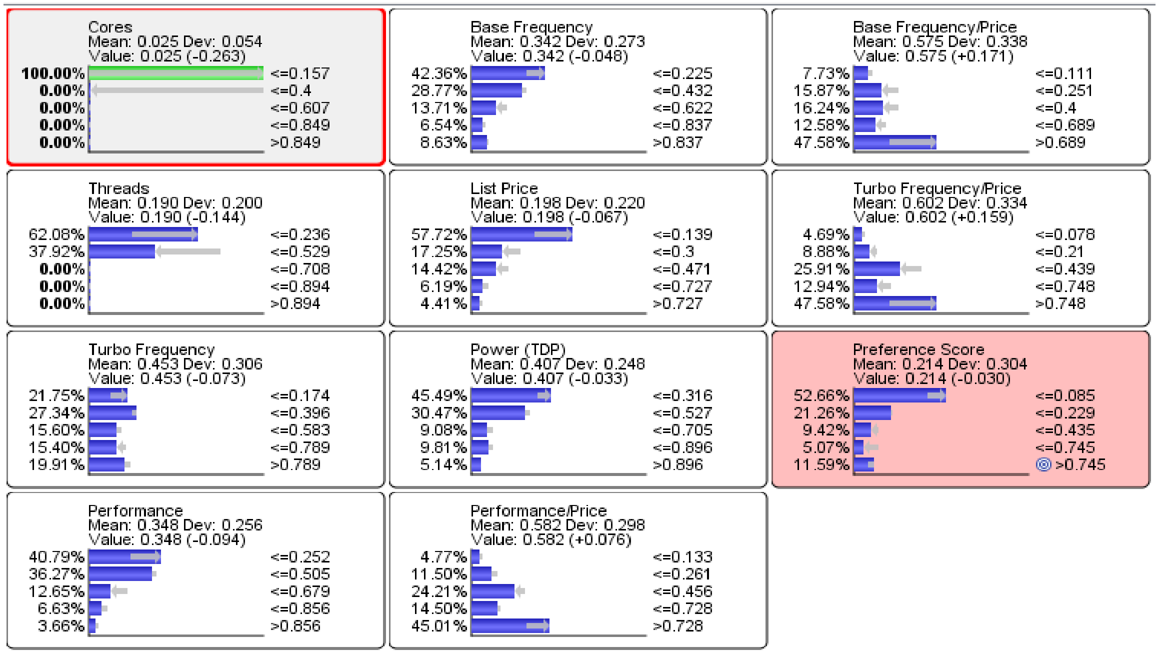


Fig. 3u LM algorithm illustration part 3

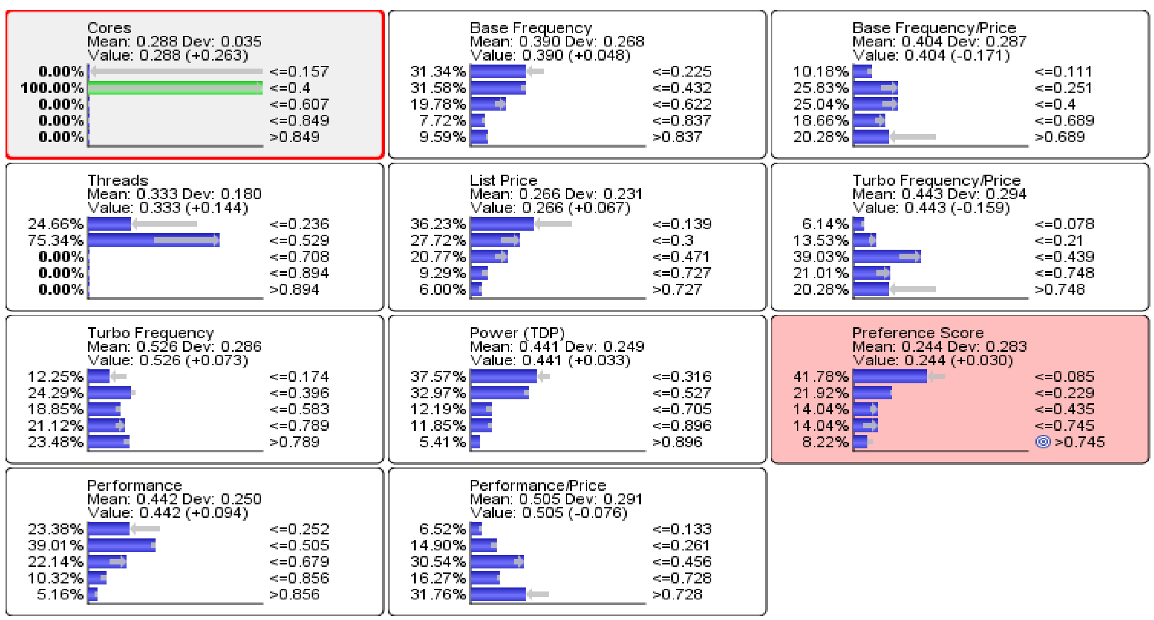


Fig. 3v LM algorithm illustration part 4

The direct effect visualization function in BayesiaLab plots the changes in mean value of the “Preference Score” distribution as we change the levels of “Cores”, while fixing the distributions of the 9 confounding features.

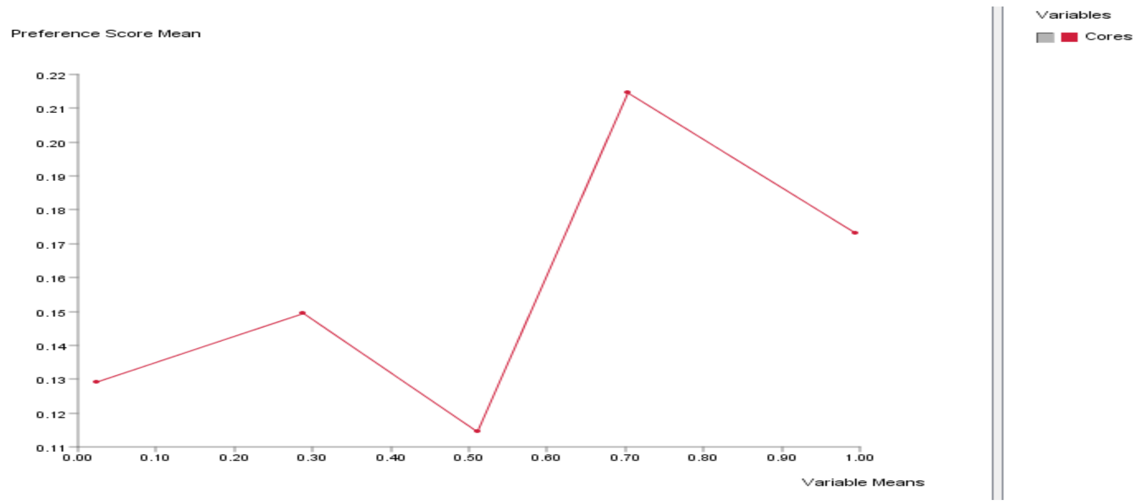


Fig. 3w Actual effect of “Cores” on “Preference Score” for the “Everything Cluster”

As we can observe in Fig. 3w, the feature “Cores” has a random, zig-zag effect on the “Preference Score” for the “Everything Cluster” customers. The peak value for the “Preference Score” mean (0.22) is reached corresponding to a value near the 3rd quartile (0.7) with respect to the “Cores” mean. Overall no strong, consistent pattern is apparent from the chart.

