Predicting and Interpreting Students Performance using Supervised Learning and Shapley Additive Explanations

by

Wenbo Tian

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2018 by the
Graduate Supervisory Committee:

Ihan Hsiao, Chair
Rida Bazzi
Hasan Davulcu

ARIZONA STATE UNIVERSITY

May 2019

ABSTRACT

Due to large data resources generated by online educational applications, Educational Data Mining (EDM) has improved learning effects in different ways: Students Visualization, Recommendations for students, Students Modeling, Grouping Students, etc. A lot of programming assignments have the features like automating submissions, examining the test cases to verify the correctness, but limited studies compared different statistical techniques with latest frameworks, and interpreted models in an unified approach.

In this thesis, several data mining algorithms have been applied to analyze students' code assignment submission data from a real classroom study. The goal of this work is to explore and predict students' performances. Multiple machine learning models and the model accuracy were evaluated based on the Shapley Additive Explanation.

The Cross-Validation shows the Gradient Boosting Decision Tree has the best precision 85.93% with average 82.90%. Features like Component grade, Due Date, Submission Times have higher impact than others. Baseline model received lower precision due to lack of non-linear fitting.

DEDICATION

To my parents, professors and friends

ACKNOWLEDGMENTS

I would like to thank my thesis advisor Prof. Ihan Hsiao firstly. I couldn’t finish this thesis and any research work without her guidance and patience. During my master study, she supervised me regularly, and always gave me valuable suggestions. I am also very thankful to Prof. Rida Bazzi and Prof. Yezhou Yang. Without the permission to access the dataset, we could not get any analysis result. I really thank my parents for always supporting me in my whole life.

Lastly, I want to say thank you to every professor, every team member in our research lab, and my friends for supporting me.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Data mining, also called Knowledge Discovery in Databases (KDD), is the field of discovering novel and potentially useful information from large amounts of data (1999). It has been proposed that educational data mining methods are often different from standard data mining methods, due to the need to explicitly account for (and the opportunities to exploit) the multi-level hierarchy and non-independence in educational data [Baker in press]. For this reason, it is increasingly common to see the use of models drawn from the psychometrics literature in educational data mining publications [1].

Educational data mining technology comprehensively applies the theories and techniques of education, computer science, psychology and statistics to solve problems in educational research and teaching practice. By analyzing and mining education-related data, EDM technology can discover and Solve various problems in education, such as assisting managers in making decisions, helping teachers improve courses, and improving students' learning efficiency. The complexity of educational issues and the interdisciplinary nature of EDM in data sources, data characteristics, research Methods and application purposes show their uniqueness.

In the past few years, revolutionary changes have taken place in both the education and information fields. Online learning systems, smartphone applications and social networks have provided a large number of applications and data for EDM research. Take the online learning system MOODLE [3] as an example. As of 2013, it has served more than 60 million students and teachers worldwide [4]. As of June 2012, the number of global smartphone users exceeded 1 billion [5], and the number of social media Facebook users exceeded 2.2 billion. People [6]. Massive open online courses (MOOCs) are new teaching models that have emerged in the past two years. By the end of 2014, the number of users registered on the MOOCs website Coursera has exceeded 10 million [7] Obviously, EDM is also in the era of "big data". This special background indicates that EDM research will develop rapidly in recent years.

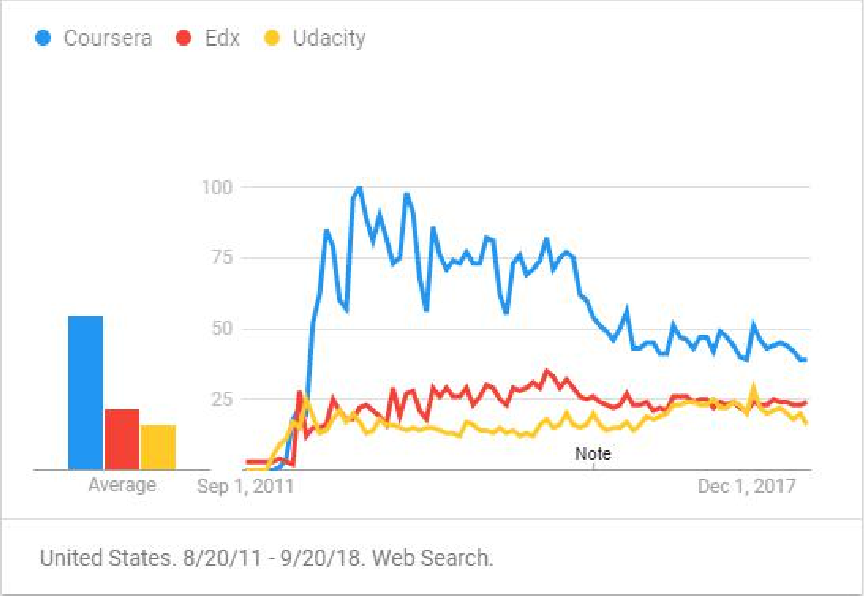


Figure 1.1: The E-Learning Trend

1.1 Motivation

Through techniques such as EDM and LA, it can help teachers effectively improve their teaching. For example, the teacher can check the time the students stayed on the same question, judge whether they have reviewed the course after answering the wrong question and count the number of questions they asked online and how much they participated in the discussion.

Using the data analysis results of EDM and LA, teachers can better understand students, observe the students' learning process to find the most appropriate teaching methods and teaching sequences, and adopt different teaching methods and teaching strategies for students with different personalities. So, with the educational dataset from one specific course, our goal is to build a statistical model using EDM which shows the student current grade and the room for improvement. This study also helps instructors to adjust course schedule in time.

1.2 Research Questions

This thesis addresses following research questions:

1) Which data mining algorithm is more suitable to predict students' performance by mining historical data?

2) How do we interpret the data mining model when it's not linear model?

1.3 Organization

Compared with previous EDM review papers, in this thesis, we firstly introduce the dataset from CSE340 course, and design the features based on every submission status. After that, we generate the predictor by using 3 different machine learning method and interpret the results by introducing SHAP value.

The thesis organization is as follow. Chapter 2 is about related work of EDM. Chapter 3 review the methodologies. Chapter4 involves the results analysis and evaluation. The last chapter is conclusion and future work.

Chapter 2

RELATED WORK

In this section, we review the related work from several aspects: (1) The principle of Educational Data Mining, (2) The recent work in Educational Data Mining.

2.1 The Principle of EDM

The most closely related disciplines with EDM are computer science, pedagogy, and statistics, the interaction between every two subjects has generated data mining and machine learning respectively (DM&ML), computer-based education (CBE), and learning analytics (LA). The characteristics of EDM can be seen by comparison with these three areas.

The main difference between EDM and general DM&ML research lies in the educational discipline characteristics of its data, which are reflected in the following aspects:

Multidisciplinary: EDM data usually involves concepts and techniques in pedagogy, psychology, and sociology, such as teaching purposes, learning experiences, teaching assessments, interests, motivations, teamwork, relationships, and family backgrounds. For this type of data, Researchers must be able to understand their concepts as well as the techniques for measuring and evaluating them.

Multi-level: The multi-level nature of EDM data comes from the structure of educational institutions and teaching materials. For example, students can be organized by school district, school, department and class, and the teaching content can be organized according to courses, chapters, knowledge points and concepts.

Multi-precision: EDM data usually contains time scales. A teaching study may span several years or even a lifetime, or it may be recorded with millisecond precision. This allows researchers to analyze data with different time precision.

Multiple scenarios: The multi-scenario characteristics of EDM data come from the characteristics of the education discipline itself. A student's experience in acquiring knowledge is related to the time, place, teacher and environment of the teaching, and also to the students' own motivations, abilities and emotions. Changes may lead to different learning experiences.

Multiple semantics: The multi-semantic nature of EDM data comes from several aspects, such as the ambiguity of the behavior of teachers and students, the ambiguity of natural language used by teachers and students, the noise data in the educational environment or the missing data. Even the interpretation of the same data by different educational theories can lead to ambiguity.

The main difference between EDM and general CBE research lies in the difference in application purpose. The latter aims to assist or replace the traditional teaching process, while EDM is dedicated to the realization of functions that are lacking or difficult to accomplish in traditional teaching.

The main difference between EDM and general LA research is the technology used: the latter mostly uses statistics, while EDM mostly uses machine learning and data mining techniques. From another perspective, LA focuses on describing events that have occurred or their results, and EDM focuses on discovering new knowledge and new models.

2.2 The recent work in Educational Data Mining

The normal workflow of EDM includes three stages of preprocessing, data mining and evaluation. From an educational point of view, this is a knowledge found in the data generated by the educational environment, and then used to improve the educational environment. Romero and Ventura [2007] categorize work in educational data mining into the following categories: Statistics and visualization, Web mining.

The normal workflow of EDM includes three stages of preprocessing, data mining and evaluation. From an educational point of view, this is a knowledge found in the data generated by the educational environment, and then used to improve the educational environment. Romero and Ventura (2007) categorize work in educational data mining into the following categories: Statistics and visualization, Web mining.

