

Statistical & Machine Learning

Individual Project

Written By: Mui Han Ma

**Content**

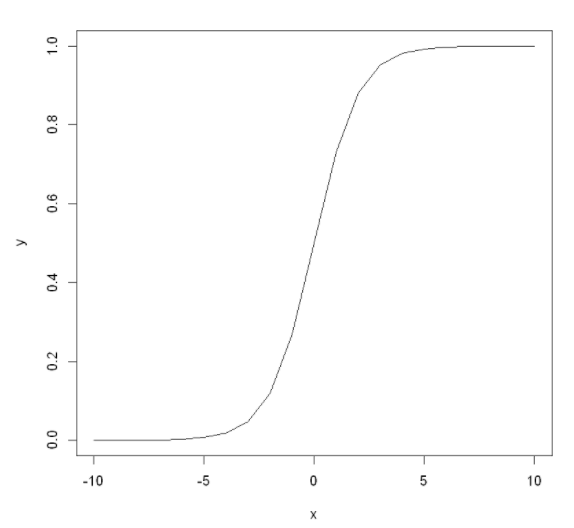
1. Background
2. Credit Card Dataset
3. Machine Learning Algorithms
   1. Logistic Regression
   2. Decision Tree
   3. Boosting Tree
   4. Random Forest
   5. K-Nearest Neighbors (KNN)
4. Experimental Setup
   1. Variable selection
   2. Cross-validation method
   3. Evaluation metric
5. Results & Conclusion

Appendix I: Reference

1. Background  
   The aim of this report is to explain and apply 5 machine learning algorithms to the credit card default data set. It is a classification case as it is to determine either the credit card customer will have default payment or not. The following 5 machine learning algorithms are selected and explained in detail: (i) logistic regression, (ii) decision tree, (iii) boosting tree, (iv) random forest and (v) K-nearest neighbors (KNN).
2. Credit Card Dataset  
   There are 20,000 observations with 25 variables (including target variable of actual default payment) from the credit card customers, except for the target variable, there are 4 types of information. The first type is the background information of the customer, like the credit given by the credit company, gender, and age. The second type is the history of status of payment from April – September 2005. The third type is the bill statement amount for the same period. And the last type is the payment amount made by the customer for the same period. The whole dataset split into train dataset (70% of the whole dataset) and test dataset (30% of the whole dataset).
3. Machine Learning Algorithms  
   The following section explains the selected 5 algorithms under the following framework:  
   (i) Mechanism of this algorithm

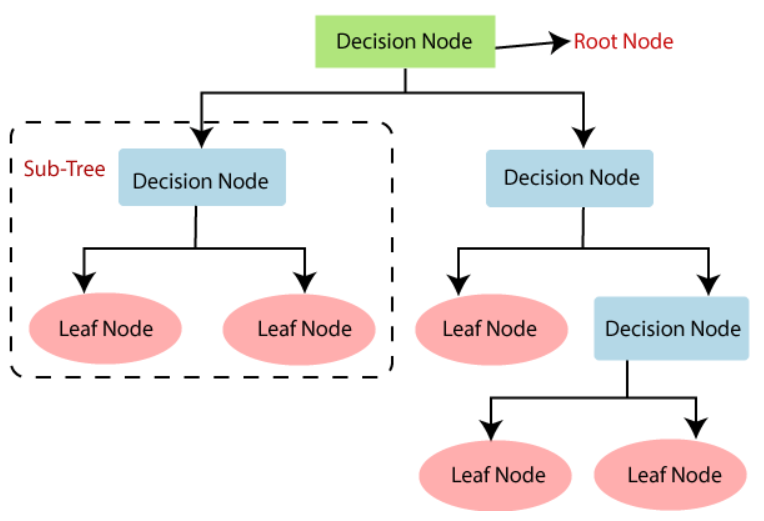
(ii) Advantages by using this algorithm  
(iii) Disadvantages by using this algorithm

* 1. Logistic Regression
     1. Mechanism of this algorithm  
        This algorithm is to predict the likelihood of this event to be occurred, like in this case, the likelihood of default payment or not in credit card. It transformed the linear regression equation by applying the sigmoid function, illustration as follows.  
          
