**Predictive Analysis on Default of Credit Card Clients Data**

**Xiwei Shen & Fangzhou Yuan**

**Introduction**

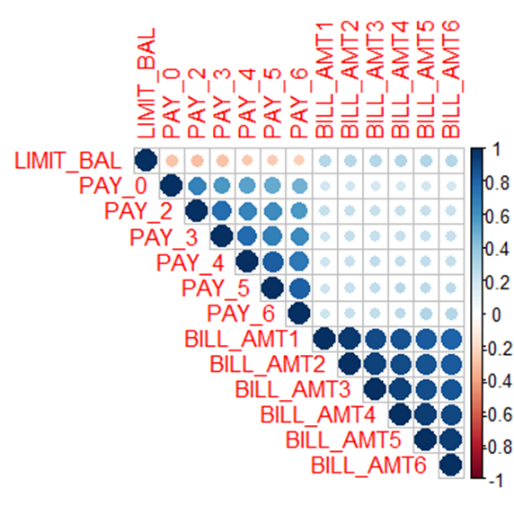
The dataset we worked on originally came from a bank in Taiwan. In recent years, the credit card issuers in Taiwan tend to increase market share by over-issuing cash and credit cards to unqualified applicants. At the same time, most cardholders, irrespective of their repayment ability, used their credit card for excessive consumption. The crisis caused the blow to cardholders’ financial credibility and it is a big challenge for the banks. This project aims at predicting the probability of default payment next month of credit card clients. Explanatory columns provided in the dataset includes payment history and amount of the past six month and other descriptive information of the clients such as marital status, age, sex, etc.

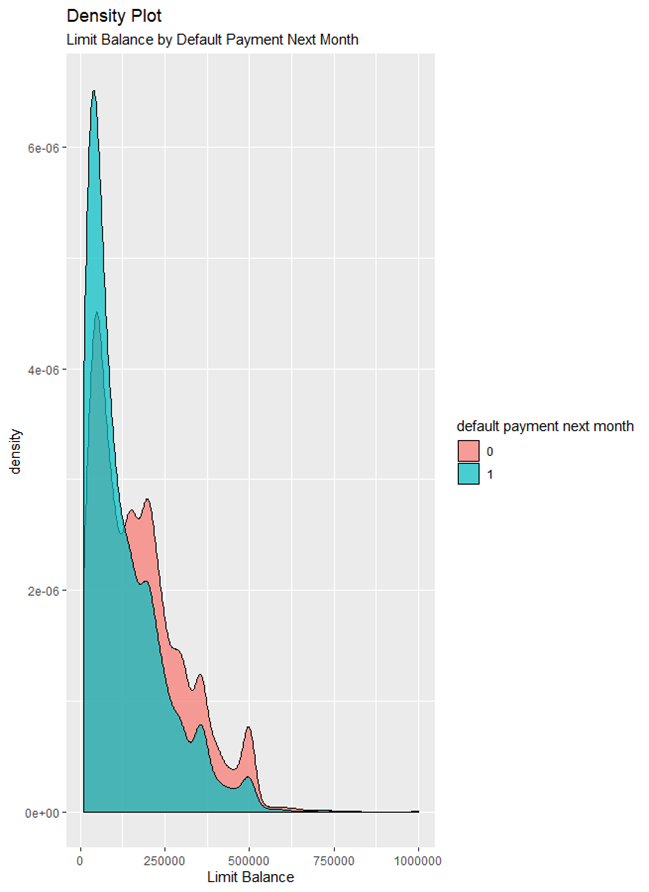
**Previous Work**

Credit card offers a flexible loan method for customers to use bank’s money for a short period of time (Merikoski, 1). By paying the credit card bills (holistically or partially as required), a customer is recognized as creditable, and such financial actions keep running. Otherwise, the credit card will be defaulted. If default payments are overlooked and not being predicted, it will result in significant financial losses to the bank on top of the damaged credit rating of the customer. Thus, making a prediction on customers’ risk of default represents significant business optimization for all banks.