From the recent Educational Data Mining in Computer Science Education (CSEDM) Workshop, researchers Partho Mandal and I-Han Hsiao (2018) use differential mining [7] to explore students' problem-solving strategies. In this work, Students' problem-solving activities on multiple choice questions were collected from a semester-long computer science programming course in 2016 Fall semester. Based on each question's correctness, complexity, topic, and time, the frequent behavioral patterns were extracted to build the problem-solving sequences. Seven distinct learning behaviors were discovered based on these patterns between high and low performing students, which provided insight into students' meta-cognitive skills and thought processes.

Besides differential mining, researchers Mohammed Alzaid and I-Han Hsiao (2018) personalize self-assessing quizzes in programming courses [8]. This work presents an adaptive quizzing recommender for introductory programming courses. It enhanced the flow design of the question attempts to provide learners with the capability to evaluate the given set of questions and extends the to include a personalized recommended question. The implemented approach aims to enable the learners to build their programming confidence and steadily master the concepts. This work also aims to enhance the coverage of the dataset of questions. It will provide the learners with the ability to take control and enhance their learning outcome which may lead them to adopt a better learning strategy.

From the Predictive Modelling of Student Reviewing Behaviors in an Introductory

Programming Course [9], researchers Yancy Vance Paredes, David Azcona, I-Han Hsiao and Alan F. Smeaton (2018) developed predictive models based on students’ reviewing behaviors in an Introductory Programming course. These patterns were captured using an educational technology that students used to review their graded paper-based assessments. Models were trained and tested with the goal of identifying students’ academic performance and those who might need assistance. The results of the retrospective analysis show a reasonable accuracy. This suggests the possibility of developing interventions for students, such as providing feedback in the form of effective reviewing strategies.

In order to reduce the state space of programming problems [10], researchers Rui Zhi, Thomas Price, Nicholas Lytle, Yihuan Dong and Tiffany Barnes (2018) present a procedure for defining a small but meaningful programming state space based on the presence or absence of features of correct solution code. They present a procedure to create these features using a panel of human experts, as well as a data-driven method to derive them automatically. We compare the expert and data-driven features, the resulting state spaces, and how student progress through them. The results show that both approaches dramatically reduce the state-space compared to traditional code-states and that the data-driven approach has high overlap with the expert features.

Chapter 3

METHODOLOGY

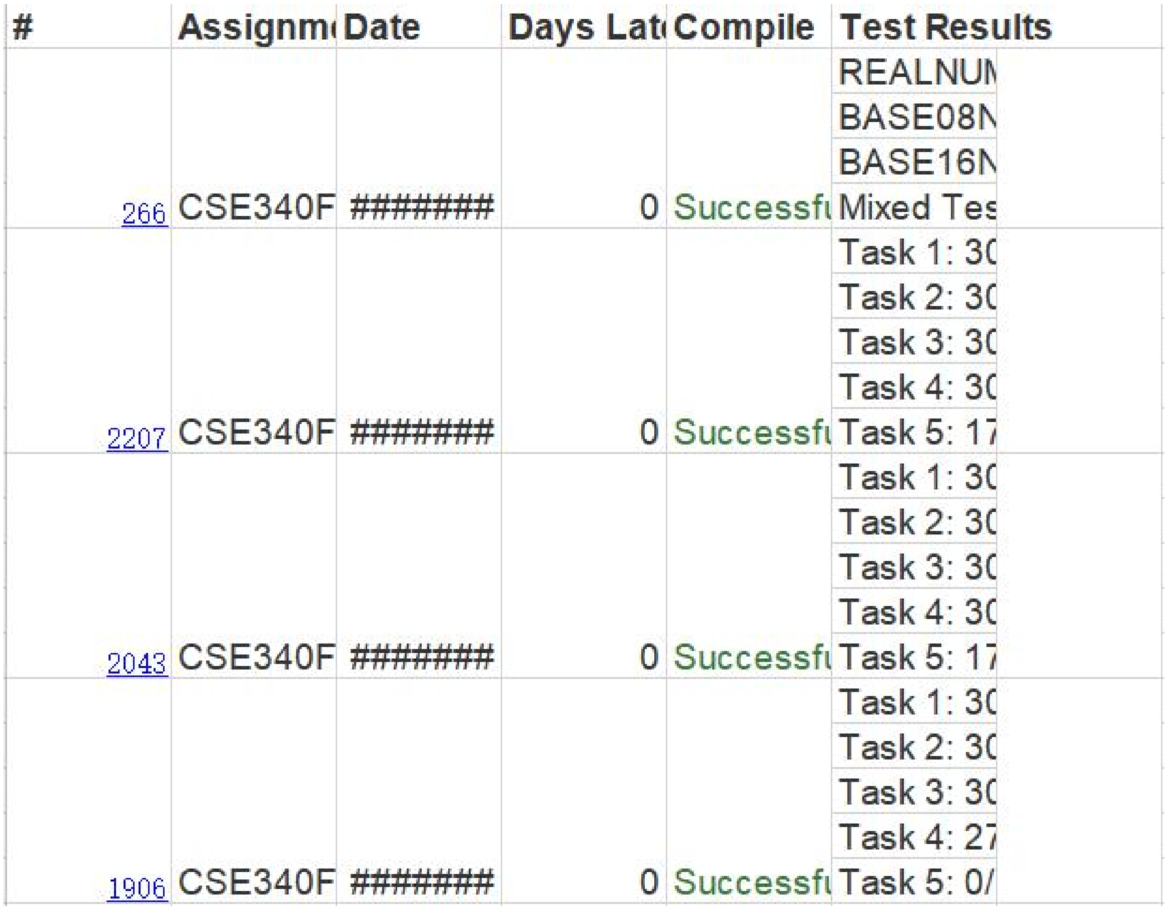
In this chapter, I will explain the methodology of training predictor to predict students' performance. I organize this chapter in two parts. First, I will explain the dataset we use, grading criteria and some fundamental statistics. After that, I will go through my data mining pipeline from dataset preprocessing, feature desing, feature normalize, training strategy to results analysis and model explanation.

3.1 Dataset and Grading criteria

The dataset we use in this thesis is from the course CSE340 Principles of Programming Languages at Arizona State University. The dataset was recorded in Spring 2017 and has 248 students' submissions based on time series. For each student, we get the real-world data of both successful submissions and failure submissions.

Figure 3.1 shows the screenshot of the dataset structure. For each sheet, the dataset records every submission status. Here we have attributes like Assignment number, Submission date, Delay days, Compile status, and Test Results. The Figure 3,2 - Figure 3.7 show the dataset statistics.

The dataset we use in this thesis is from the course CSE340 Principles of Programming Languages at Arizona State University. The dataset was recorded in Spring 2017 and has 248 students' submissions based on time series. For each student, we get the real-world data of both successful submissions and failure submissions.

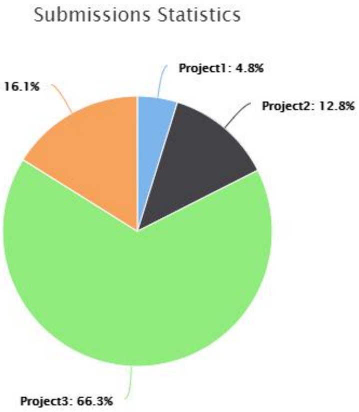


#	Assignm	Date	Days Lat	Compile	Test Results
266	CSE340F	#######	0	Successfı	REALNUN BASE08N BASE16N Mixed Tes
2207	CSE340F	#######	0	Successfı	Task 1: 3(Task 2: 3(Task 3: 3(Task 4: 3(Task 5: 17
2043	CSE340F	#######	0	Successfı	Task 1: 3(Task 2: 3(Task 3: 3(Task 4: 3(Task 5: 17
1906	CSE340F	#######	0	Successfı	Task 1: 3(Task 2: 3(Task 3: 3(Task 4: 27 Task 5: 0/

Figure 3.1: Original Dataset Structure

Each student has 4 coding projects during the whole semester. Each project has a specific compiler topic. The project1 requires students to extend lexical analyzer to support REALNUM, BASE08NUM, BASE16NUM. The project2 is to determine the number of grammar rules, useless symbols, and calculate FIRST sets, FOLLOW sets. The project3 is about parsing. The proejct4 is to describe statement semantics.

According to the course requirement, each project has 4 or 5 tasks to solve, and every task has different weight. The total grade is dependent on each task grade and the delay days. Each delayed submission will get penalty.