Similarly, the actual causal effects (or direct effects) of 9 other features on the “Preference Score” were computed, and their charts are plotted below (Fig. 3x– Fig. 3af):

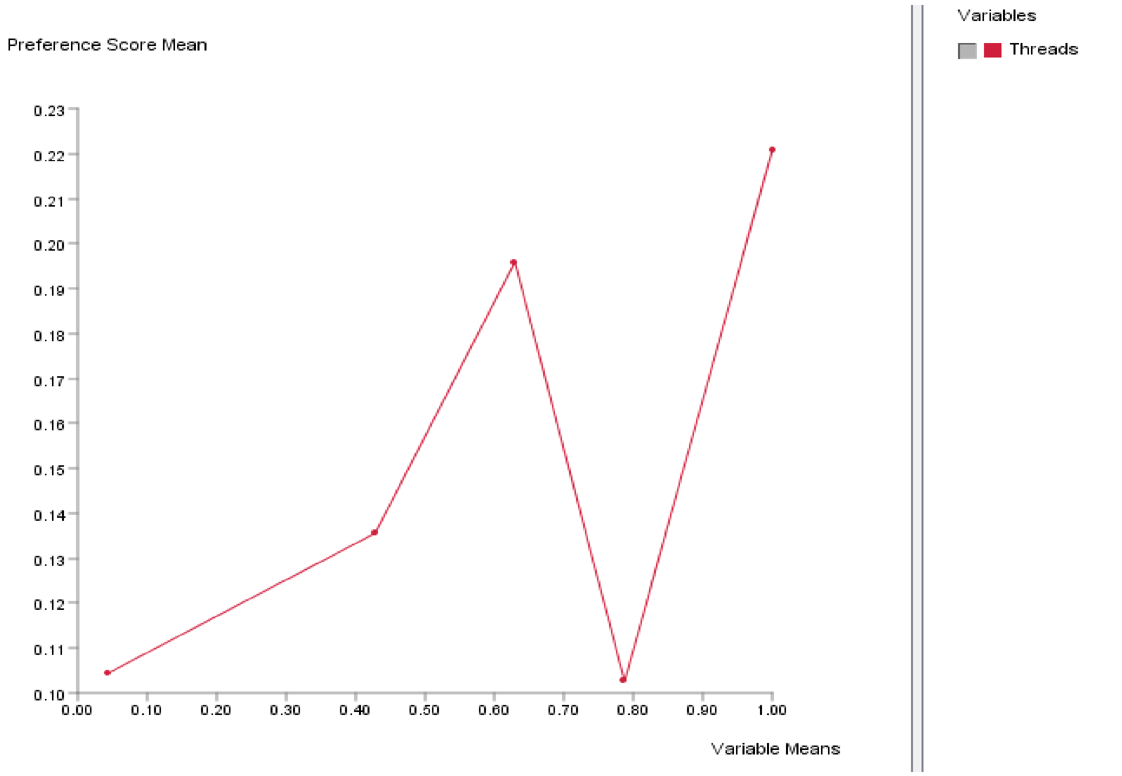


Fig. 3x Actual effect of "Threads" on "Preference Score" for the "Everything Cluster"

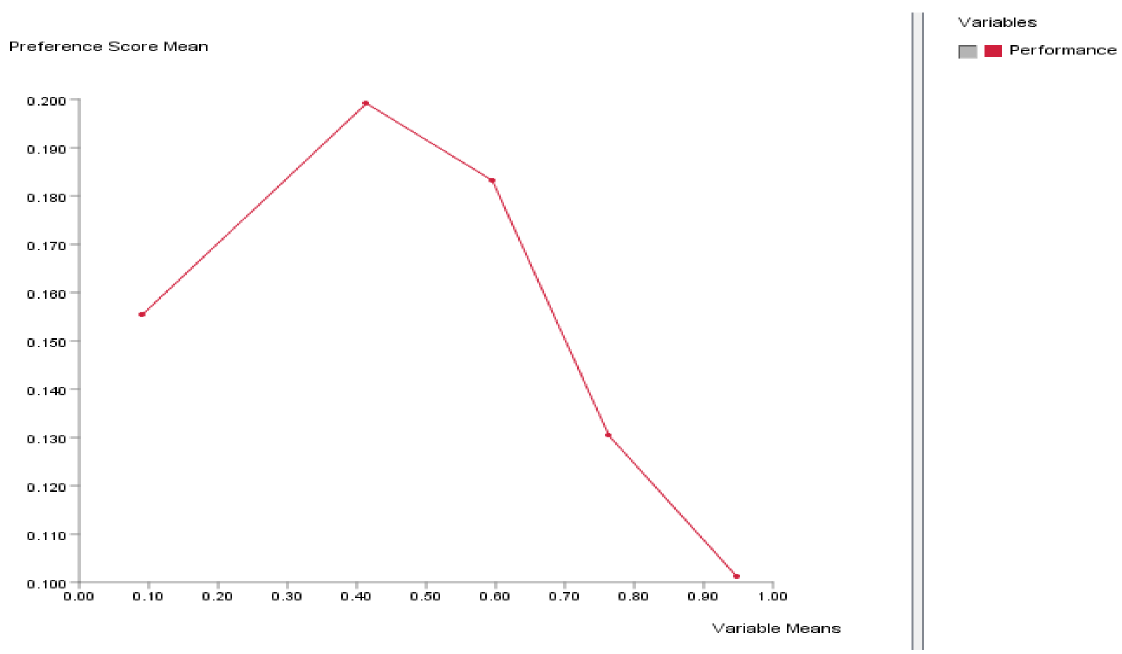


Fig. 3y Actual effect of "Performance" on "Preference Score" for the "Everything Cluster"