Based on the data given, many statistical methods can be employed as potential solutions, such as discriminant analysis, logistic regression, Bayes classifier, and K Nearest Neighbors. In addition, advanced machine learning algorithms, Artificial Neural Networks and Artificial Intelligence algorithms were also applied to develop models for risk prediction. To accomplish the goal, we developed various supervised and unsupervised models and compared them in terms of accuracy, kappa, recall, error rate, etc. to determine which model fits the problem best.

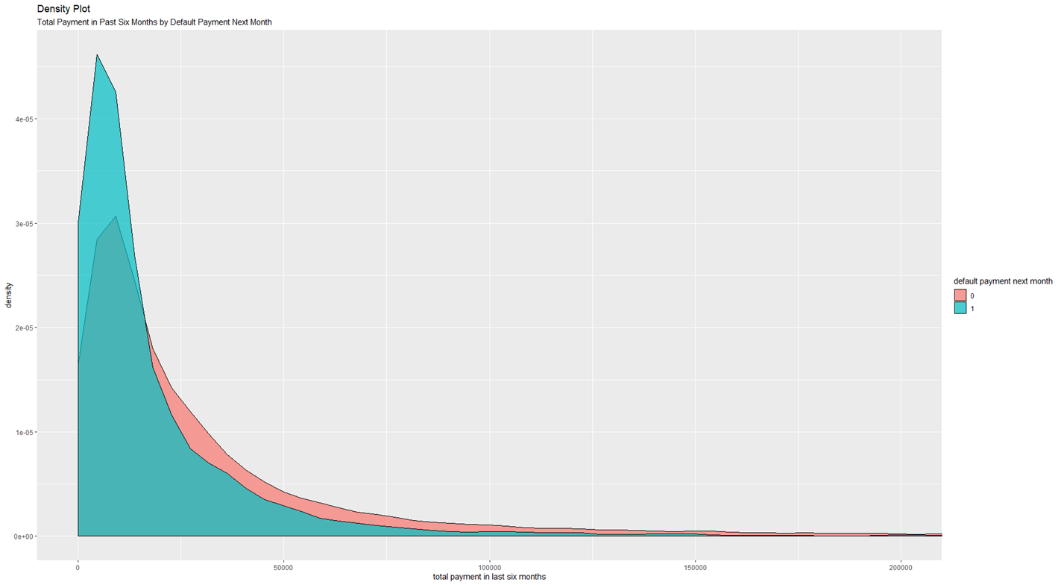
**Preliminary Data Analysis**

The data consists of 30,000 customers and 25 columns of variables. Each sample corresponds to a single credit card account. The majority of the columns are numeric values including amount of given credit, customer age, amount of monthly bills for the past six months, and the amount of payments made in the last six months. While the rest are categorical variables including customer gender, education level, marital status, repayment status in the past six months, and the target variable, default payment for next month.