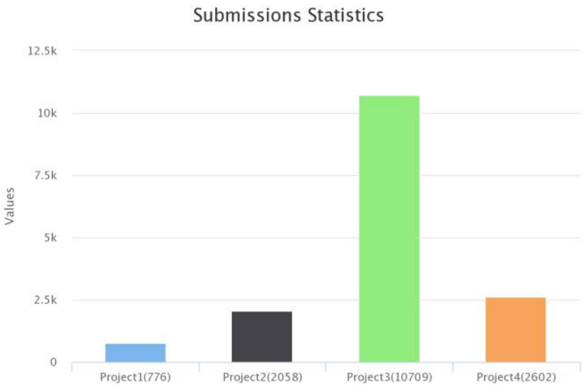
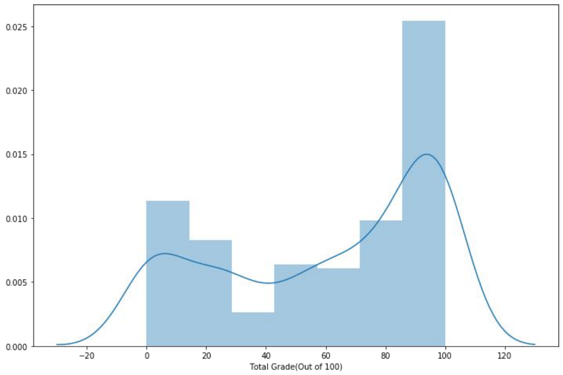


Figure 3.2: The submission statistics in Pie chart

Figure 3.3: The submission statistics in Bar chart



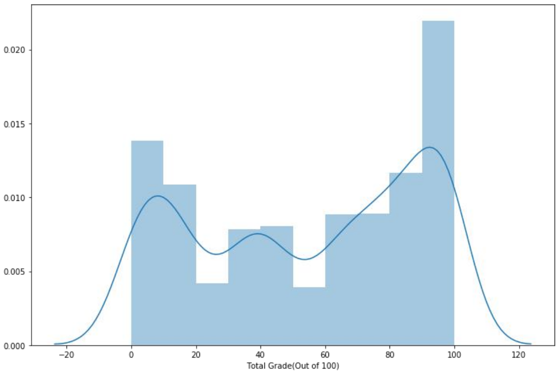
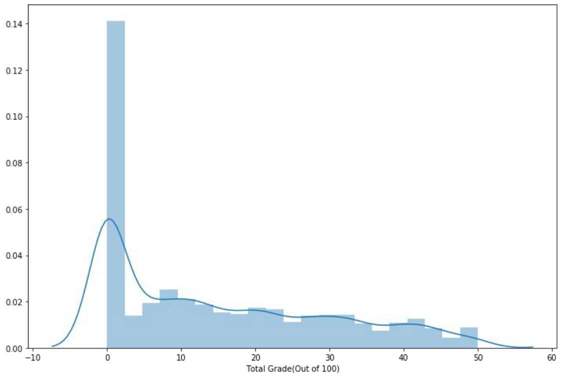


Figure 3.4: The submission grade distribution of project1

Figure 3.5: The submission grade distribution of project2



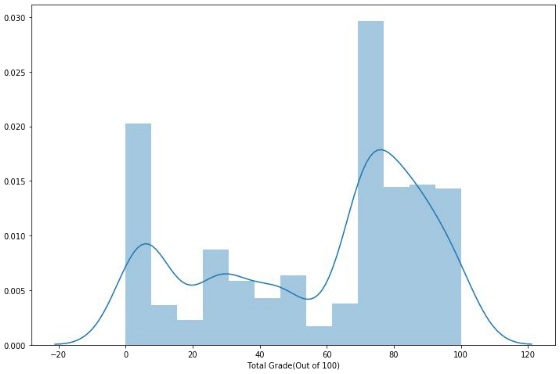


Figure 3.6: The submission grade distribution of project3

Figure 3.7: The submission grade distribution of project4

3.2 Feature Design

Feature Design is one of most important part in machine learning. It’s a process consisting of three sub-modules: feature construction, feature extraction, feature selection. Feature Construction is building new features from raw data requires identifying physical features. Feature Extraction is automatically constructing new features, transforming the original features into a set of features with significant physical or statistical significance or kernel. For example, time stamp, geometric features, textures, etc. Feature selection is selecting a set of most statistically significant feature subsets from the feature set and delete the irrelevant features to achieve the dimension reduction effect.

Figure 3.1 shows the screenshot of the dataset structure. During the data cleaning process, we removed several invalid data. If the student didn’t make any submission, or the student drop the class during the course. We would mark them as invalid data and remove in order to reduce data noise.

In the original dataset, based on the course syllabus, we can divide the timestamp feature into ‘Remaining time’, ‘Delay times’, and ‘Total Submissions’. After that, we also can calculate the day submission frequency and compiler failures based on current timestamp. In order to merge submissions with different number of grade part, we can add one binary bit to judge if the submission has 4 parts or 5 parts.

We only have 5 features initially. In order to make data better enough. We extend them and build new important features by feature engineering. By applying feature combination and feature correlation tactics, we get total 15 features in training dataset, and 18 features after one-hot encoding. Table 3.1 shows the features name and description after feature engineering.

Table 3.1: Feature Description

Features	Description
Remaining time	How much time is left to complete the task?
Compiler Failures	The submission failure times so far
Number of submissions	The total number of submissions
Grade1	The grade of part1
Grade2	The grade of part2
Grade3	The grade of part3
Grade4	The grade of part4
Grade5	The grade of part5
Total Grade	The total grade after deducting penalty
Has 4 parts	If the project has only 4 grading parts?
Delay times	The delayed days of the submission
Day frequency	How many submission times per day
Is Weekday?	If the submission happened on weekday
Project Number	The project number

According to the paper [14], The Pearson correlation coefficient is used to determine whether each feature is closely related, and if it is relevant, it is a repeating feature and can be removed. If every feature we enter into the machine learning model is unique, we can generate best result.

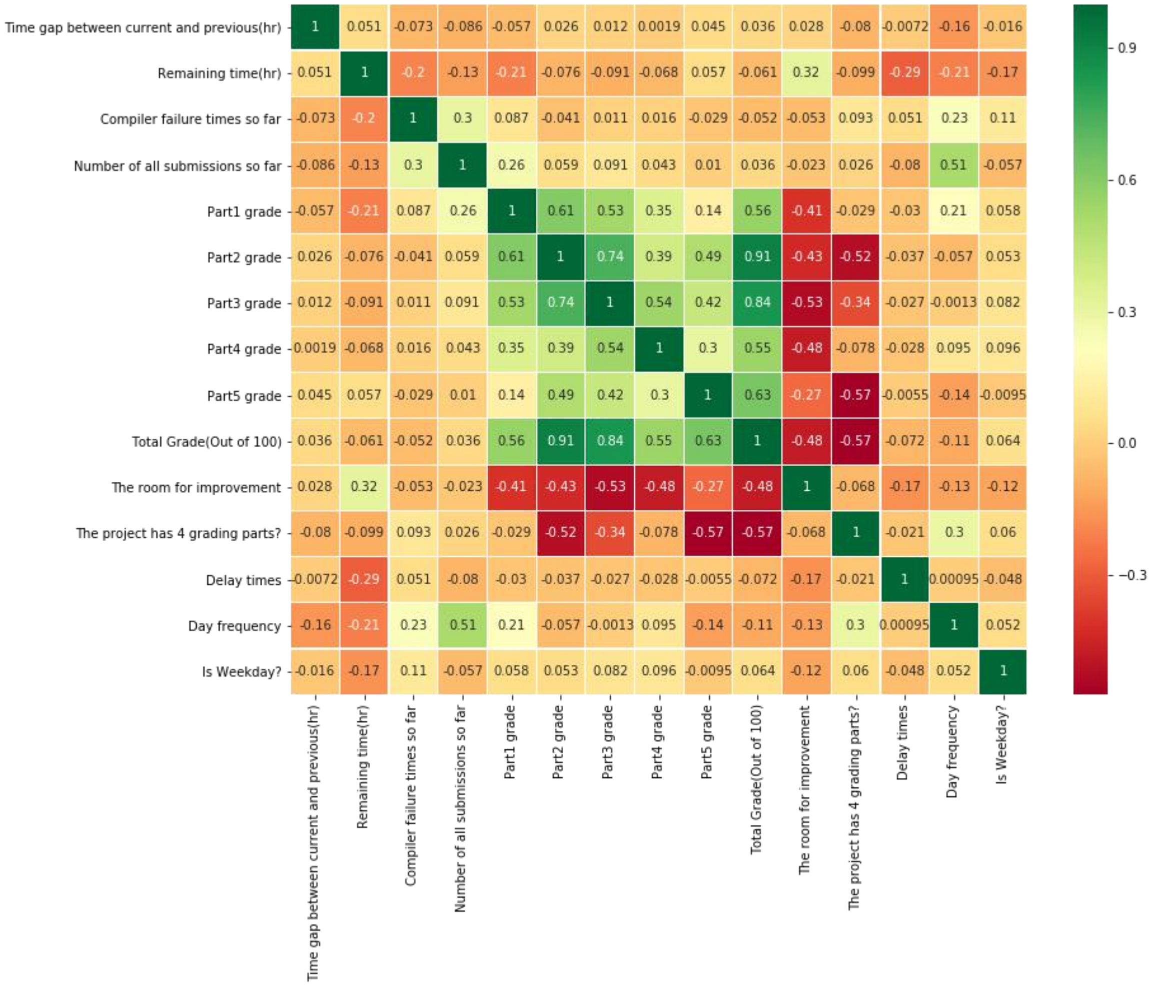


Figure 3.8: Feature Correlation

The Formula is:

$$\rho_{x,\,y} = \frac{\operatorname{cov}(X,Y)}{\sigma_x \sigma_y} = \frac{E((X-\mu_x)(Y-\mu_y))}{\sigma_x \sigma_y}$$

Cov(X, Y) is to find the co-variance of the array X and array Y. The figure 3.8 shows the correlation between every two features in our training dataset.