        The linear regression equation is as follows:  
        Where y is the target variable (i.e. default payment) and , … are explanatory variables.  
          
        Sigmoid function (logistic function):  
          
          
          
          
          
        The following is the graph of a sigmoid function:

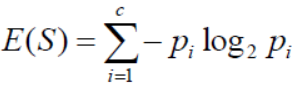
  
  
Apply sigmoid function on linear regression:

  
Sigmoid function gives a ‘S’ shaped curve that can transform the value from linear regression into a value between 0 and 1 in logistic regression. If the linear regression goes to positive infinity, y predicted in logistic regression will become 1, and if the curve in linear regression goes to negative infinity, y predicted will become 0 in logistic regression. If the output of the sigmoid function is more than 0.5, it shows a likelihood of a yes (i.e. default credit card payment), and if it is less than 0.5, it shows a likelihood of a no (i.e. no default credit card payment).  
  
In addition, there is an assumption that the target variable follows Bernoulli Distribution.  
  
There are 3 types of logistic regression, binary logistic regression, multinomial logistic regression, and ordinal logistic regression. In this case, binary logistic regression is used as there are only two possible outcomes, default payment or no default payment.

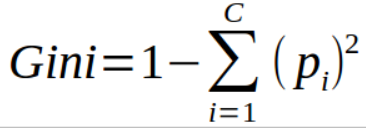
* + 1. Advantages by using this algorithm
       - Easy to train, implement and interpret
       - Gives a measure of how a predictor is relevant and the direction (i.e. positive or negative)
       - It learns fast at classifying unknown observations
    2. Disadvantages by using this algorithm
       - May lead to overfitting if the number of features is more than the number of observations
       - Cannot solve non-linear problems
       - Hard to have complex relationships
  1. Decision Tree
     1. Mechanism of this algorithm  
        This is an algorithm by referring the data features to make prediction of the target valuable. The outcome is like a tree, with the root node as the whole data set at the beginning, the decision nodes as the features and the branches as the sub-section of the tree, and the leaf node as the outcome.

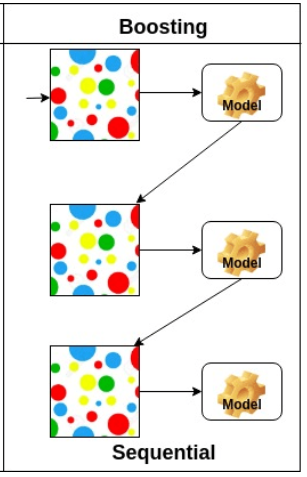
  
In order to build a decision tree, it has to determine the root of the tree (i.e. root node) first and then each of the splitting to divide the decision node to sub-trees, if it is a final output, it is the leaf node. To note that, there is also a technology called pruning for removing the branches which are unwanted.

There are two measures for the decision tree, the first one is information gain. It is to measure how much information a feature provides about a class. The following is the formula for calculating information gain:



The second one is Gini Index which is to measure the purity used when creating a decision tree. A lower Gini Index should be chosen for having a better split. The following is the formula for calculating the Gini Index:



* + 1. Advantages by using this algorithm
       - Easy to understand and interpret
       - Can be displayed by graph
       - Less effort needed for data pre-processing
       - No scaling or normalization of data needed
       - Useful in data exploration
       - Can capture non-linear relationships
    2. Disadvantages by using this algorithm
       - Unstable algorithm, small change in data may lead a large change of the structure of the tree
       - More time to train if there are many features
       - Overfitting easily
       - High variance
       - Easily affected by noise
  1. Boosting Tree
     1. Mechanism of this algorithm  
        It is a combination of two components, decision tree algorithm and boosting methods. It fits many decision trees to further improve the accuracy from the decision tree algorithm. Each of the decision trees are generated based on the previous generated tree, therefore it learns slowly from the original data set from time to time.  
        