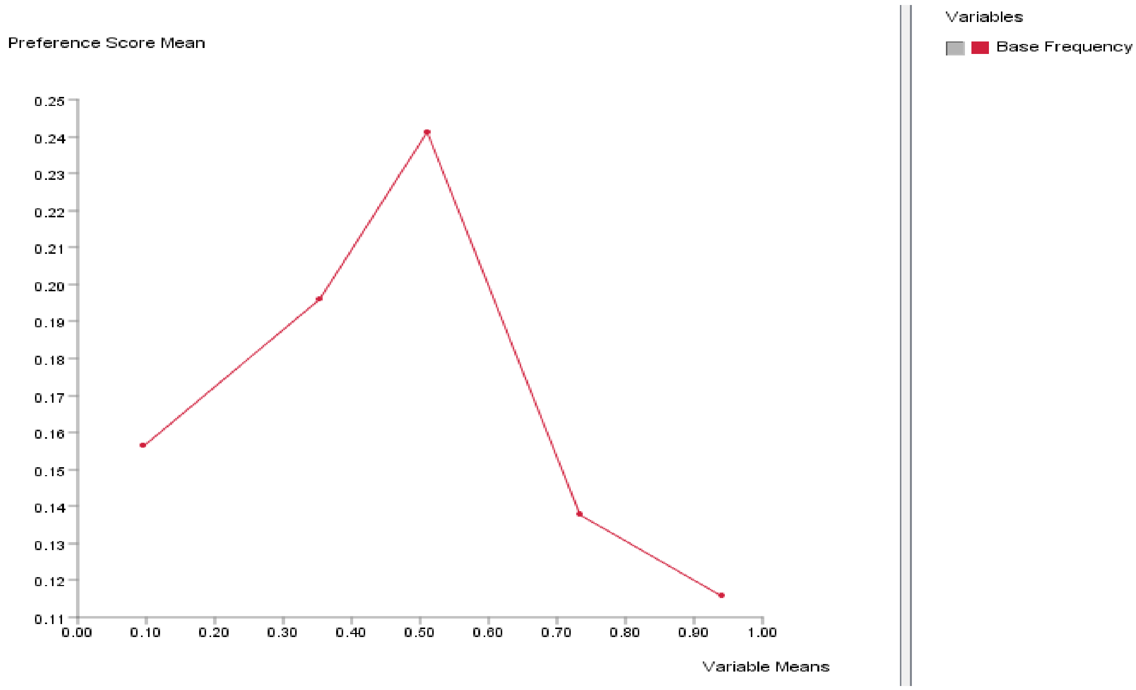


Fig. 3z Actual effect of “Base Frequency” on “Preference Score” for the “Everything Cluster”

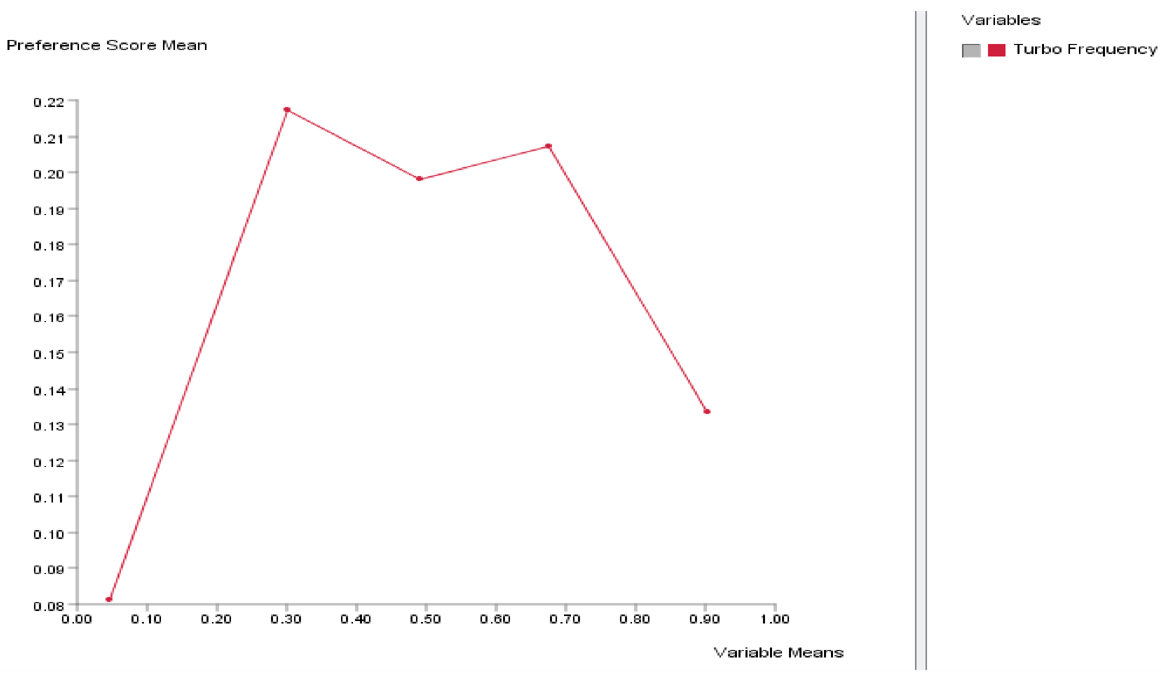


Fig. 3aa Actual effect of “Turbo Frequency” on “Preference Score” for the “Everything Cluster”