3.3 Permutation Importance

There are multiple ways to measure feature importance. During our experiments, we mainly use Permutation Importance and SHAP value impact to measure feature importance.

Permutation importance is calculated after a model has been fitted. So we won't change the model or change what predictions we'd get for a given value of height, sock-count, etc.

The way to do permutation importance is to randomly re-ordering a single column should cause less accurate predictions, since the resulting data no longer corresponds to anything observed in the real world. Model accuracy especially suffers if we shuffle a column that the model relied on heavily for predictions. In this case, shuffling height at age 10 would cause terrible predictions. If we shuffled socks owned instead, the resulting predictions wouldn't suffer nearly as much.

3.4 Baseline Model

Baseline Model is a model of predicting known problems and their data sets using simple heuristics, statistical rules, random rules, or previously used algorithms in the field. It is usually done before the formal work, providing a support for the performance of the later work to evaluate its performance, that is, the performance of the model proposed later is at least better than the baseline model.

Here we use Linear Regression as our baseline model. Linear Regression is a regression analysis that models the relationship between one or more independent variables and dependent variables using a least squares function called a linear regression equation. This function is a linear combination of one or more model parameters called regression coefficients. The case of only one independent variable is called simple regression, and the case of more than one independent variable is called multiple regression.

Linear Regression has advantages that the results are easy to understand, the computation is relatively easier. However, for non-linear dataset, the fitting of Linear Regression is poor.

3.5 Neural Network

Neural networks can help group unlabeled data, classify the data, or output continuous values after supervised training. Typical neural network applications in classification use logistic regression classifiers at the last level of the network (Converting a continuous value to a categorical value)

Figure 3.9 shows the screenshot of the dataset structure. During the data cleaning process, we removed several invalid data. If the student didn't make any submission, or the student drop the class during the course. We would mark them as invalid data and remove in order to reduce data noise.

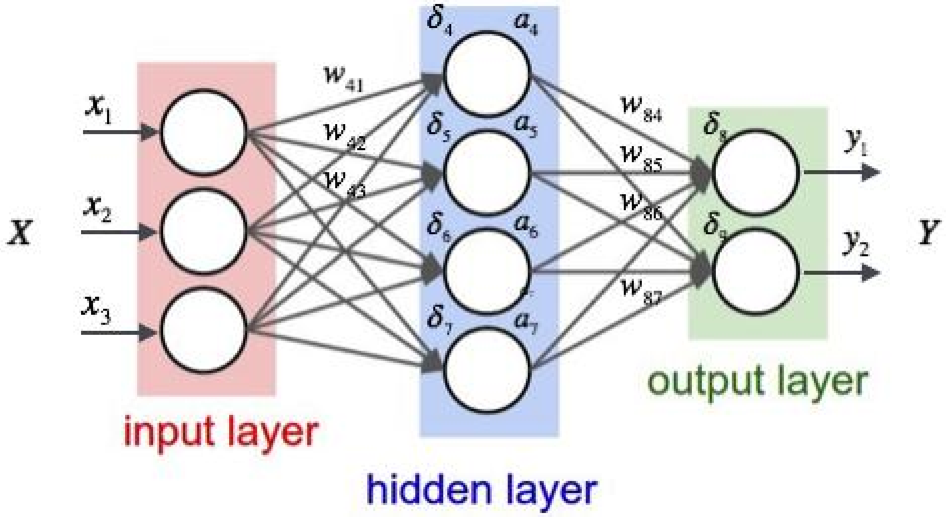


Figure 3.9: Neural Network

In the above figure, x represents the input, and the feature propagates forward in the layer in front of the network. Many x's are connected to each neuron in the last hidden layer, and each x will be multiplied by a corresponding weight w. These products and an offset are sent to an activation function ReLU (= max (x, 0)), which is a widely used as activation function, and does not appear as saturated as the sigmoid activation function. For each hidden layer, the neuron enters an activation value at the output node of the network and calculates the sum of these activation values as the final output. That is, using the neural network to do the regression will have an output node, and this node is only the front activation values of the nodes are added. The resulting ŷ is the independent variable obtained by all your x mappings.

3.6 Gradient Boosting Decision Tree

Decision tree is a basic classification and regression method. The decision tree model has a fast classification, and the model is easy to visualize, but at the same time it is easy to overfit.

In the classification problem, boosting learns multiple classifiers by changing the weight of the training samples (increasing the weight of the faulty samples and reducing the weight of the sampled samples), and linearly combining these classifiers to improve the classification performance.

Gradient Boosting is a method of Boosting. The main idea is that each time the model is built based on the gradient direction of the model loss function established. The loss function is to evaluate the model performance (generally the degree of fit + regular term), and the smaller the loss function, the better the performance. And let the loss function continue to decline, the model can be continuously modified to improve performance, the best way is to make the loss function down the gradient direction (the fastest decline in the direction of the theoretical gradient).

```
SET $F_0(x) = \arg\min_{\rho} \sum_{i=1}^{N} L(y,\rho)$ //Initial base learner
FOR m = 1 to M do :
    $-g_m(x_i) = -\left[\frac{\partial L(y_i,F(x_i))}{\partial F(x_i)}\right]_{F(x_i)=F_{m-1}(x_i)}$ , i = 1...N //Gradient direction
    $a_m = \arg\min_{\alpha,\beta} \sum_{i=1}^{N} \left[-g_m(x_i) - \beta h(x_i;a)\right]^2$ //Parameters in the regression tree
    $\mathrm{b_m} = \arg\min_{\beta} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \beta h(x_i;a_m))$ // Weighting factor of the regression tree
    $\mathrm{F_m}(x) = F_{m-1}(x) + v\beta_m h(x;a_m)$ //Update the prediction function
END FOR
END Algorithm
```

3.7 Accuracy Standard

The coefficient of determination means how much dependent variable obtained by the regression equation can be interpreted by the independent variable.

The coefficient of determination (R2) is also called the coefficient of determination or the goodness of fit. It is a representation of the extent to which the regression equation explains the variation of the dependent variable, or how well the equation fits the observation.

$$R^2(\mathrm{y},\hat{\mathrm{y}}) = 1 - \frac{\sum_{i=0}^{n_{samples}-1}(y_i-\hat{\mathrm{y}}_{\mathrm{i}})^2}{\sum_{i=0}^{n_{samples}-1}(y_i-\bar{y}_i)^2}$$

The greater the goodness of fit, the higher the degree of interpretation of the dependent variable by the independent variable, and the variation caused by the independent variable is higher than the percentage of the total change. The denser the observation point is near the regression line.

3.8 SHAP Value

For most machine learning-based projects, we always focus only on results, not on interpretability. But after all, people are not machines. They must convince people that machines are better than people. At least at this stage, interpretation is especially important. However, research in this area is obviously outdated compared to the various emerging neural network methods. Here we introduce the latest interpretability method SHAP Value [15] to explain our models' precision.

The shapley value method means that the income is equal to its own contribution and is a distribution method. It is commonly used for issues such as the rational distribution of benefits in economic activities. The introduction of the shapley value method has brought significant influence on the theoretical breakthrough of the cooperative game and its subsequent development.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] .$$

That is, the SHAP values of all features sum up to explain why my prediction was different from the baseline. This allows us to decompose a prediction in a graph like this:

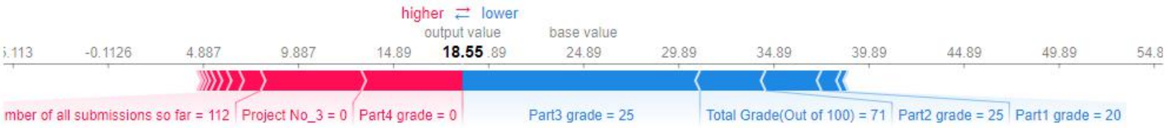


Figure 3.10: SHAP Summary Plot

The Figure 3.10 shows one sample point in our dataset. Here the average value of the output is 24.89. We have positive features like part4_grade, number of all submissions, negative features like part3_grade, total grade, part2_grade. Based on their interaction, the final result becomes 18.55.

The meaning of the shaley value of each dimension feature is: the greater the value, the more positive the effect on the objective function, and the smaller the value, the more negative the impact on the objective function.

Chapter 4

EVALUATION

In this chapter, we will focus on the data analysis, training results, and model interpretation. We evaluate the results based on training precision, and cross-validation score of three different machine learning methods: Linear Regression, Neural Network, and Decision tree. We record the max value, min value, mean for model comparison. Here are the evaluation objectives:

1. The prediction precision: How accurate are our algorithms on real data with different parameter settings (measured by R-2 score)?

2. Interpretability: Can we use SHAP value to explain the internal logic of the non-linear models?

After getting the accuracy of all models, we'll pick the best model to analysis. If the model is linear regression, we can directly use variable weight to indicate the feature importance. If the model is tree-based or non-linear, we'll introduce SHAP method to analysis local interpretability.