The following are the steps for how a boosting tree works  
Step 1: Calculate the mean of the dependent label

Step 2: Calculate the residuals of each of the observations

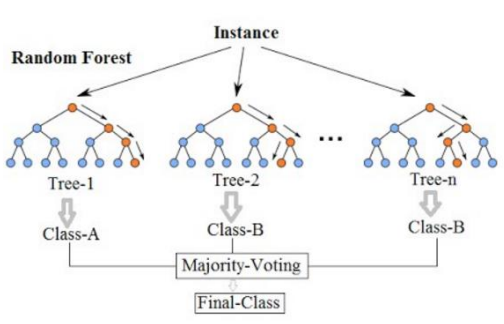
Step 3: Build a decision tree with every leaf contains the prediction of the residual

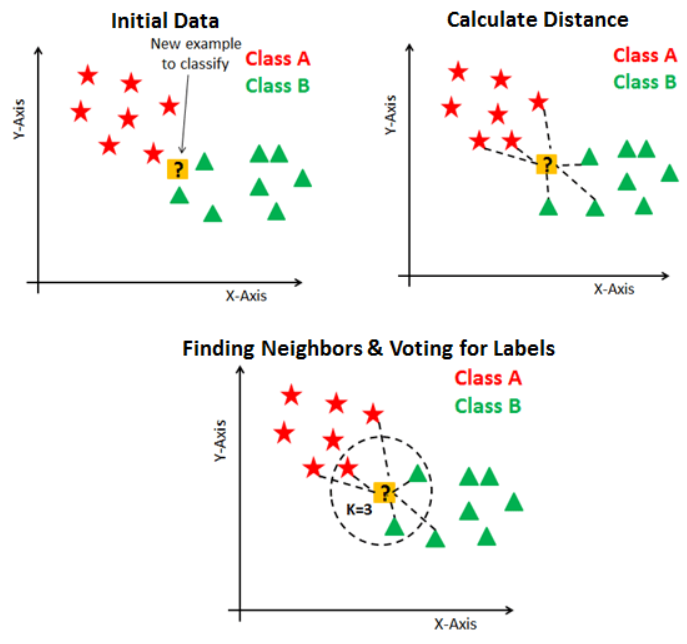
Step 4: Add the outcome (times the learning rate) with the mean of the dependent label calculated in Step 1 and have a new dependent label  


Step 5: repeat Step 2 to Step 4 until it reaches the number of iterations used in the hyperparameter

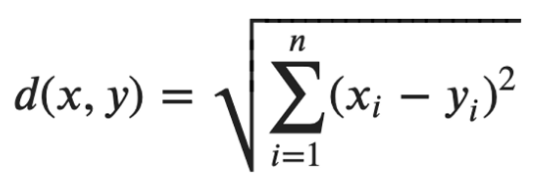
Therefore, it is a sequential tree as each of the tree relies on the information from the previous trees (times the learning rate).

* + 1. Advantages by using this algorithm
       - Can use with different types of responses
       - Random process
       - Algorithm detects the best fit solution
       - Adaptable to missing values and outliers
    2. Disadvantages by using this algorithm
       - Need at least two variables to run
       - Overemphasize outliers which leads overfitting
       - Needs lots of resource (i.e. time and computational power) to execute
       - Less interpretable compared with the previous two algorithms
  1. Random Forest
     1. Mechanism of this algorithm  
        This algorithm is a collection of decision trees, unlike boosting tree mentioned in the previous section in which boosting learns continuously in the process, for random forest, each of the decision trees generates independent prediction. When it comes to an end result, it takes the average value for the regression problem while maximum vote for the classification problem.