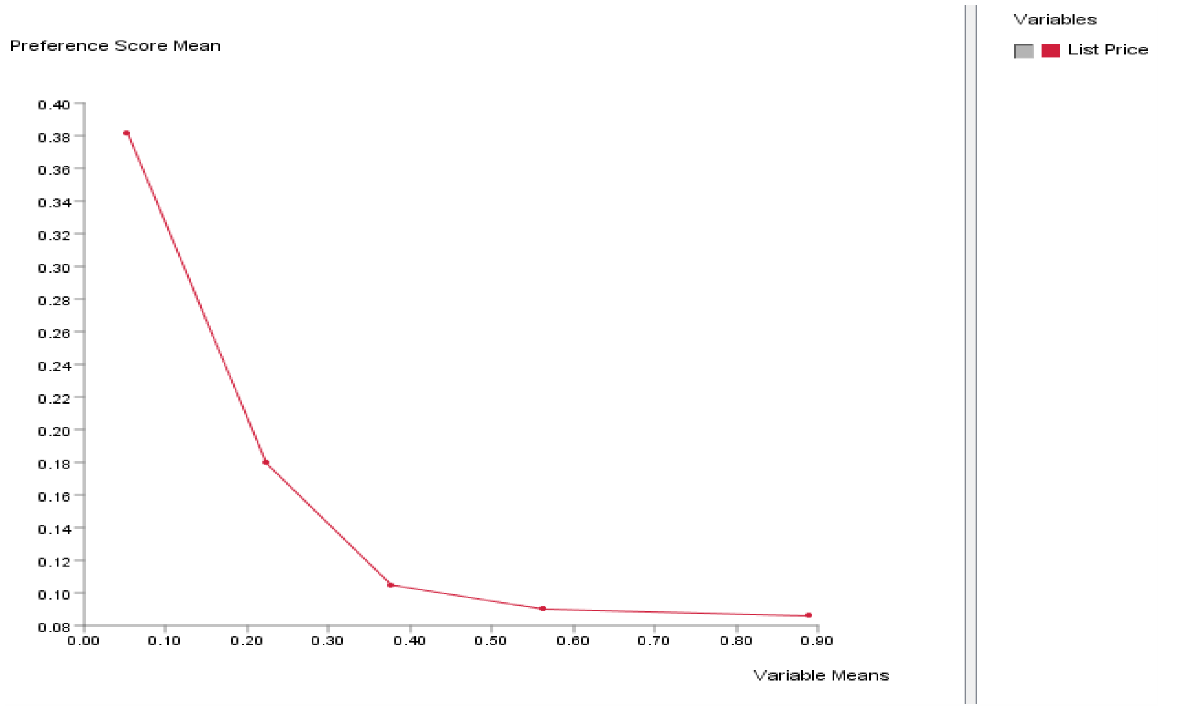


Fig. 3ab Actual effect of ”List Price” on “Preference Score” for the “Everything Cluster”

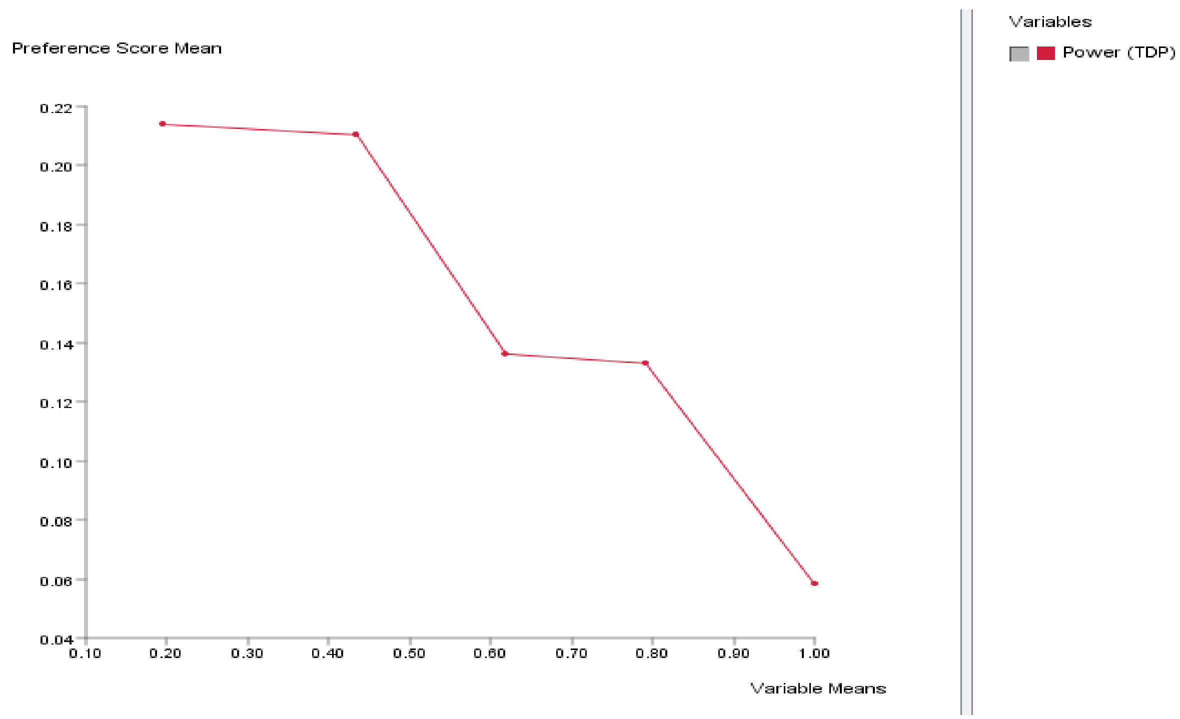


Fig. 3ac Actual effect of “Power” on “Preference Score” for the “Everything Cluster”

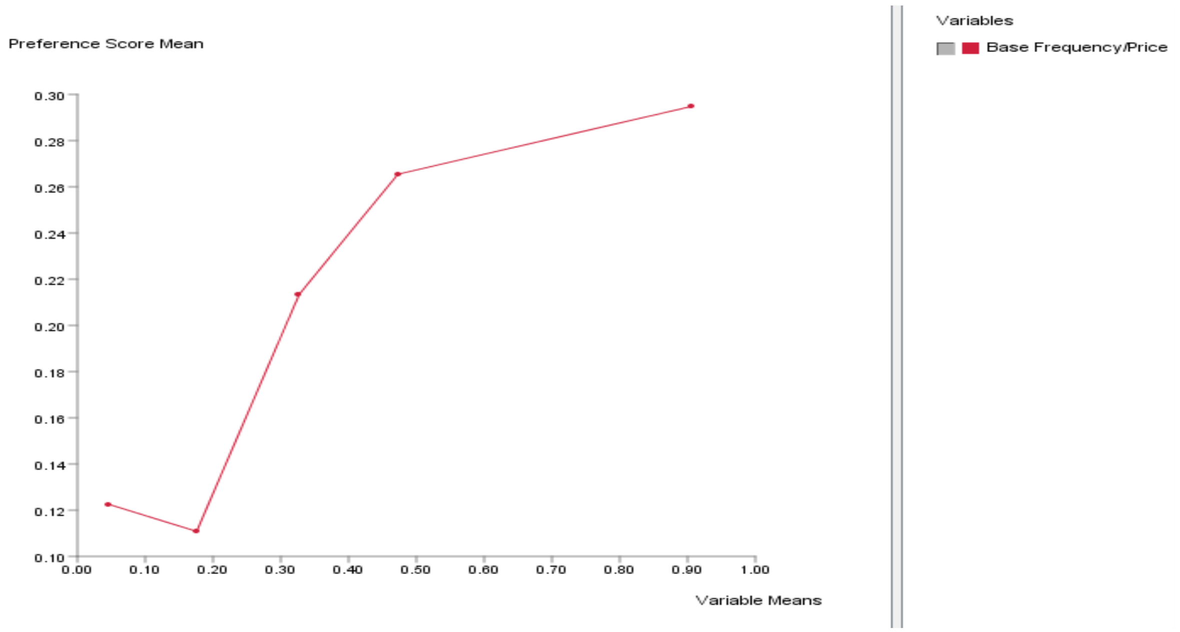


Fig. 3ad Actual effect of "Base Frequency/Price" on "Preference Score" for the "Everything Cluster"