Figure 4.1 shows the parameter settings of the training process, and Figure 4.2 shows the flowchart of our evaluation. Here we use 7 different learning rate and 2 folding patterns to train the predictor. According to the Shapley value, every time we input data in both interpreter and predictor. The predictor model will give us the accuracy value, and the interpreter will illustrate the impact of both negative features and positive features. Also, we will use feature dependence plot to explain the relationship of every two features.

Table 4.1: Parameter Settings

Learning Rates	0.001, 0.03, 0.1, 0.3, 1, 3, 10
Max Iterations	500
Folds	5, 10

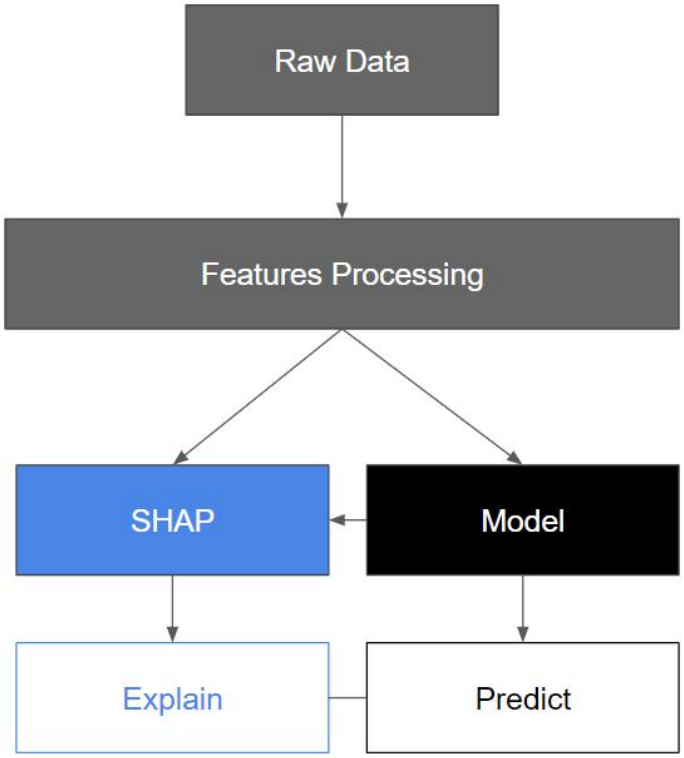


Figure 4.1: Experiment Flowchart

4.1 Training Results

As we clarified in chapter 3, we mainly use 3 machine learning algorithms to generate the predictor: linear regression, Neural Network, and Decision tree. So we will get at least 3 group data for comparison. Here we applied 2 frameworks (XGboost and lightGBM) to train the decision tree, and we stored the basic statistics metrics: mean, max, min, and gap.

The table 4.2 shows the summary of cross-validation training, we find out the decision tree with XGboost always gets the best result: average precision is 82.65%, the optimal case is 84.34%, the worst case is 81.46%, and gap is 2.88%, which means the GBDT method is more stable and can achieve better precision.

Table 4.2: Summary of Cross-validation

Method	5-Fold				10-Fold			
	Avg	Max	Min	Gap	Avg	Max	Min	Gap
LR	59.96	62.42	58.04	4.38	59.93	63.69	56.71	6.98
NN	68.76	71.23	66.36	4.87	68.67	73.45	64.00	9.45
XG	82.65	84.34	81.46	2.88	82.90	85.93	79.21	6.72
LGBM	77.79	80.25	75.99	4.26	78.01	81.47	74.88	6.59

Also we observe that the linear regression always generates lowest result. That baseline method does have advantage of easy interpretability, but it also reflects the non-linear property of the real-world dataset after comparison with other curve-fitting method like neural network and decision tree.

4.2 Model explanations

As we explained in chapter 3, we introduced SHAP value to interpret the high-accuracy model.

If we take many explanations such as the one shown above, rotate them 90 degrees, and then stack them horizontally, we can see explanations for an entire dataset. Some data sample has below-average predictions because of the overall negative feature impact. If we dive into each feature, we can find new results.

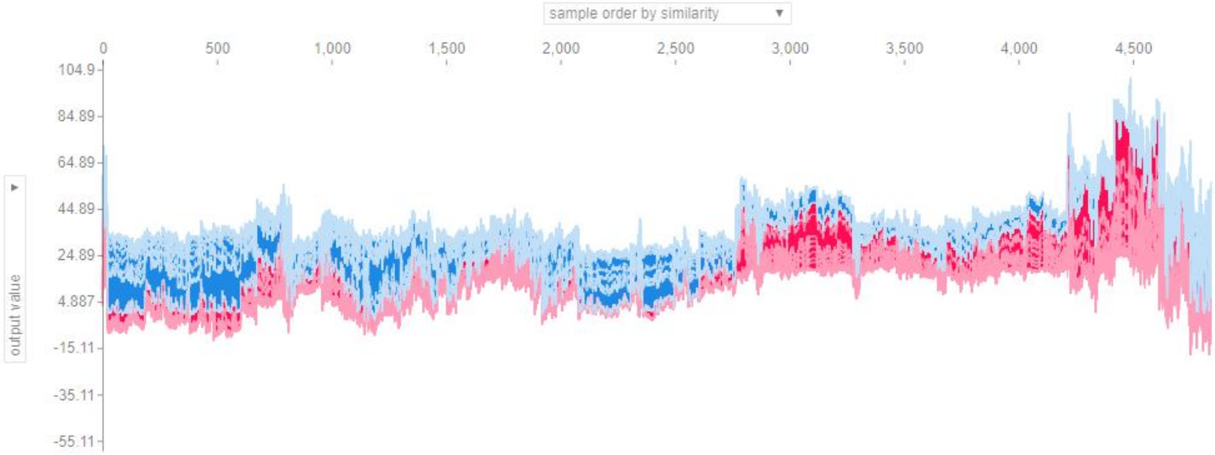


Figure 4.2: SHAP Summary of all samples

Figure 4.3 shows the overall impact of each feature contributing the SHAP output. Figure 4.5 - Figure 4.7 shows the dependence between 3 noticeable features and the total grade. If the dependence can be consistent with the feature correlation, we could say that our model interprets the dataset correctly.

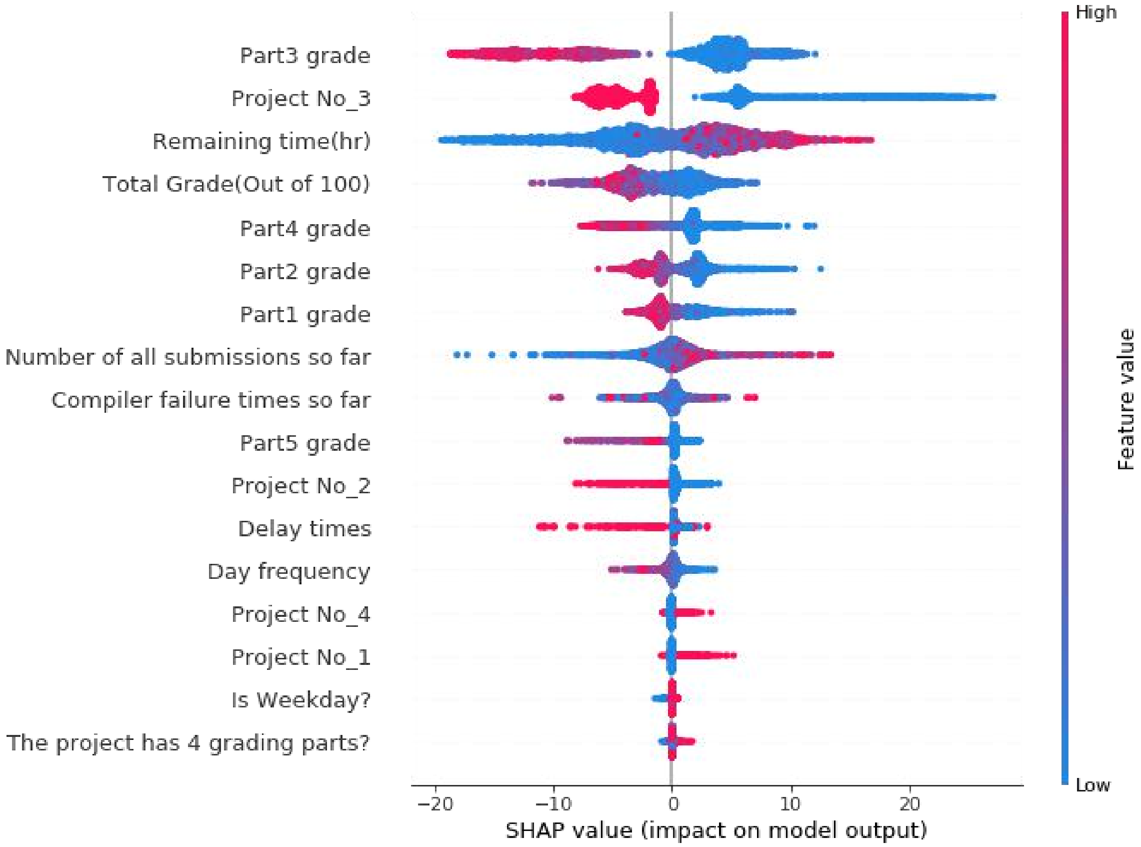


Figure 4.3: Summary of of all feature effects

According to the Figure 4.3, the part3 grade overall has higher impact than other features, which means the change of part 3 can have more noticeable influence than others. If the part 3 grade is higher, the room for improvement would be reduced accordingly.

