Walter P. Moore MIDAS Data Challenge

Team 1: Zhihao Guo, Hongxin Yang, Gang Yang, Jen Sheng Wong

Abstract

In this project, we use data collected from Walter P. Moore's historical projects to model a project's profitability with high precision, and also identify the metrics that are good predictors of the project's profitability. We pre-process the data, conduct exploratory data analysis and perform feature engineering, then apply some classification algorithm based on our understanding of the data and the question. Models include Logistics Regression, Random Forest, XGBoost. We found XGBoost to have the best performance as well as strong interpretability. From the model inference, we are able to determine the important variables that are highly influential on the profitability of the projects, and we visualize it.

Introduction

Walter P. Moore is an international company of engineers, architects, innovators, and creative people who solve some of the world's most complex structural, technological, and infrastructure challenges. Each year, the company received contracts to do projects, which they price out using various methods. The goal of this project is to utilize the available data and variables the company provided us and to predict the profitability of any project, and determine the features that are related to the project's profitability.

Methods

Data Processing

Monotone Columns and Missing Values

The original data has a shape of the 464585 rows and 86 columns. Since this is a classification problem, and we decided to use the decision tree method to model it, we deleted the monotone columns. A monotone column means that the majority values in this column are the same, which will end up having little contribution to the decision tree splitting. We choose 70% as the threshold, picking columns where the most frequent value weighs more than 70% of this column. After that, we handle the missing values. There are three columns with missing values: *ClientCode*, *ClientPostalCode*, and *ProjectEngineer*, where *ClientPostalCode* is a monotone column and already been deleted. We filled the missing value of *ProjectEngineer* with an empty string, saying that we don't know the ID of this project engineer, and we simply dropped the missing *ClientCode*.

Sub-Projects and Project Family

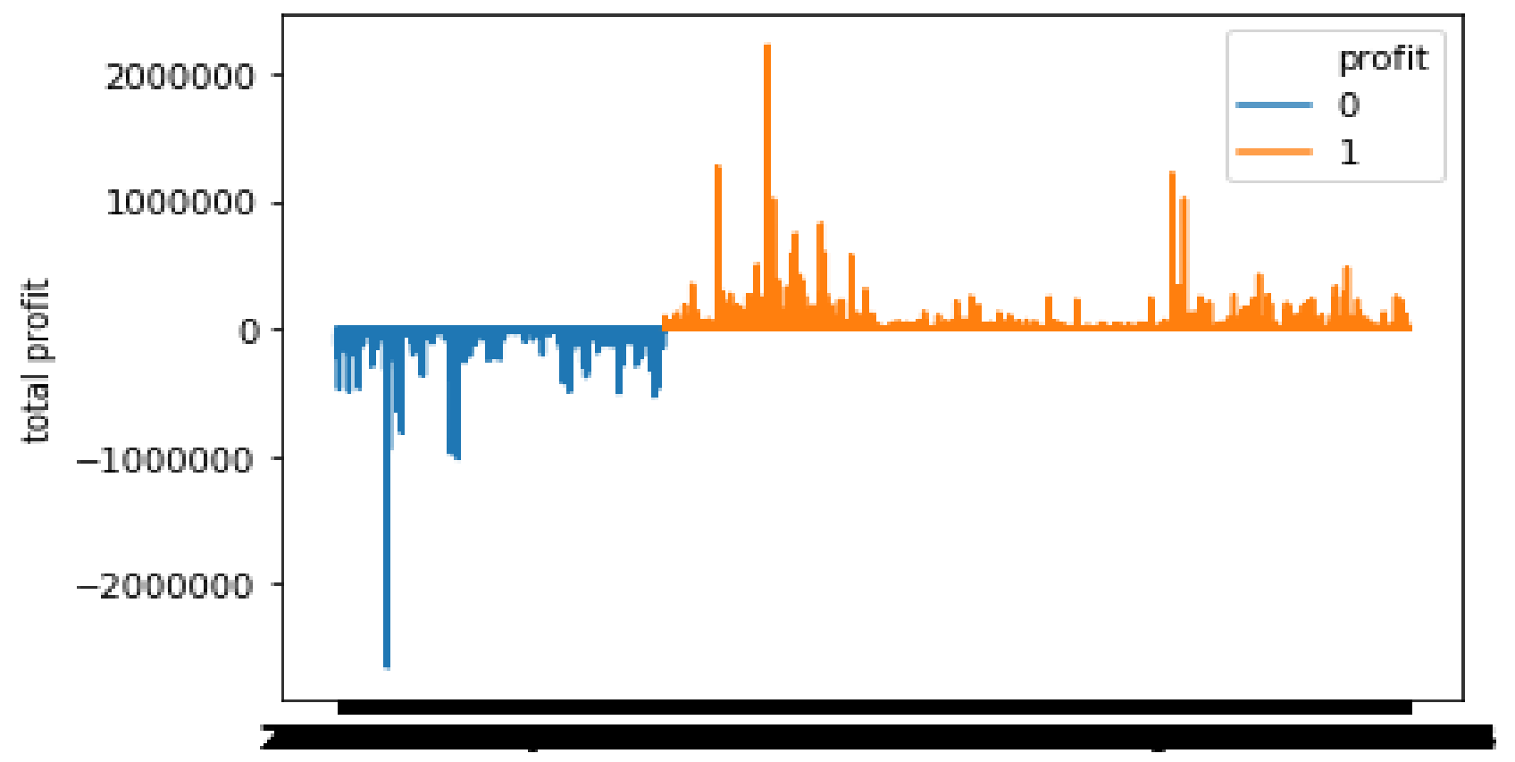
We group the sub-projects into their project families and treat them as one project. There are 16672 unique sub-projects and can be grouped into 10126 project families.