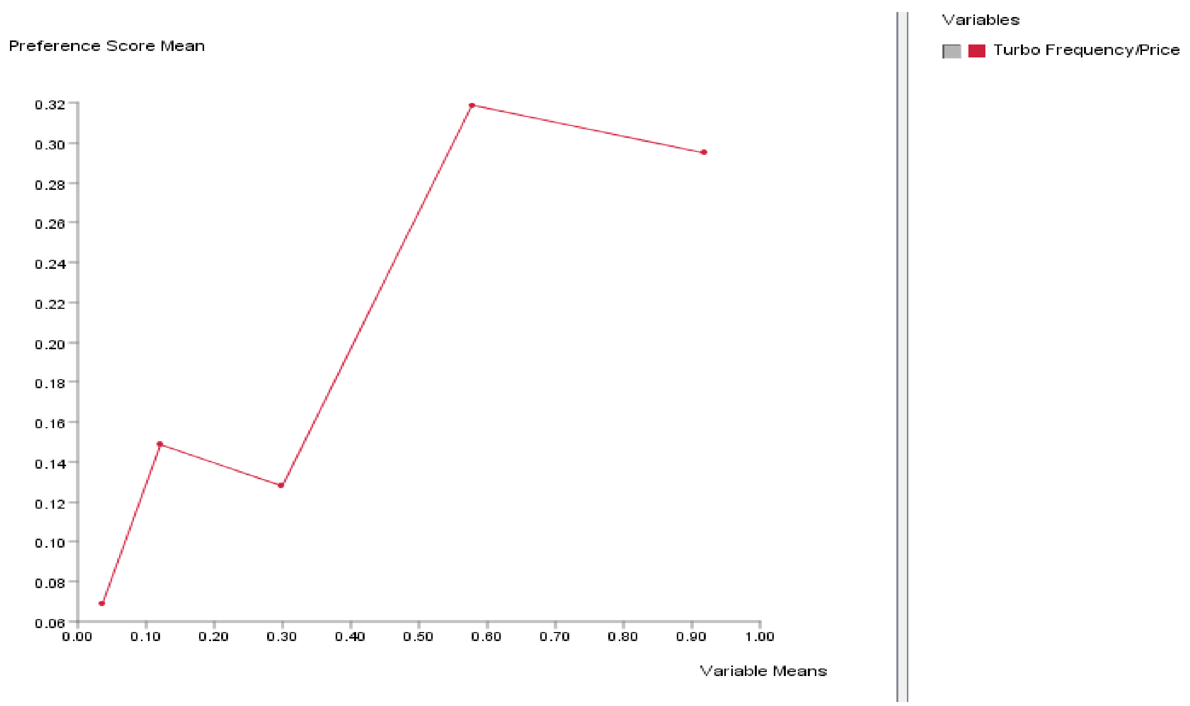


Fig. 3ae Actual effect of "Turbo Frequency/Price" on "Preference Score" for the "Everything Cluster"

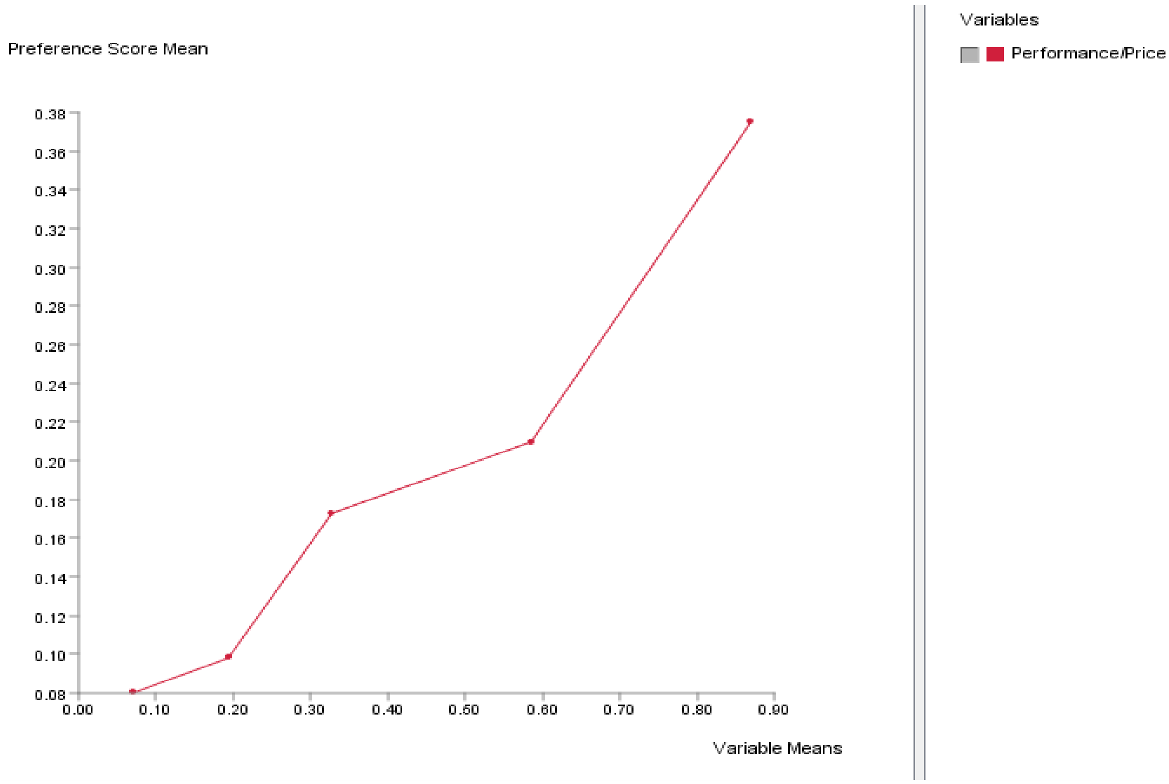


Fig. 3af Actual effect of "Performance/Price" on "Preference Score" for the "Everything Cluster"