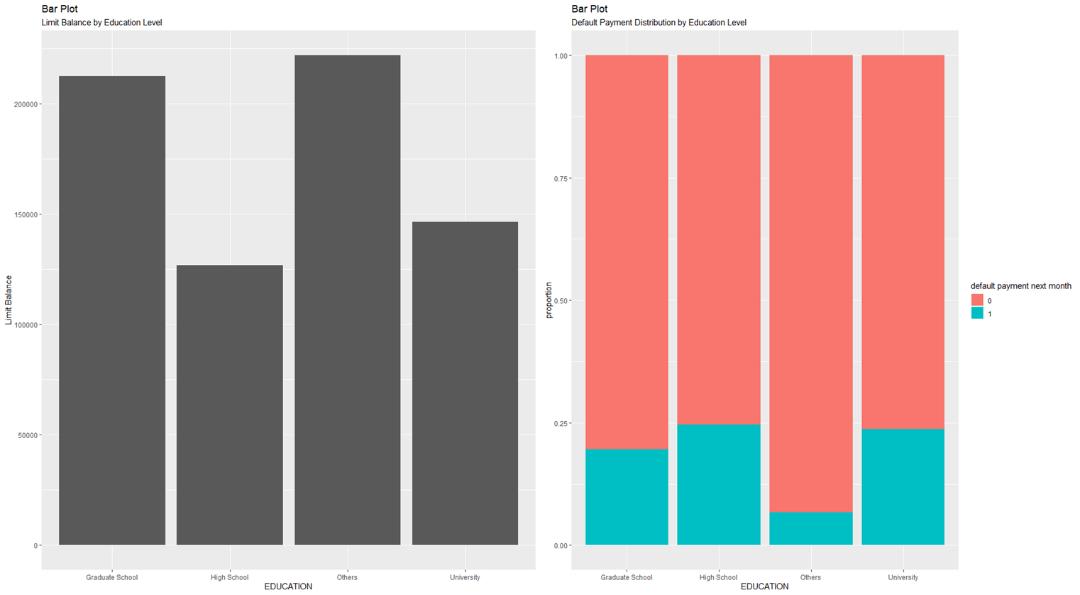
We started our analysis by making a correlation matrix, and the visualization shown on the left is an interesting part. The PAY\_\* columns include information of the repayment status of previous months, and the BILL\_AMT\* columns represents the amount of bill statement of these months. As we can see they are colored with dark blue, we can conclude that the client’s usage of their credit card is consistent on a monthly basis. People who had delayed payments further in the history leads to delayed payments in more recent history (similar to a snowball effect).

We thought a default payment next month can possibly be explained by a client’s limit balance. Because usually, in a bank credit system, the limit balance is highly correlated with one’s credit score. Thus, having a high limit balance demonstrates a client’s financial success with the payment history.

From the density plot on the left, we can tell that when the limit balance is less than 125,000, majority of the clients tend to have a default payment next month. While for clients with limit balance larger than 125,000, paying duly is a more common phenomenon. Besides personal financial status, a higher interest might count as well.

