MIDAS Midpoint Report

1. 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.

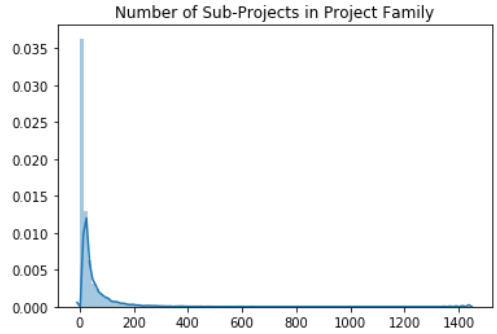
2. Data Preprocessing and Exploratory Data Analysis

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

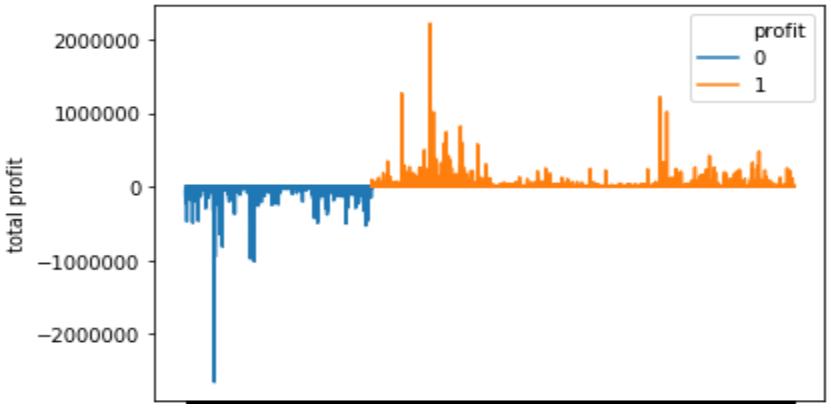
As described in the spec, projects with the same first 8-digits belong to the same project family, and for this project, we are interested in the profitability of a project family as a whole. Thus, we need to aggregate the rows in the data and sum up some variables to get the information for a project family. There are 16672 unique sub-projects and can be grouped into 10126 project families. We also found that the maximum number of sub-projects a project family has is 1438 projects and 92% of project families have less than 100 sub-projects. The distribution is shown in the below graph.



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.

Profitability Calculation and Encoding

After we aggregate the sub-projects into project families, we calculated the *profit* and *wip* 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.



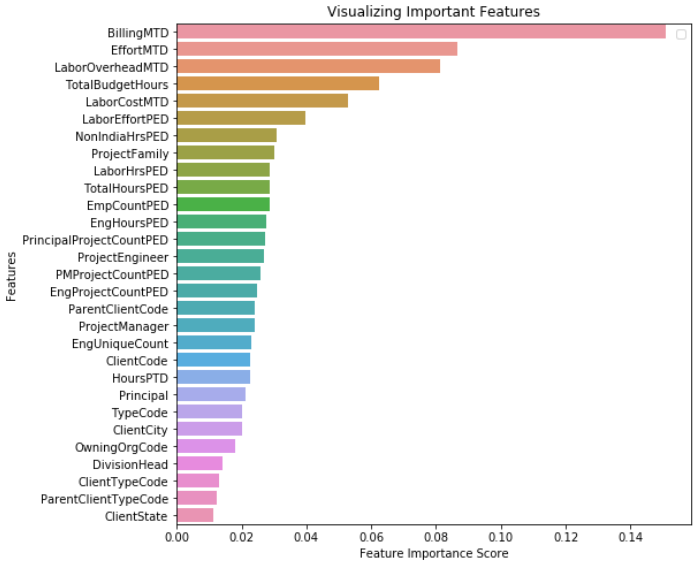
Profit and lost for all the profit families

For the actual profit, the overall profit for all profitable projects is 89,262,541, while the overall loss is 59,890,599. The median gain of profitable projects is 2,705 while the media lost for unprofitable projects is 3,060. There are some projects that are extremely profitable (2m profit) and also some projects that are huge failures (2m lose).

3. Model and Results

We used Random Forest to do the classification and the response is the binary variable for profitability while the rest being the predictors. We did a 75%-25% train-test split and set the maximum trees the algorithm grows to be 100 to save some time. The accuracy is around 0.85, which is not better than 0.9 from the black-box method, but is still pretty close.

The advantage of using Random Forest is that it can generate a variable importance diagram that has strong interpretability.



From the plot, the top 5 contributing variables are *BillingMTD, EffortMTD, LaborOverheadMTD, TotalBudgetHours*, and *LarborCostMTD*. Together they added up to 0.434, nearly half of the total importance.

4. Discussion

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, the billing, effort, labor overhead cost, total hours of budget, and the cost of labor are very influential on the final profitability, so the company needs to pay special attention to those factors next time they encounter a new project.

5. Future

In the future, we will study time series of projects such as the trends of profitability in years. We will also include labor percentage, which is another variable we calculated from the spec, to include the influence of labor in the profit. Furthermore, we plan to do clustering on profitability and study if there are any commonalities within the projects that are profitable and projects that are not, so we can better advise the company to plan wisely. Regression over the actual amount of profit is also very interesting where we can see the influence of each variable to the profit. Lastly, we plan to apply other methods that have better interpretability such as XGBoosting.