For each project, it has variables with suffix "Family" indicating the values for its family. However, we found that it is not reliable to directly use "Family" variables for the project family. For example, for project family "2300002", the variable FamilyCostPTD is not equal to the sum of the CostPTD of all its sub-projects. Meanwhile, we also determined that the "PTD" variables are not reliable as well. Using the same example, the CostPTD is not monotonically increasing and not equivalent to the sum of the previous CostMTD. To solve this, we calculated those "Family" variables for ourselves using "MTD" variables. To calculate the FamilyCostPTD for project family "2300002", we used the CostMTD for its sub-projects in all accounting periods and sum them up overall sub-projects. For other string objects, such as the location of the company, they are unique for each project family so we keep it still. The result dataframe has a shape of 10126 rows and 29 columns, where each row indicates a single project family.

Results

Profitability Calculation and Encoding

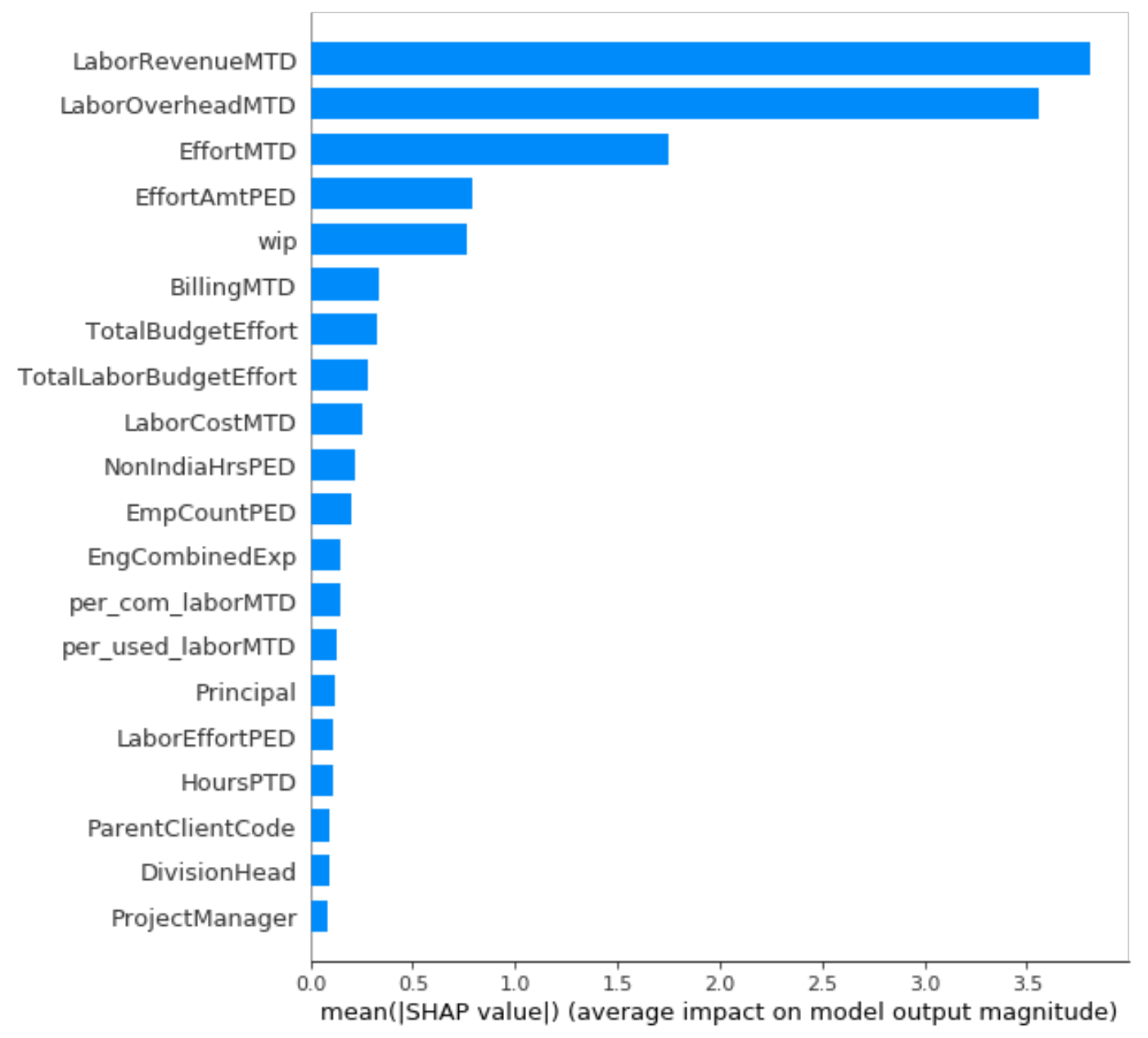
We calculate the *profit*, *wip*, *% Complete Labor*, *% Used Labor* using the formula provided in the spec, and we encode the profit into a binary variable based on its negativity. A value 1 means this project family is profitable and vice versa. The result shows that there are 7029 profitable projects and 3097 projects where the company lost money. The ratio of profitable versus unprofitable is around 2.27:1. A close look at the values are presented in the graph below:

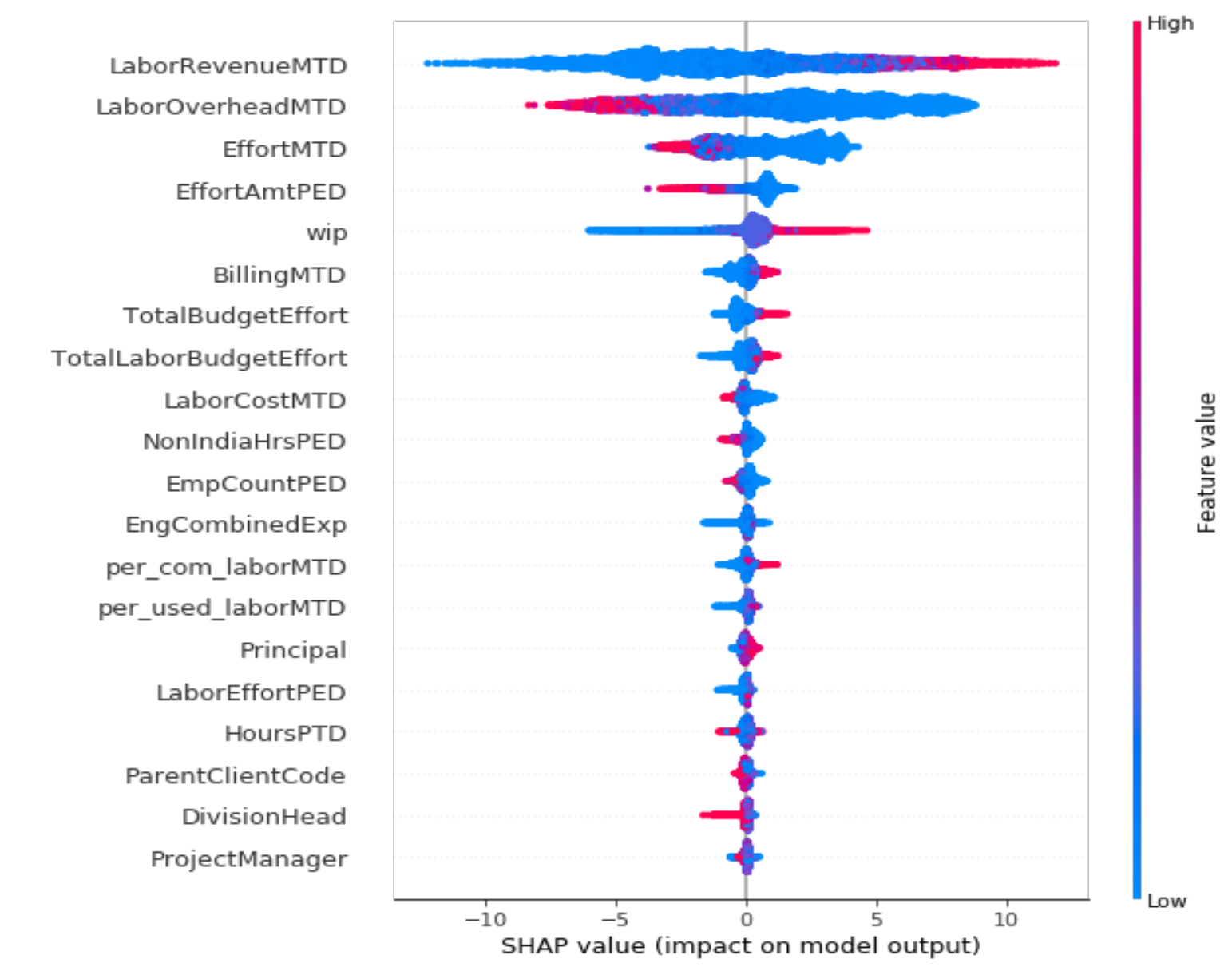


Profit and loss for all the profit families

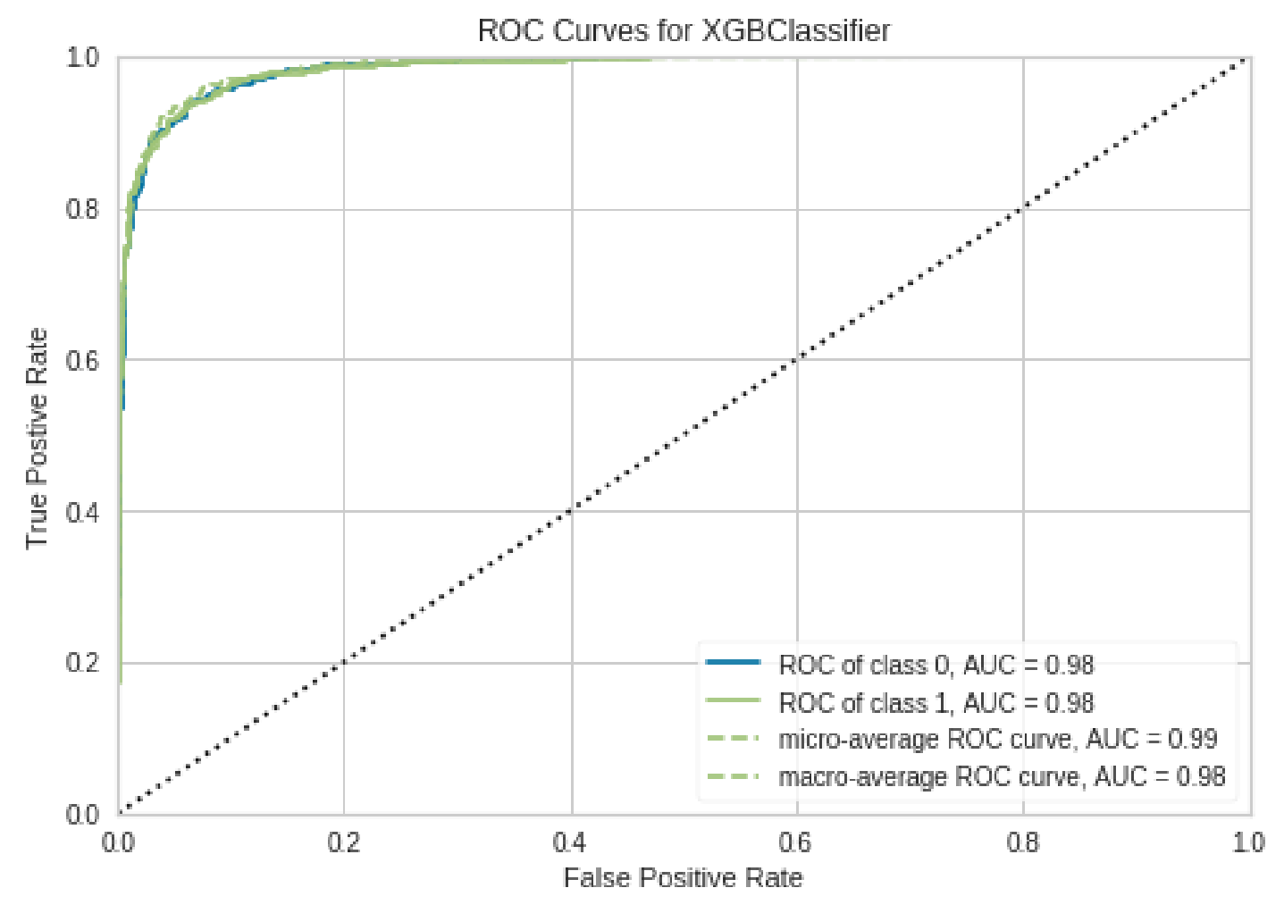
For the actual profit, the overall profit for this dataset is 32584396, while on average each project family will have 3218 profit. The average gain of profit projects is 12765 while the average loss for unprofitable projects is 18451. Some projects are extremely profitable (2m profit) and also some projects that are huge failures (2m lose).