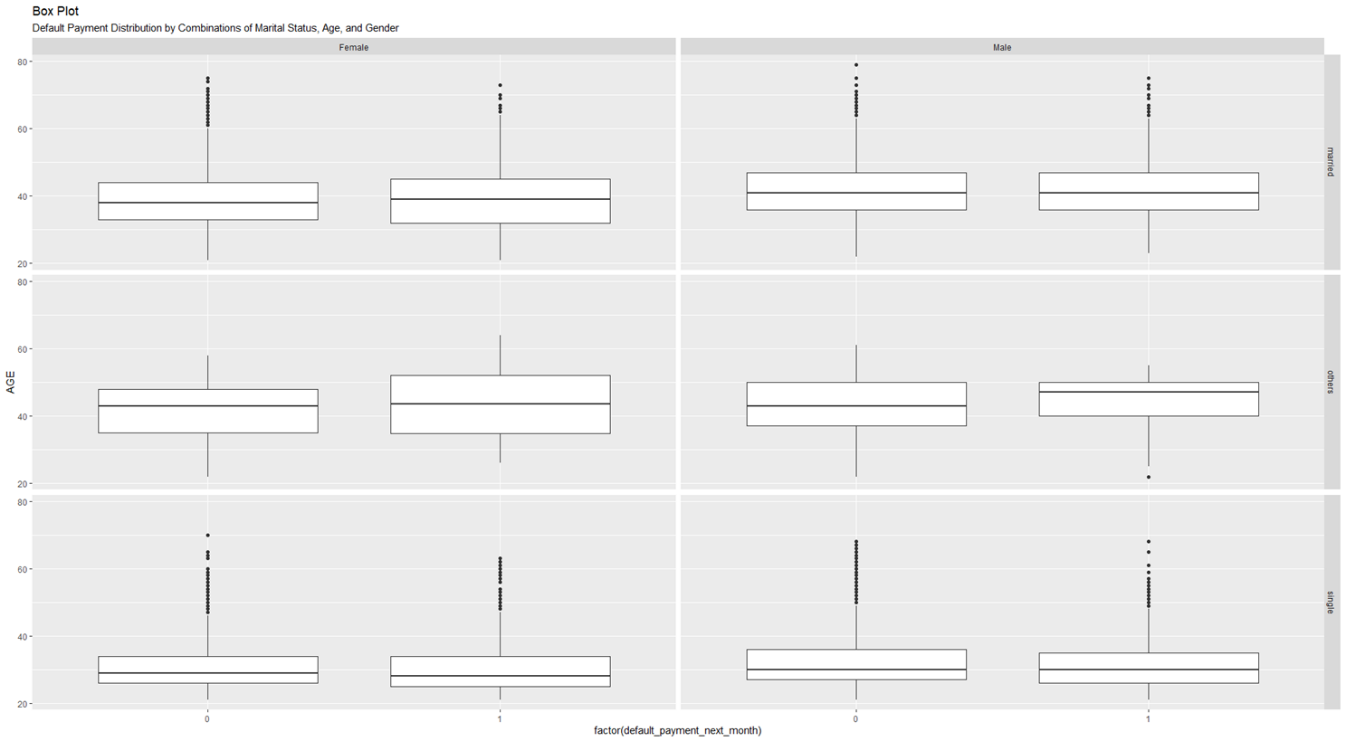


Then we would like to look at another density plot of the default payment next month distribution by the total payment amount made in the last six months. For clients with and a default payment next month, a large proportion of the population lies in the range of 0<payment<15000. For the clients pay for the credit card on time, the distribution is very similar, the major difference is that the distribution is more flattened and have a wider tails right skewed distribution, which means people with higher payment amount in the past six months tend to be reliable and will pay duly next month. The conclusion here is consistent with the one we drew from the previous plot.



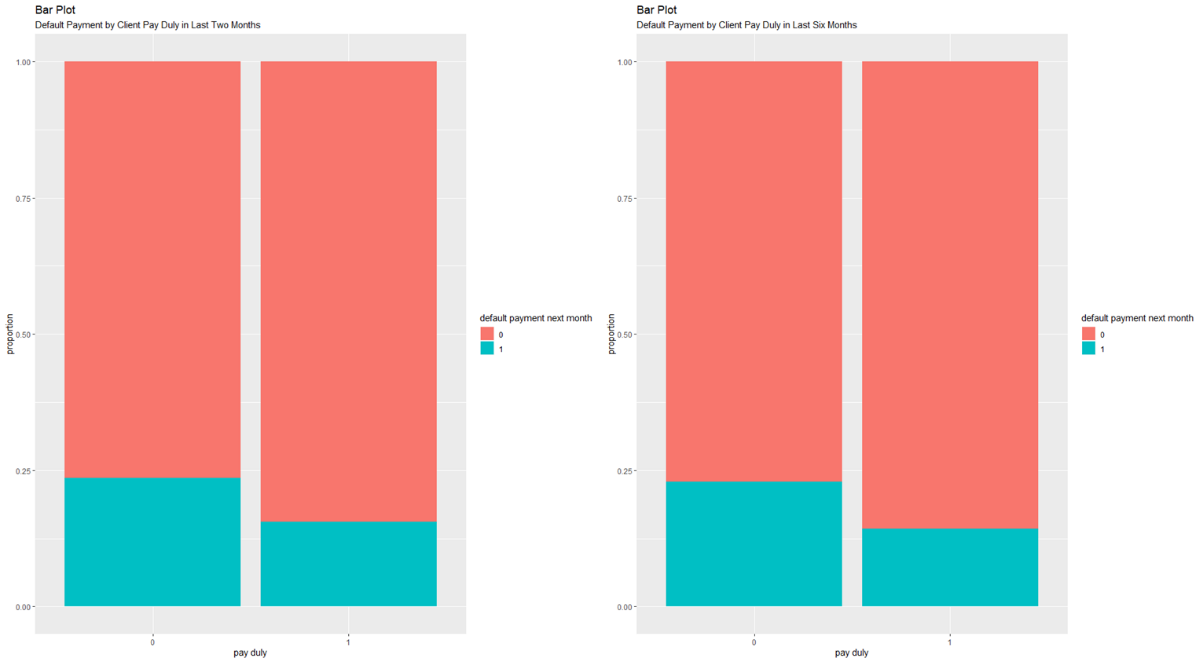
The two bar plots above demonstrate the limit balance and default payment of clients with different educational background. For balance limit, the amount given is proportional to the education level, a higher degree is associated with more limit balance. It is worth noticing that the “other” level of education, which accounts only to a very small percentage of all data points, has the highest limit balance. We do not have any further information about what the “other” level stands for.