Like "Cores", "Threads" have a zig-zag effect on "Preference Score", with no clear pattern apparent in Fig. 3x. "Performance" (Fig. 3y) and "Base Frequency" (Fig. 3z) exhibit an inverted "V" pattern reaching a maximum "Preference Score" mean of roughly 0.20 at 0.4 and 0.25 at 0.5 respectively. This suggests an existence of a sweet spot for both "Performance" and "Base Frequency" around midway of their respective feature values. However, the maximum mean "Preference Score" attained is relatively less (<0.25) for both these features. "Turbo-Frequency" exhibits an inverted "W" curve (Fig. 3aa) with 2 sweet spots, with a maximum mean "Preference Score" of 0.22 attained near the 1st quartile. "List Price" (Fig. 3ab) and "Power" (Fig. 3ac) exhibit a consistent declining pattern with respect to the “Preference Score” mean, suggesting that the “Everything Cluster” customers are price sensitive and, they prefer purchasing processors that consume less power. Perhaps the most insightful result from phase 2 concerning the “Everything Cluster” is the effect of “Performance/Price” on “Preference Score” (Fig. 3af). A strong, consistent increasing pattern in “Preference Score” mean, with an increasing “Performance/Price” mean suggests that the “Everything Cluster” customers prefer purchasing processors that offer good performance for the price they pay. Similarly, “Base Frequency/Price” (Fig. 3ad) and “Turbo Frequency/Price” (Fig. 3ae) exhibit a consistent increasing pattern with respect to the “Preference Score” (though not as strong as “Performance/Price”). Similar analysis can be performed for other customer clusters as well.

Chapter 4
CONCLUSION

Understanding customer preference is crucial for new product planning and marketing decisions. This thesis illustrated how data mining techniques can be used for understanding customer preference from historical data, instead of relying entirely on expert judgment or expensive marketing research experiments. However, mining customer preference from historical data is not a trivial task and requires careful validation by domain expertise as discussed in chapters 2 and 3. Using Intel Corporation as an example, this thesis presented a decision support framework that provides a holistic view on customer preference by following a 2-phase procedure. Phase 1 uses cluster analysis to create product profiles in step 1 and further creates customer profiles based on their preference levels for these product profiles in step 2. Phase 2 then delves deep into each of the customer profiles that are created in phase 1 and investigates causality behind their preference using Bayesian networks.

The customer sales data of processors from Intel server market's DP segment was used as an input to the 2-phase procedure. The data consists of 117 processors that span over four generations: Westmere, Sandy Bridge, Ivy Bridge and Haswell. From phase1, five processor clusters (or processor types) namely, Top-Bin, Tier 2-Bin, Value-Bin, Lowend-Bin, Low Core High Power Bin) were identified based on processor features in step 1. The five processor clusters were then validated using the processor features from the Broadwell generation (a newer generation of processors launched in Q1 2016 after Haswell). In step

2, seven customer clusters: Top-Bin Cluster, Tier2-Bin Cluster, Value-Cluster, Top-Bin and Tier2 Cluster, Tier2 and Value Cluster, Lowend Cluster, and, Everything Cluster were identified based on their preference levels for these processor types.

Phase 2 then determines the casual effects of the individual processor features on preference of the seven customer clusters and, identifies the preferred features that drove sales within each cluster. Phase 2 is elaborated using the example of the “Everything Cluster” in section 3.3.1.

As illustrated in section 3.2.3, the potential uses of phase 1 include:

1) Forecasting the demand mix of the new generation processors, i.e., how much an existing customer is going to purchase from each of the five processor types in a new generation.
2) When a new generation of processors is set to be launched, making better product recommendations for existing customers and new customers who are similar to the existing customers

Phase 2 enables a deeper understanding of customer preference at a product feature level and can be a useful input for product design and development of future generation processors.

Moving forward, future work includes:

1) Validation of phase-1 demand mix forecasts based on sales data of new generation processors from Broadwell.

2) Research on how the input from phase 2 can be used efficiently for product design and development decisions of new generation processors.

REFERENCES

[1] M. Singh, “New Product Forecasting.”

[2] R. G. Cooper and S. J. Edgett, “Benchmarking best practices performance results and the role of senior management,” *Prod. Dev. Inst.*, vol. 1, no. 2, p. 7, 2003.

[3] “Intel,” *Wikipedia*. 22-Jun-2017.

[4] K. G. Kempf, F. Erhun, E. F. Hertzler, T. R. Rosenberg, and C. Peng, “Optimizing capital investment decisions at Intel Corporation,” *Interfaces*, vol. 43, no. 1, pp. 62–78, 2013.

[5] “Where Can I Find Information about Intel Customers?,” *Intel*. [Online]. Available: http://www.intel.com/content/www/uk/en/support/programs/000015044.html. [Accessed: 19-Jun-2017].

[6] Moore, G. E., Cramming more components onto integrated circuits. *Electronics Magazine* (April 19, 1965), 114–117.

[7] “Tick-tock model,” *Wikipedia*. 15-Jun-2017.

[8] “Revealed Preference Data - Survey Analysis.” [Online]. Available: http://surveyanalysis.org/wiki/Revealed_Preference_Data. [Accessed: 28-Jun-2017].

[9] “Stated Preference Data - Survey Analysis.” [Online]. Available: http://surveyanalysis.org/wiki/Stated_Preference_Data. [Accessed: 28-Jun-2017].

[10] “Choice modelling,” *Wikipedia*. 05-Feb-2017.

[11] M. Szenberg, L. Ramrattan, and A. A. Gottesman, *Samuelsonian economics and the twenty-first century*. Oxford University Press, 2006.

[12] H. Q. Chen, T. Honda, and M. C. Yang, “Approaches for Identifying Consumer Preferences for the Design of Technology Products: A Case Study of Residential Solar Panels,” *J. Mech. Des.*, vol. 135, no. 6, pp. 061007-061007-12, May 2013.

[13] O. Netzer and V. Srinivasan, “Adaptive Self-Explication of Multiattribute Preferences,” *J. Mark. Res.*, vol. 48, no. 1, pp. 140–156, Feb. 2011.

[14] “Conjoint Analysis vs Self-Explicated Method: A Comparison,” *TRC Market Research*. [Online]. Available: http://www.trchome.com/107-market-research-knowledge/white-paper-library-complete/209-conjoint-analysis-vs-self-explicated-method-complete. [Accessed: 03-Jul-2017].