The SHAP plot is also displayed below. The x-axis is the SHAP value, which means the average importance of this variable in the model, giving all different possible orders of the splits. The y-axis is the variable. The color means the actual value of the variable, and the width of the shape indicates the number of points it has to the certain variable. For example, the majority data of LaborRevenueMTD lays above SHAP value = -5, and high value will push the classification to positive.

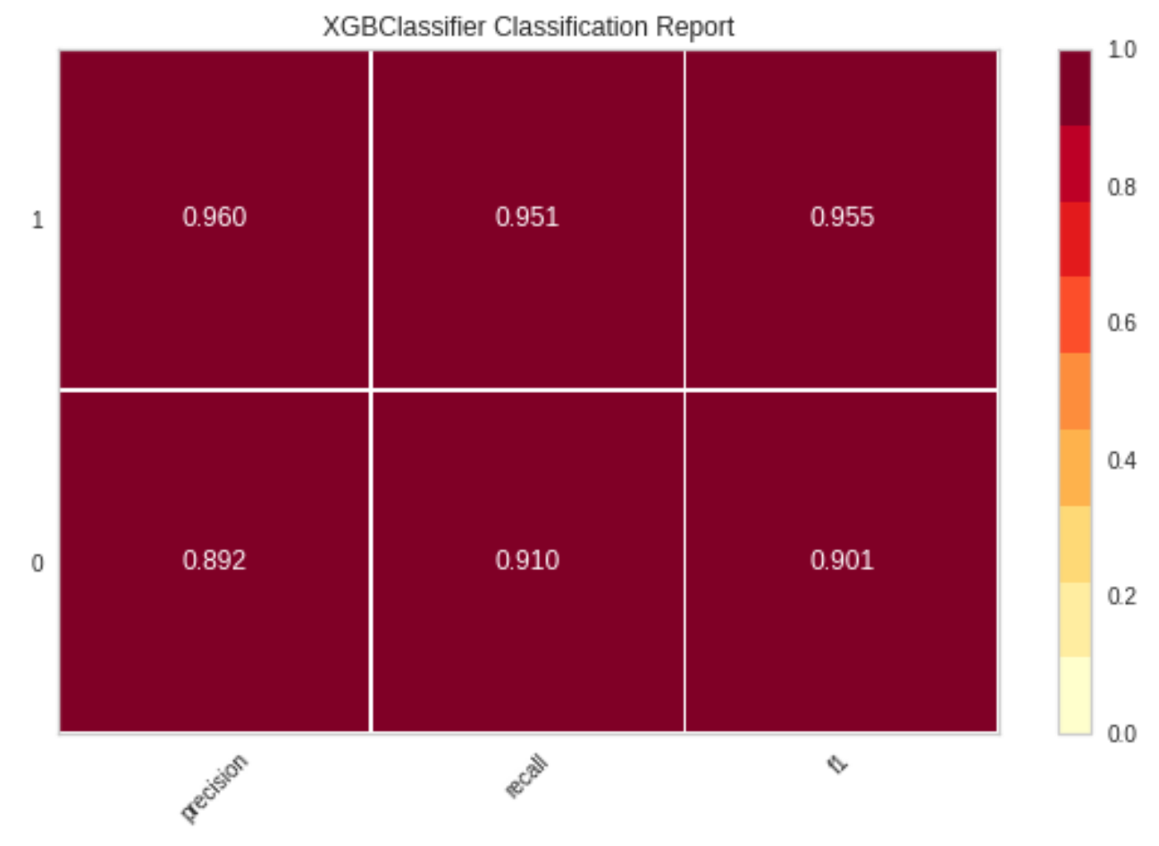




Last but not least, we use different metrics to evaluate our model. Note that Walter P Moore has evaluated these data before using black-box methods with approximately 90% accuracy. Our model has an accuracy of 94% from cross-validation, with a standard deviation of 0.1. Moreover, we also calculate the Recall, Precision, F1-score, and plot the ROC curve to get the AUC value. The AUC is about 0.99, and all precisions and recalls are above 0.9.



ROC Curve



Recall, Precision, and F1-score

Model

Great Interpretability -- Decision Tree Methods

To balance the interpretability and the performance, we used ensemble decision tree algorithms like Random Forest and Gradient Boosting. Decision Tree is of great interpretability because the information gain naturally explains the importance of features and do the feature selection.

Variance-Bias Tradeoff -- Random Forest and Gradient Boosting

Ensemble algorithms fix the deficiency of overfitting of a single decision tree and maintain the advantage of interpretability. Common Ensemble methods include bagging(Random Forest) and Boosting(Gradient Boosting). Random Forest reduces the variance by bootstrapping aggregation but a little increases bias. Comparing to the random forest, Gradient Boosting generates a model with lower bias but taking more training time and it's harder to tune. In this case, we took both models and the Gradient Boosting had greater performance. We also used 10-Fold Cross-Validation to tune and evaluation both of our models.

XGBoost Results and Variable Importance Plot

Powerful interpretability of Decision Tree is we can generate variable importance plot from it, which tells what variables contribute most to the tree split and heavily influence the prediction. The variable importance plot of XGBoost is shown. The higher position a variable is placed on the graph, with a long vertical bar, means more important this variable is to predict the profitability.

Importance Indicator -- SHAP (SHapley Additive exPlanations)