Moving on to the default payment ratio of each education level, the plot shows almost the same trend with the limit balance plot. Higher education leads to less possibility of default payment. Again, the “other” education level has the lowest probability of default payment.



Since the client age is approximately a normal distribution center around 30 (no data points below 21), the difference among the boxplot may look trivial. The two plots for the “other” level as of marriage accounts to a very small proportion of the data, so we are not going to focus on them here.

Comparing the plots with different gender, we can see that the distribution is roughly the same no matter a client is married or not. Whereas for the plots with different marital status, the boxplots for married clients have wider range of IQR which shows that the age attribute has more variability especially for the group of clients with default payment next month.



The above bar plots show the default payment next month based on the clients’ most recent two and six months history of either they had paid for the credit card on time. From the plots we can observe that if a client has consecutively paid duly for the most recent months, one is more likely going to pay duly for the next month. The result of the plot is consistent with what we found from the correlation table.

Then, the two plots are almost identical, which tells us the default payment next month depend more on very recent history rather than the ones a lot earlier. This information gives us a hint that we can do dimension reduction when building models to accomplish better efficiency without losing considerable amount of predictive power. In addition, if we are going to keep all the payment history, we can employ models that can give weights to columns so that the payment status six months ago is weighted less significant than last month.

Through our exploratory analysis and visualization, we found some data quality issues that may further influence our model performance. For example, the “other” levels in the marriage and education variables. Other than investigating further about the dataset, another possible solution is to get rid of the rows with such information or disregard these attributes. The final decision will be made based on model performance evaluation. When evaluating the models, we can focus on the false negative besides regular specs generated from the confusion matrix. Because if a bank is over-issuing credit cards and would like to prevent cardholders from default payments, a model that less incline to return false negatives can help restrict the qualification and limit the financial loss.

**Machine Learning Methods**

We used 5 different machine learning algorithms in our final project, including Logistic Regression, Naive Bayes Classification, K Nearest Neighbors, Decision Tree, and Linear Support Vector Machine.

**Logistic Regression**

Logistic regression a statistical method for predicting binary classes, which fits our target variable, default payment next month. Logistic regression is a special case of linear regression where the dependent variable is categorical in nature. It uses a log of odds as the dependent variable and predict the probability of occurrence of a binary event using a logit function. When applying the Sigmoid function to the linear regression equation, we can have

where the polynomial equals to the dependent variable y, and xn are explanatory variables. Logistic regressions assume the independent variables have little or no multicollinearity, but in our case, the explanatory variables include monthly bills and monthly payment, which could affect the model performance.

**K Nearest Neighbors**

K nearest neighbors is a supervised machine learning algorithm that stores all available data points and classifies new data points based on a similarity measure. The key idea of this algorithm is that data points closer to each other tend to behave similarly. By using this algorithm, a data point is classified by calculating the distance between the data point and its K nearest neighbors, and then assign the data point to its nearest neighbor. The major distance functions that can be used for the KNN algorithm including Euclidean distance, Manhattan distance, Minkowski distance, and Chi-square distance. The Euclidean distance function is the most widely used distance metric in the KNN algorithm.