If we look at the Remaining time, we can find that the closer the deadline is, the less improvement can be made. For the Total submission, we find similar result that the higher submissions would increase the room for improvement.

The Figure 4.4 shows the feature impact in bar chart, here the impact in descending order is: part3 grade, remaining time, total grade, part4 grade, part2 grade, part1 grade, total submissions, failure times, part5 grade, delayed days, and day frequency, which is consistent with the Figure 4.2.

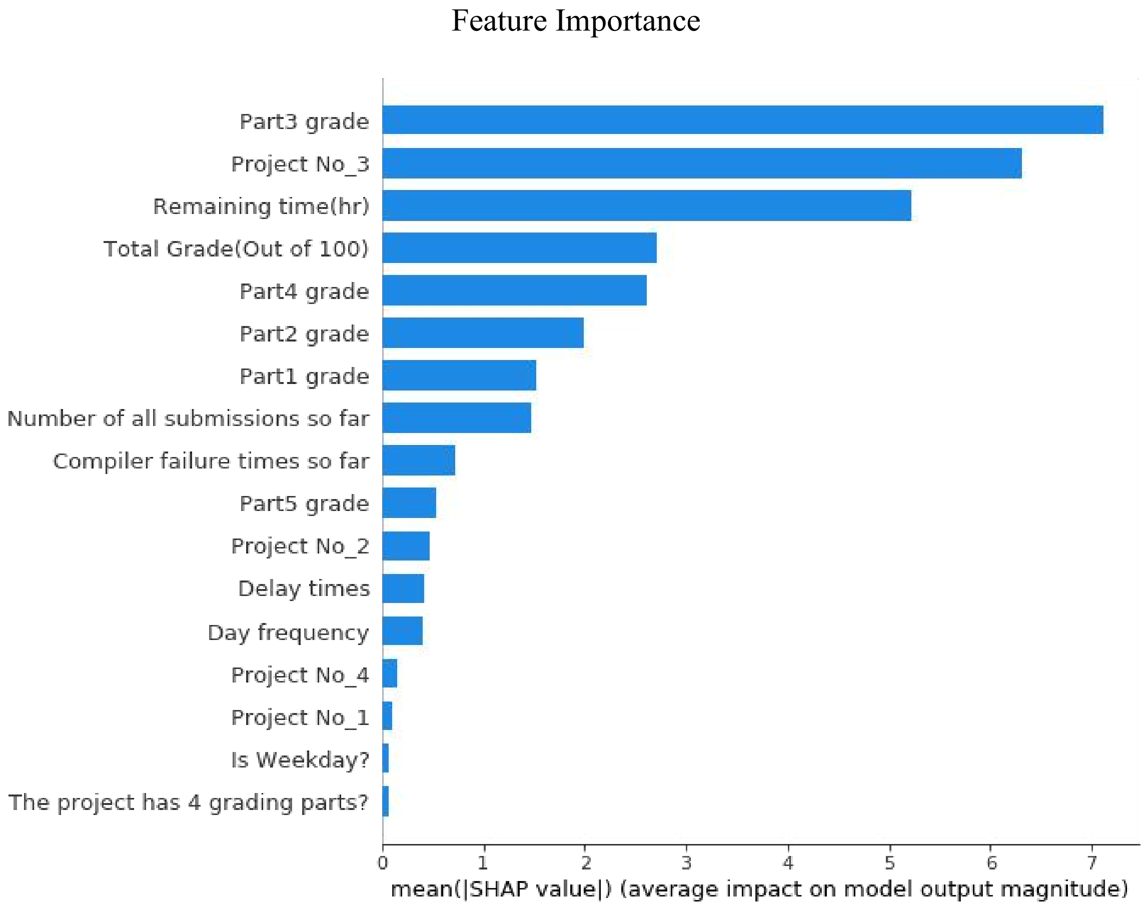


Figure 4.4: Feature impact ranking

After computing every local SHAP value for every submission, we also can analyze the dependence between every pair of features by mapping all specific pairs of features on coordinate axis.

The Figure 4.5 shows the relationship between feature ‘Remaining time’ and feature ‘Total grade’. According to left bottom corner of the figure, we can say that the closer the deadline is, the room for improvement will be greatly reduced so that early submission would result in good grade. If we go through x coordinate from left to right, the overall trend is the earlier submissions can generate higher grades.

The feature dependence

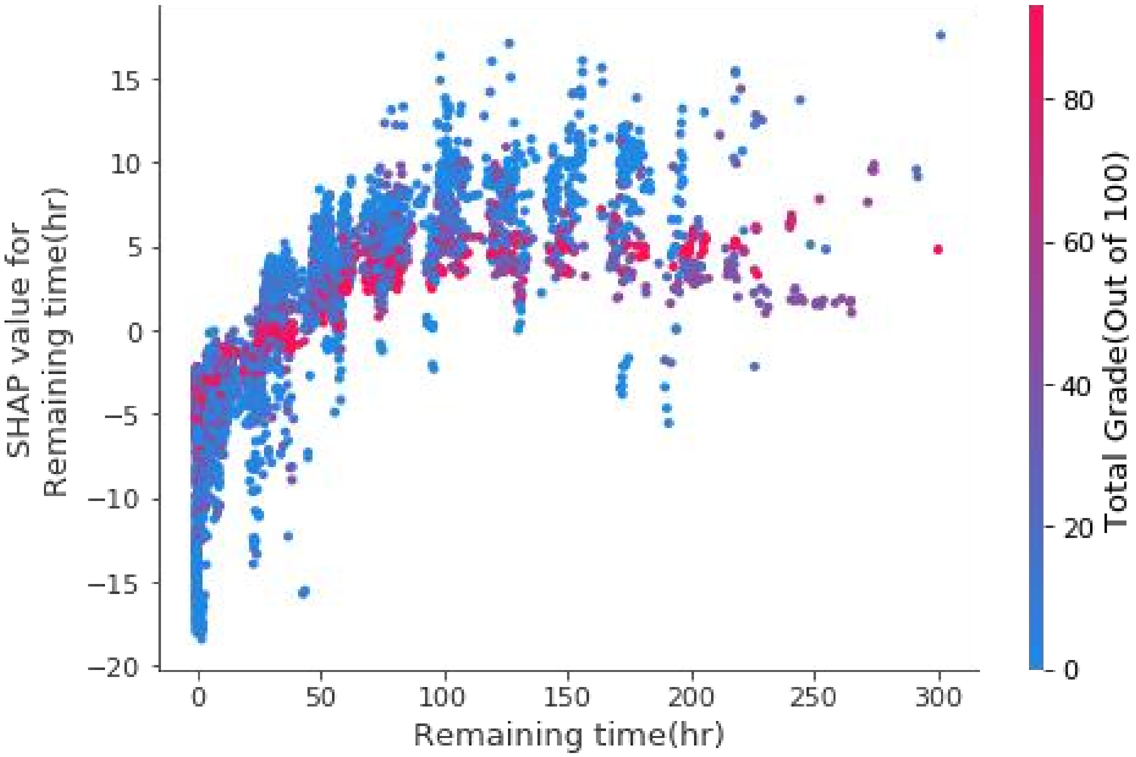


Figure 4.5: The dependence contribution between Reaming hours and Total grade

The Figure 4.6 shows the relationship between feature ‘Number of all submissions so far’ and feature ‘Total grade’. Here we notice that low-grade density is much higher during the submission 0-50. The more submission made, the higher the grade should be. Overall the good-grade samples don’t have a huge influence on the result because SHAP value is not big enough. But during submission 0-50, the lower grade would decrease the room for improvement.