[15] M. Easterby-Smith, “The design, analysis and interpretation of repertory grids,” *Int. J. Man-Mach. Stud.*, vol. 13, no. 1, pp. 3–24, Jul. 1980.

[16] F. B. Tan and M. G. Hunter, “The Repertory Grid Technique: A Method for the Study of Cognition in Information Systems,” *MIS Q.*, vol. 26, no. 1, pp. 39–57, 2002.

[17] S. Cohen and others, “Maximum difference scaling: improved measures of importance and preference for segmentation,” in *Sawtooth Software Conference Proceedings, Sawtooth Software, Inc*, 2003, vol. 530, pp. 61–74.

[18] P. E. Green, J. D. Carroll, and S. M. Goldberg, “A General Approach to Product Design Optimization Via Conjoint Analysis,” *J. Mark.*, vol. 45, no. 3, pp. 17–37, 1981.

[19] P. E. Green and V. Srinivasan, “Conjoint Analysis in Marketing: New Developments with Implications for Research and Practice,” *J. Mark.*, vol. 54, no. 4, pp. 3–19, 1990.

[20] E. F. MacDonald, R. Gonzalez, and P. Y. Papalambros, “Preference Inconsistency in Multidisciplinary Design Decision Making,” *J. Mech. Des.*, vol. 131, no. 3, pp. 031009-031009-13, Mar. 2009.

[21] D. Horsky, P. Nelson, and S. S. Posavac, “Stating Preference for the Ethereal but Choosing the Concrete: How the Tangibility of Attributes Affects Attribute Weighting in Value Elicitation and Choice,” *J. Consum. Psychol.*, vol. 14, no. 1–2, pp. 132–140, 2004.

[22] R. G. Cummings, D. S. Brookshire, R. C. Bishop, and K. J. Arrow, *Valuing environmental goods: an assessment of the contingent valuation method*. Rowman and Littlefield Pub Incorporated, 1986.

[23] D. Kahneman and J. L. Knetsch, "Valuing public goods: The purchase of moral satisfaction," *J. Environ. Econ. Manag.*, vol. 22, no. 1, pp. 57–70, Jan. 1992.

[24] J. Beshears, J. J. Choi, D. Laibson, and B. C. Madrian, "How are preferences revealed?," *J. Public Econ.*, vol. 92, no. 8–9, pp. 1787–1794, Aug. 2008.

[25] W. Adamowicz, J. Louviere, and M. Williams, "Combining Revealed and Stated Preference Methods for Valuing Environmental Amenities," *J. Environ. Econ. Manag.*, vol. 26, no. 3, pp. 271–292, May 1994.

[26] "Cluster analysis," *Wikipedia*. 09-Jul-2017.

[27] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, 1st ed. 2013, Corr. 7th printing 2017 edition. New York: Springer, 2013.

[28] E. Mooi and M. Sarstedt, "Cluster Analysis," in *A Concise Guide to Market Research*, Springer Berlin Heidelberg, 2010, pp. 237–284.

[29] "Distance and Similarity Measures—Wolfram Language Documentation." [Online]. Available: http://reference.wolfram.com/language/guide/DistanceAndSimilarityMeasures.html.en. [Accessed: 16-Jul-2017].

[30] V. Estivill-Castro, "Why So Many Clustering Algorithms: A Position Paper," *SIGKDD Explor Newsl*, vol. 4, no. 1, pp. 65–75, Jun. 2002.

[31] G. Punj and D. W. Stewart, "Cluster Analysis in Marketing Research: Review and Suggestions for Application," *J. Mark. Res.*, vol. 20, no. 2, pp. 134–148, 1983.

[32] J. Pearl, *Causality*. Cambridge University Press, 2009.

[33] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[34] S. Conrady and L. Jouffe, *Bayesian Networks and BayesiaLab: A Practical Introduction for Researchers*. Bayesia USA, 2015.

[35] J. Pearl, M. Glymour, and N. P. Jewell, *Causal Inference in Statistics: A Primer*. John Wiley and Sons, 2016.

[36] S. Conrady and L. Jouffe, “Introduction to Bayesian Networks and BayesiaLab,” *September*, vol. 3, no. 201, p. 3, 2013.

[37] S. Conrady and L. Jouffe, “Chapter 11 Causality and Optimization (Draft),” 2016.

[38] I. Shpitser, T. VanderWeele, and J. M. Robins, “On the Validity of Covariate Adjustment for Estimating Causal Effects,” *ArXiv12033515 Cs Stat*, Mar. 2012.

[39] T. J. VanderWeele and I. Shpitser, “A new criterion for confounder selection,” *Biometrics*, vol. 67, no. 4, pp. 1406–1413, Dec. 2011.

[40] B. USA, “Bayesian Network Literature.” [Online]. Available: http://www.bayesia.com/bayesian-network-literature. [Accessed: 04-Jul-2017].

[41] F. L. Seixas, B. Zadrozny, J. Laks, A. Conci, and D. C. Muchaluat Saade, “A Bayesian network decision model for supporting the diagnosis of dementia, Alzheimer′s disease and mild cognitive impairment,” *Comput. Biol. Med.*, vol. 51, pp. 140–158, Aug. 2014.

[42] “Modeling and predicting the occurrence of brain metastasis from lung cancer by Bayesian network: A case study of Taiwan.” [Online]. Available: http://www.sciencedirect.com.ezproxy1.lib.asu.edu/science/article/pii/S001048251400033X. [Accessed: 04-Jul-2017].

[43] S. Jagdale, M. H. Kolekara, and U. P. Khot, “Smart Sensing Using Bayesian Network for Computer Aided Diagnostic Systems,” *Procedia Comput. Sci.*, vol. 45, pp. 762–769, 2015.

[44] X. Wu, H. Liu, L. Zhang, M. J. Skibniewski, Q. Deng, and J. Teng, “A dynamic Bayesian network based approach to safety decision support in tunnel construction,” *Reliab. Eng. Syst. Saf.*, vol. 134, pp. 157–168, Feb. 2015.

[45] L. Mkrtchyan, L. Podofillini, and V. N. Dang, “Bayesian belief networks for human reliability analysis: A review of applications and gaps,” *Reliab. Eng. Syst. Saf.*, vol. 139, pp. 1–16, Jul. 2015.

[46] P. C. Matthews, “A Bayesian support tool for morphological design,” *Adv. Eng. Inform.*, vol. 22, no. 2, pp. 236–253, Apr. 2008.

[47] L. Deligiannidis and H. R. Arabnia, *Emerging Trends in Image Processing, Computer Vision and Pattern Recognition*. Morgan Kaufmann, 2014.