As a distance-based algorithm, KNN is sensitive to irrelevant attributes and outliers. It requires a proper way to deal with outliers during the data preprocessing. Another required step is to convert categorical attributes to numerical attributes if the dataset has mixed types of data. For the distance-based algorithm, we also need to normalize and standardize the data in the case where variables have different measurement scales. For KNN model parameter tuning, there is only one parameter K, which is the number of neighbors. Large K value will lead to higher bias while small K value brings the model with higher chance of overfitting. We used the grid search method in our model training process and trying to find the optimal K value within range 1 to 30. In our case, as K continues to increase till 13, the model performance is getting better and the difference during this process is relatively significant. After that, these are only small increases appear. Therefore, we choose 13 as the optimal K value. We used the consistent model performance measurement method in order to compare the performance of different models. The method we used was cross validation method with 5-fold and 3 times repeat.

**Linear SVM**

Support vector machine (SVM) is a supervised machine learning model with associated learning algorithms that can be used in both classification and regression analysis. The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space, whereas N is the number of features, to distinctly classify the data points. The hyperplane that the SVM model is trying to find is the one with the maximum margin, which is the maximum distance between data points of different classes. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Margin is only decided by the closest data points on each side of the hyperplane, and such data points usually been called support vectors. This feature makes the SVM model is robust for the outliers. Distance between data points and decision boundary indicates the confidence of the prediction. The SVM model is very suitable for two-group classification problems due to the model properties, and it works with both linearly separable and linearly non-separable cases. SVM algorithms use a set of mathematical functions that are defined as the kernel. The kernel function takes data as input and transforms it into the required form (DATAFLAIR Team). Different SVM algorithms use different types of kernel functions. The kernel function is a distance function on a higher dimension where hyperplane could be found. There are several different types of kernel functions, including Linear, Polynomial, Radial basis function, and Sigmoid.

We used linear kernel function in our case. As a distance-based algorithm, SVM also requires inputs to be numerical attributes. It is also a good idea to do normalization and standardization during the data preprocessing to prevent potential problems due to different measurement scales of data. For Support Vector Machines with Linear Kernel, the parameter we need to tune is cost C, which is the penalty associated with misclassification. This parameter makes SVM become a soft margin classification, which allows a small number of misclassifications. The model with higher costs will have lower bias but the risk of overfitting. The model with lower cost will have higher bias but lower variance. We used the grid search method in our model training process and trying to find the optimal parameter C within range 0 to 5 by every 0.1. The model performance measurement method we used was 5-fold cross validation method with 3 times repeat. Finally, we found that when C equals 2.8, the model has the best performance.

**Naive Bayes Classification**

The Naive Bayes classifier is a probabilistic machine learning model based on Bayes’ theorem with the independence assumptions between predictors. In order to understand the Naive Bayes classifier, it is better to start with Bayes’ theorem.

Using Bayes theorem, we can find the probability of y happening, given that X has occurred. Here, X is the evidence, and y is the hypothesis. The assumption made here is that the predictors are independent. In this case, the variable **y** is the class variable and the variable **X**represents the predictors. Here, is the prior probability of class, is the prior probability of predictor, is the likelihood, which is the probability of predictor given class, is the posterior probability of class given predictor. The objective of Naive Bayes classifier is to use Bayes theorem to calculate the posterior probability for each class, and the class with the highest posterior probability is the outcome of the prediction. The Naive Bayes classifier can provide both classification and probability estimation, and it's robust to outliers and irrelevant attributes. Since NBC model can take both categorical and numerical variables, there is no need to convert the data type of the predictor variables. We used “naivebayes” package for model training and testing. We tuned three parameters to explore if there is an improvement in the model performance. The first one is the parameter laplace, which is the level of laplace correction. We want to add one laplace smoothing for zero probability issues. For parameter laplace, we have tried laplace level for 1 to 5 and the model has the best performance when laplace equals 1. The next parameter is that we need to decide whether to use kernel function or not. If we choose to use kernel function, a kernel density function is used to estimate the class conditional densities of metric predictors, and it often applies to numeric attributes. Since we are doing classification and our target variable is a two levels factor, so we choose to use normal density function. The parameter adjust stands for bandwidth adjustment, and we found the best outcome exists when adjust stays at 1. The model performance measurement method we used was 5-fold cross validation method with 3 times repeat.