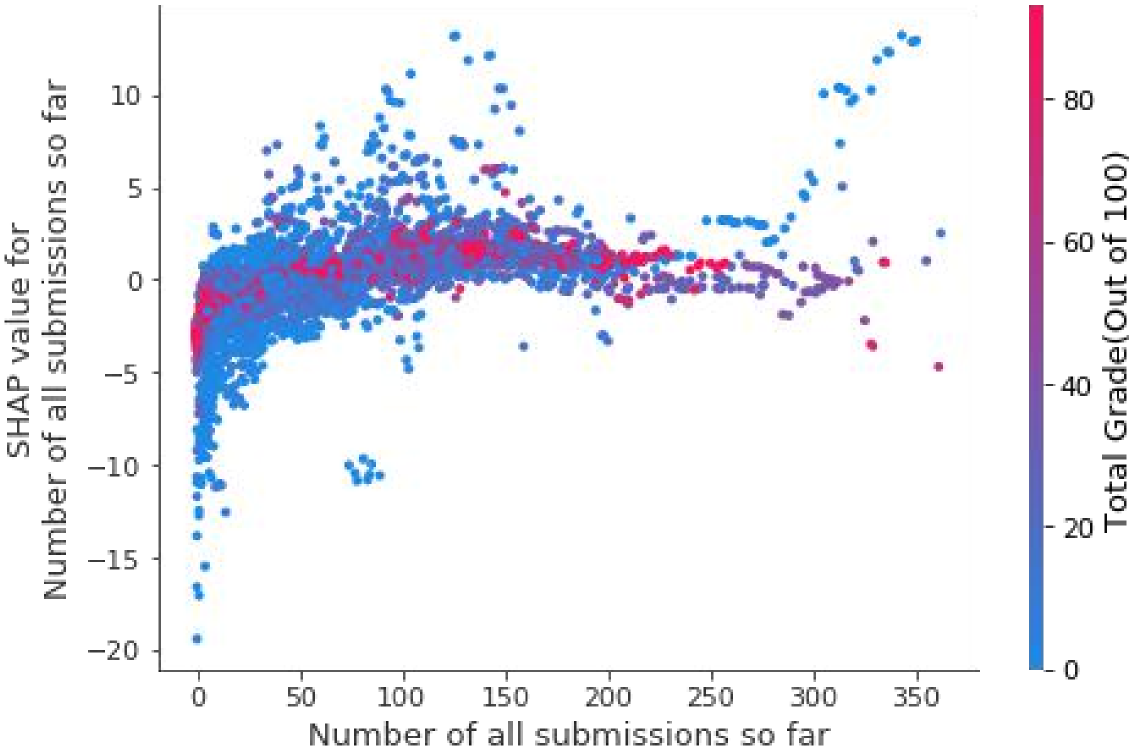


Figure 4.6: The dependence contribution between submission times and Total grade

The Figure 4.7 shows how feature ‘Delay times’ have impact on the output. We observe that the higher ‘Delay times’ is, the lower the SHAP value is, which is consistent with our grading rule that delayed submission would have penalty to the maximum grade. Therefore, the feature ‘Delay times’ always has negative effect to our prediction ‘the room for improvident’.

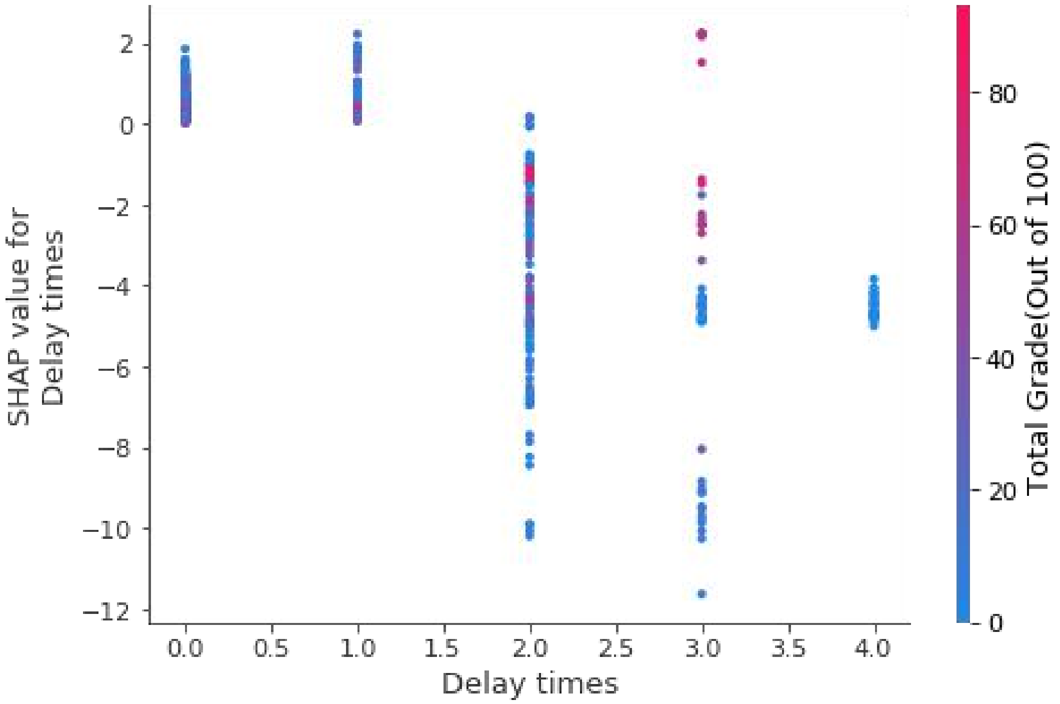


Figure 4.7: The dependence contribution between Delayed days and Total grade

The Figure 4.8 - Figure 4.12 show the dependence relationship between each component grade and the total grade. We can see from figures that although each component topic is different, the overall trend is higher component grade will decrease the room for improvement, which is consistent with our assumption that top performer could not improve much more than low performer. Besides, from Figure 4.8 - Figure 4.10, we find very similar results that more red data points come out as each component score get higher, which means good final grade is caused by good component grades.