SHAP is a cutting-edge method that can give the numeric metric of the impact of features and data points. Mathematically, SHAP is a kind of local linear model fitting the data and maintaining the local accuracy and consistency. Comparing to the natural importance(information gain) of Decision Tree, SHAP is easier to explain because it not only has an effect on the feature but also explains the impact of every single data point. The formula of SHAP shown below, the coefficient $\phi_i \in \mathbb{R}$. SHAP value of i-th feature among data point z'

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$

Consistency

The original importance doesn't maintain the consistency, which means different models will give uncomparable values of importance. SHAP gives a measurable result and maintains the measurement, that is, SHAP values satisfy the following property. For different models, f and f', if $f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$, then $\phi_i(f', x) \geq \phi_i(f, x)$

Conclusions

We see that most projects are profitable and the company is profitable overall, so the company is on the right track and making correct decisions. However, when there is a loss, the company tends to lose more than what they earn when there is a gain since the average loss is higher than the average gain. This suggests that the company needs to do its best to avoid picking unprofitable projects since it will significantly jeopardize the company's finance and will eat up the profit of more than one profitable project. Lastly, LaborRevenue, LaborOverhead, and Effort and the top three most influential variables to the profitability. Among them, only LaborRevenue is positively related to profitability while the other two will decrease the profit when increase. However, most of the LaborRevenue is pretty low, which is still dragging the profit downwards. Lastly, we successfully build a classifier that outperforms the existent classifier that the company has, as well as strong interpretability and good classification ability with high precision and recall. It could be used as an alternative when the company wants to determine the profitability of a new project.

The recommendations we advise the company is to increase its LaborRevenue while decreasing the LaborOverhead and the Effort. Also, pay attention to other variables such as wip and Billing too.

The key limitation of our work is the lack of time series analysis. Thus, in the future, we would like do time series analysis, such as study the auto-coefficient of the important variables, and fit time series model such as ARIMA.

Poster produced by Faculty & Curriculum Support (FACS), Georgetown University School of Medicine