**Decision Tree**

Decision Tree is a supervised machine learning algorithm used for classification and regression. Decision tree model builds classification or regression models in the form of a tree structure. The model is like a tree where each node represents an attribute, each branch represents a decision(rule) and each leaf represents an outcome. The objective of the Decision Tree model is to improve dataset purity by repeatedly splitting with attributes into smaller and smaller subsets. So how does the tree split? Entropy is one of the criteria to split a node in a decision tree. It's a measurement of impurity or uncertainty.

Here, P is the probability respectively in a certain node. If a node has high entropy, it has high impurity or high uncertainty. Decision Tree model wants to reduce uncertainty. Therefore, it needs a metric to measure the reduction of this uncertainty in the target variable given additional information about it. This metric is called Information Gain. This equation represents the information gain from X on Y:

The information gain can be calculated by subtracting the entropy of Y given X from the entropy of just Y to calculate the reduction of uncertainty about Y given an additional piece of information X about Y. The greater the reduction in this uncertainty, the more information is gained about Y from X (Asutosh). Information gain is the key that is used by Decision Tree algorithms to construct its tree. An attribute with the highest Information gain will be split first. The Decision Tree model continues to split the dataset until all the data points belong to the same class, or all the records have the same attribute values, or the model reaches its pre-defined depth.

Decision Tree model can handle both numerical and categorical attributes, which greatly simplifies our data preprocessing steps. In order to better understand the tree diagram, we did convert column "PAY\_0" to "PAY\_6" from numbers to its actual meaning in words. We have tuned three parameters in our training process. The first one is Maxdepth, which is the maximum depth of any node of the final tree, the root node counted as depth 0. The second one is Minbucket, which is the minimum number of observations in any terminal node. The third parameter is Minsplit, which is the smallest number of observations in the parent node that could be split further. After tried different combination of these parameters, we found when minsplit equals 50, minbucket equals 20, maxdepth equals 5, the model has the best performance. As always, the model performance measurement method we used was 5-fold cross validation method with 3 times repeat.

**Evaluation & Conclusion**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model Performance | | | | |
|  | Accuracy | Sensitivity | Specificity | AUC |
| Logistic Regression | 0.820 | 0.953 | 0.358 | 0.773 |
| KNN | 0.819 | 0.940 | 0.395 | 0.739 |
| Linear SVM | 0.807 | 0.972 | 0.231 |  |
| Naïve Bayes | 0.795 | 0.978 | 0.158 | 0.740 |
| Decision Tree | 0.820 | 0.956 | 0.347 | 0.778 |

Putting accuracy, sensitivity, specificity and area under curve all into consideration, we conclude that the Decision Tree model fits the problem the best. While the accuracy and sensitivity of different models applied are relatively similar, the specificity varies a lot with models, which indicates that the performance of predicting the negative cases is not as good as for the positive cases. A possible factor that cause this issue might be the data imbalance. The majority of the data points belong to the class of default payment next month = 0, customers who will pay for their credit card before the due date. Thus, While the overall prediction accuracy is high. Another evidence can be found on the false positive value, i.e. customers who will default payments but predicted as pay duly. If we have more data points of default payments, we can expect the specificity to be higher.

Interestingly, we found the logistic regression, as a simple and classic statistical model with no hyperparameter tuning, performs well among all other machine learning models. We tried to reduce the multicollinearity before running the model, but the results are the same with applying the model to the original data. By inference, we can try Artificial Neural Network models and see how they perform on this dataset.

**Reference List**

Asutosh, *“How does a Decision Tree decide where to split?”*, Data Science - Machine Learning - Deep Learning, 1/5/2018, Last Accessed By 5/1/2020.

DATAFLAIR TEAM, DataFlair, “*Kernel Functions-Introduction to SVM Kernel & Examples*”, 11/16/2018, Last Accessed By 5/1/2020.

Merikoski, Max., et al. “*Introduction*”, Predicting and Preventing Credit Card Default, 5/18/2018.

**Link to Shiny App**

[https://xiwsh.shinyapps.io/Project\_Shiny](https://xiwsh.shinyapps.io/Project_Shiny/)