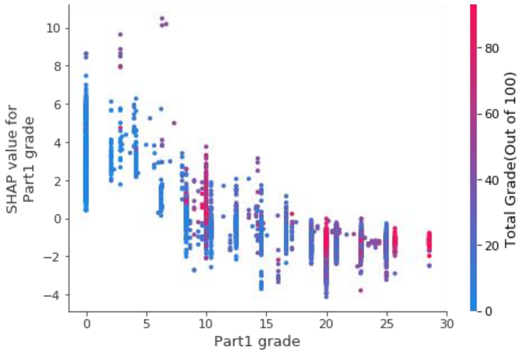


Figure 4.8: The dependence between G1 and total grade

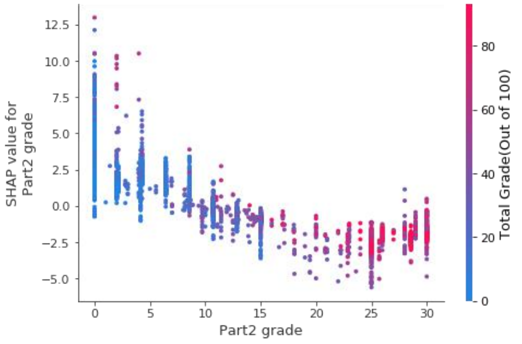


Figure 4.9: The dependence between G2 and total grade

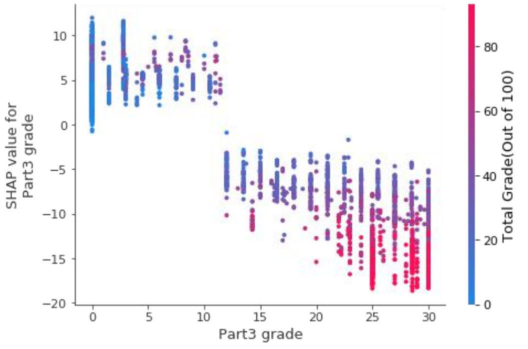


Figure 4.10: The dependence between G3 and total grade

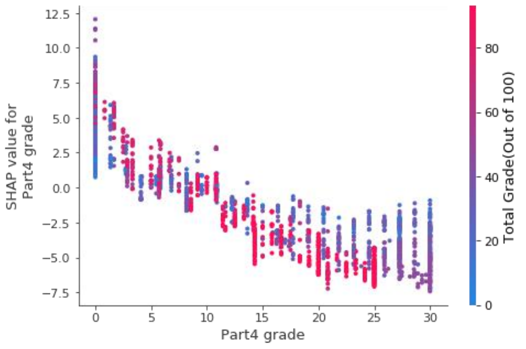


Figure 4.11: The dependence between G4 and total grade

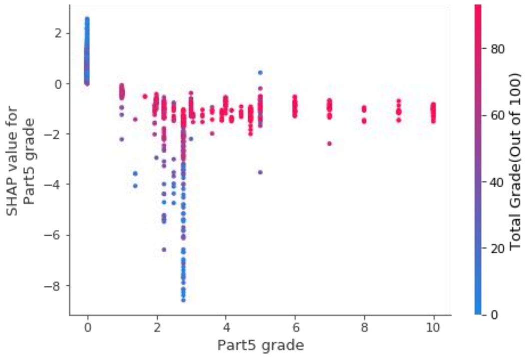


Figure 4.12: The dependence between G5 and total grade

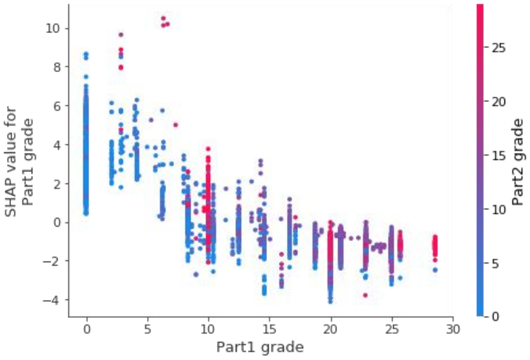


Figure 4.13: The dependence between G1 and G2

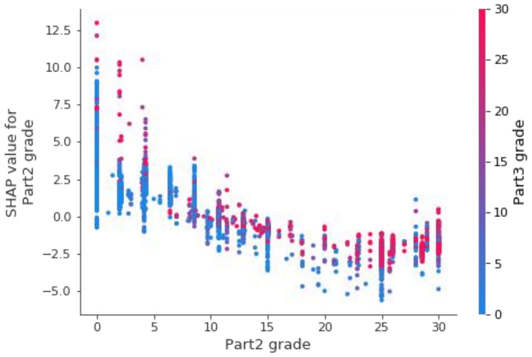


Figure 4.14: The dependence between G2 and G3

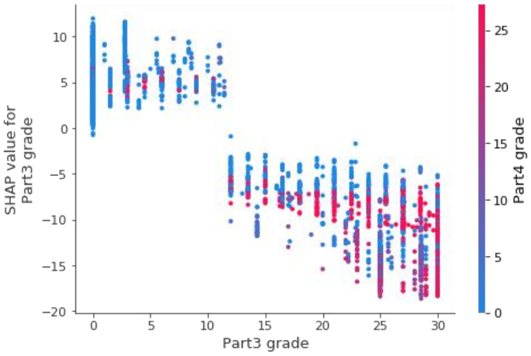


Figure 4.15: The dependence between G3 and G4

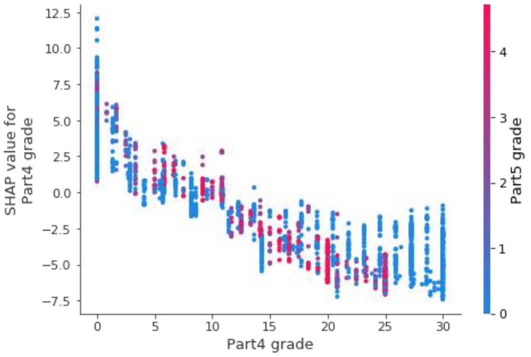


Figure 4.16: The dependence between G4 and G5

The Figure 4.13 - Figure 4.16 show the dependence relationship between every two adjacent component grades. Comparing Figure 4.8 and Figure 4.13, Figure 4.9 and Figure 4.14, we find nearly the same results on all data points. But the meaning of the figure is that the higher previous component grade would result in higher next component grade. By looking at the right corner of Figure 4.16, we find a lot of low-grade points. One possible reason is that the task with only 4 tasks are also be included, and these data have one unused feature marked as 0.

Chapter 5

CONCLUSION

5.1 Summary

After years of development, Educational Data mining research has achieved considerable results, and gradually formed a basic theoretical basis, including: classification, clustering, pattern mining and rule extraction. Educational Data mining is a technology that "digs out" potential, unprecedented knowledge from the vast amounts of data in courses. In this work, I propose the data mining pipeline to predict students' performance based on CSE340 dataset. I build feature engineering by analyzing feature importance and feature correlation, compare different data mining algorithms and do detailed analysis based on the precision value. Finally, I introduced emerging technique to improve interpretability of the high-accuracy model.

5.2 Discussion & Educational Implications

This section will discuss the results analysis and model explanation in predicting students' performance. As per evaluation results in section 4.2, Gradient Boosting Decision Tree in XGboost has the highest average prediction precision by (82.90%) followed by Gradient Boosting Decision Tree in Light GBM by (78.01%). Next, Neural Network gave the precision by (68.67%). Lastly, the method that has lower prediction precision is Linear Regression by (59.93%). These values show that we can predict students' performance and improve prediction by applying different data mining methods.

Boosting Decision Tree and Neural Networks are usually considered less suitable for data mining purposes, because knowledge models obtained under these paradigms are usually considered to be black-box mechanisms, able to attain very good accuracy rates but very difficult for people to understand. However, after we introduce the Shapley Additive Explanations, both of methods can be explained in a consistent way. By looking at the Figure 4.5, for both low scores and high scores, the feature ‘Remaining Time’ has higher negative impact when the time is close to due date, which means the score would become stable as time goes by. Figure 4.6 shows submissions of low performers is much less than submissions of high performers, and data points during 0~50 have much higher negative impact than others. One possible reason could be novices may not put enough effort to prove they can achieve high grade. For experienced students, the total submissions would have positive effect when they make mistakes or get lower grade.

As a result, getting the prediction and explanation generated through our experiment makes educators be able to identify students at risk early, especially in big programming classes. Also, it allows educators to provide appropriate advising in a timely manner.

As a data mining project, this data processing pipeline is scalable. Since other programming assignments have similar grading features and time features, it is possible to be extended to other projects like object-oriented programming, and Java Programming.

5.3 Limitations & Future Work

The main limitations of EDM is the dataset. In this research, we use the dataset from CSE340 course at Arizona State University. However, for further research, EDM lacks public datasets. Most EDM literature does not currently publish research datasets on the Internet or attached to papers. Researchers are reluctant to disclose datasets for two main reasons: First, datasets involve the privacy of research subjects, Academic ethics and legal regulations are not suitable for publication; second, the acquisition of data sets consumes a lot of time, manpower and economic costs, which is a valuable asset for researchers. However, for researchers, not publishing data sets may reduce research results. Reliability and impact; for the EDM research community, the lack of public data sets can hinder the development of EDM research. We recommend that EDM researchers share more educational dataset based on a combination of privacy protection, economic input, and academic significance.

For model interpretability, the Shapley value method needs to traverse the "all possible combinations" of the variable set. when the number of variables is large, the number of combinations is very large, resulting in a large amount of Shapley value calculation and a huge time complexity.

For future work, there are different educational dataset that can be tested by our method. Also, if we can be given big dataset, we can use latest big data technology to generate new model and observe the results.

REFERENCES

[1] Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. JEDM| Journal of Educational Data Mining, 1(1), 3-17.

[2] Anjewierden A, Kolloffel B, Hulshof C. Towards educational data mining: Using data mining methods for automated chat analysis to understand and support inquiry learning processes. In: Proc. of the Int'l Workshop on Applying Data Mining in e-Learning (ADML 2007). 2007.

[3] Cole J, Foster H. Using Moodle: Teaching with the Popular Open Source Course Management System. 2nd ed., O'Reilly Media, Inc., 2007.

[4] Lara JA, Lizcano D, Martínez MA, Pazos J, Riera T. A system for knowledge discovery in e-learning environments within the European higher education area — Application to student data from open university of madrid. UDIMA. Computers & Education, 2014,72:23-36.

[5] Worldwide smartphone user base hits 1 billion. 2012.

[6] Facebook users reach 2.2 billion, one third of the global population. 2014.

[7] Partho Mandal and I-Han Hsiao. (2018) Using Differential Mining to Explore Bite-Size Problem Solving Practices. Educational Data Mining in Computer Science Education (CSEDM) Workshop, 2018.

[8] Mohammed Alzaid I-Han Hsiao. (2018) Personalized Self-Assessing Quizzes in Programming Courses. Educational Data Mining in Computer Science Education (CSEDM) Workshop, 2018.

[9] Yancy Vance Paredes, David Azcona, I-Han Hsiao, Alan F. Smeaton. (2018) Predictive Modelling of Student Reviewing Behaviors in an Introductory Programming Course. Educational Data Mining in Computer Science Education (CSEDM) Workshop, 2018.

[10] Rui Zhi, Thomas W. Price, Nicholas Lytle, Yihuan Dong and Tiffany Barnes. (2018) Reducing the State Space of Programming Problems through Data-Driven Feature Detection. Educational Data Mining in Computer Science Education (CSEDM) Workshop, 2018.

[11] Coursera. https://www.coursera.org/

[12] Romero C, Ventura S. Data mining in education. Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery, 2013, 3(1):12-27.

[13] Hand DJ, Mannila H, Smyth P. Principles of Data Mining. The MIT Press, 2001.

[14] Peng Y, Kou G, Shi Y, Chen Z. A descriptive framework for the field of data mining

and knowledge discovery. Int’l Journal of Information Technology & Decision Making, 2008,7(4):639-682 .

[15] Goda, Y., Yamada, M., Kato, H., Matsuda, T., Saito, Y., & Miyagawa, H. (2015). Procrastination and other learning behavioral types in e-learning and their relationship with learning outcomes. Learning and Individual Differences, 37, 72-80.

[16] Cohen, M. S., Yan, V. X., Halamish, V., & Bjork, R. A. (2013). Do students think that difficult or valuable materials should be restudied sooner rather than later? Journal of Experimental Psychology: Learning, Memory, and Cognition, 39(6), 1682-1696.

[17] Kristopher J. Preacher, Patrick J. Curran, Daniel J. Bauer. (2006). Computational Tools for Probing Interactions in Multiple Linear Regression, Multilevel Modeling, and Latent Curve Analysis Journal of Education and Behavioral Statistics

[18] Benesty J., Chen J., Huang Y., Cohen I. (2009) Pearson Correlation Coefficient. In: Noise Reduction in Speech Processing. Springer Topics in Signal Processing, vol 2. Springer, Berlin, Heidelberg

[19] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems 30. Curran Associates, Inc., 4768–4777.

[20] Breiman, Leo, Friedman, Jerome, Stone, Charles J, and Olshen, Richard A. Classification and regression trees. CRC press, 1984.

[21] Chen, Tianqi and Guestrin, Carlos. Xgboost: A scalable tree boosting system. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM, 2016.