[48] H. Cheng and R. Wang, “Semantic modeling of natural scenes based on contextual Bayesian networks,” *Pattern Recognit.*, vol. 43, no. 12, pp. 4042–4054, Dec. 2010.

[49] S. MacLean and G. Labahn, “A Bayesian model for recognizing handwritten mathematical expressions,” *Pattern Recognit.*, vol. 48, no. 8, pp. 2433–2445, Aug. 2015.

[50] E. W. FORGY, “Cluster analysis of multivariate data : efficiency vs. interpretability of classifications,” *Biometrics*, vol. 21, pp. 768–769, 1965.

[51] “Evaluation of clustering.” [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html. [Accessed: 30-Jul-2017].

[52] “Elbow method (clustering),” *Wikipedia*. 29-May-2016.

[53] R. Thorndike, “Who belongs in the family?,” *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.

[54] K. P. F.R.S, “LIII. On lines and planes of closest fit to systems of points in space,” *Philos. Mag.*, vol. 2, no. 11, pp. 559–572, Nov. 1901.

[55] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” *J. Educ. Psychol.*, vol. 24, no. 6, pp. 417–441, Sep. 1933.

[56] “cl_predict function | R Documentation.” [Online]. Available: https://www.rdocumentation.org/packages/clue/versions/0.3-54/topics/cl_predict. [Accessed: 16-Sep-2017].

[57] L. Jouffe and S. Conrady, “Causal Inference with Bayesian Networks.” 29-Aug-2013.

APPENDIX A

CUSTOMER PROFILES

1. Customer profile- Value Bin Cluster

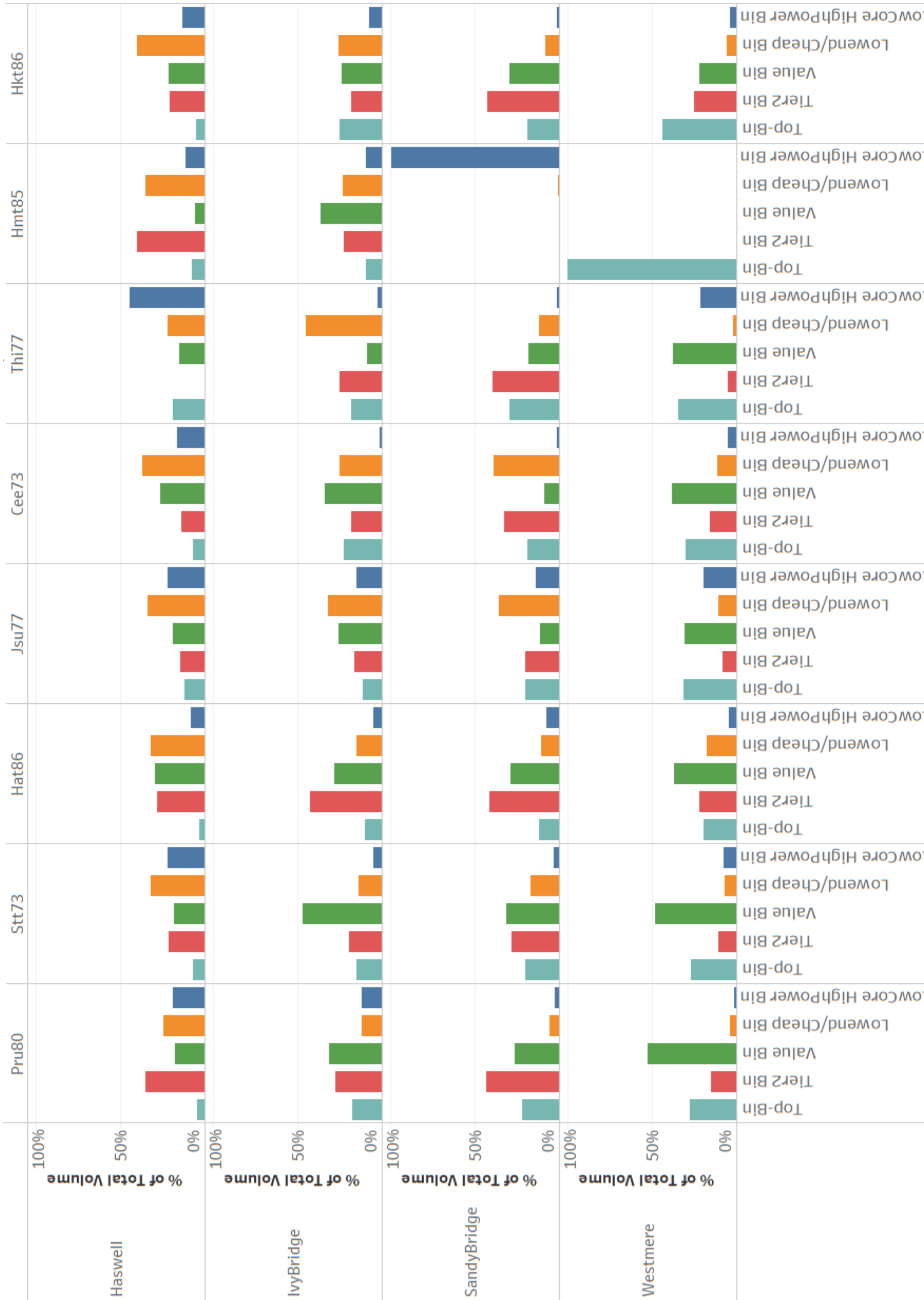
2. Customer profile- Lowend Cluster

3. Customer profile- Top Bin and Tier 2 Cluster

4. Customer profile- Value and Tier 2 Cluster


Nno79
Ett73
Utq83
Elr80
Eni77
Eir76
Haswell
IvyBridge
SandyBridge
Westmere
% of Total Volume
80%
60%
40%
20%
0%
Top-Bin
Tier2 Bin
Value Bin
Lowend/Cheap Bin
LowCore HighPower Bin

5. Customer profile- Everything Cluster


Hkt86
Hmt85
Thi77
Cee73
Jsu77
Hat86
Stt73
Pru80
LowCore HighPower Bin
Lowend/Cheap Bin
Value Bin
Tier2 Bin
Top-Bin
Haswell
IvyBridge
SandyBridge
Westmere
% of Total Volume
100%
50%
0%

Uus74
Nte81
Eec74
Aar78
Gsm76
See87
Pri75
Lcc73
Haswell
IvyBridge
SandyBridge
Westmere
% of Total Volume
100%
50%
0%
Top-Bin
Tier2 Bin
Value Bin
Lowend/Cheap Bin
LowCore HighPower Bin