In order to make the random forest works, it used a bagging method to selects a random set of data points from the data set for generating each of the decision trees. And for each of the trees, it picks the features randomly based on the subsets. This random method gives the opportunity for finding the feature importance. The feature has more influences in each decision tree, the higher of the feature importance.  
  


* + 1. Advantages by using this algorithm
       - Stable
       - Adaptable to missing values and outliers
       - No scaling or normalization of data needed
       - Work well with categorical and continuous variables
    2. Disadvantages by using this algorithm
       - Needs lots of resource (i.e. time and computational power) to execute
       - Less interpretable compared with the previous two algorithms
       - Hard to interpret as it is a result of the combination of multiple decision trees, it is hard to chase the features that used for each of the trees
  1. K-Nearest Neighbors (KNN)
     1. Mechanism of this algorithm  
        The name K-Nearest Neighbors represents that there are the number of K (which is an integer) neighbors in which have the shortest distance to the label. At first, random points are selected for starting labels. And then all the observations try to find out which label has the nearest distance to them by using Euclidean distance method. After they find the nearest label to form a group, the label to take reference to is readjusted, the observations then regrouped to the nearest label, the following steps reperform until the label is stable.  
          
         

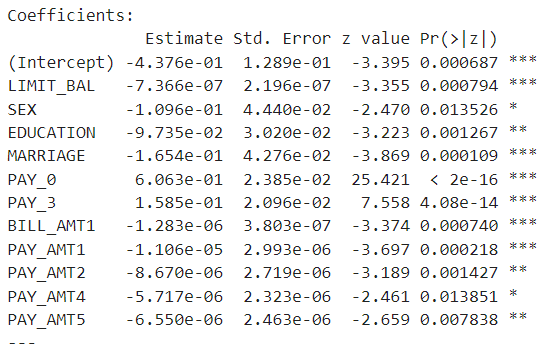
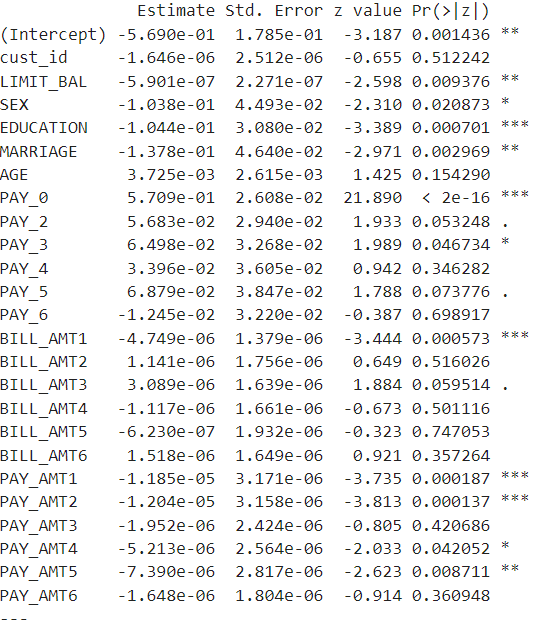
The following shows the Euclidean distance equation of how to calculate each of the observation to the label:



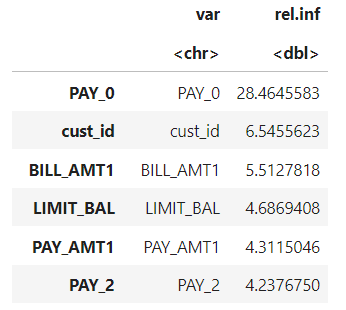
* + 1. Advantages by using this algorithm
       - Easy to understand and implement
       - Immediately adapt to new training data
       - Flexibility to choose distance metric
    2. Disadvantages by using this algorithm
       - Underfitting or overfitting if we choose the number of nearest neighbors, K, incorrectly
       - Not function well with large amount of data
       - Not function well with high dimensionality, it makes the distance calculation more complex
       - Very sensitive to missing values and outliers
       - Unstable when the K is approaching to 1 while very stable for K

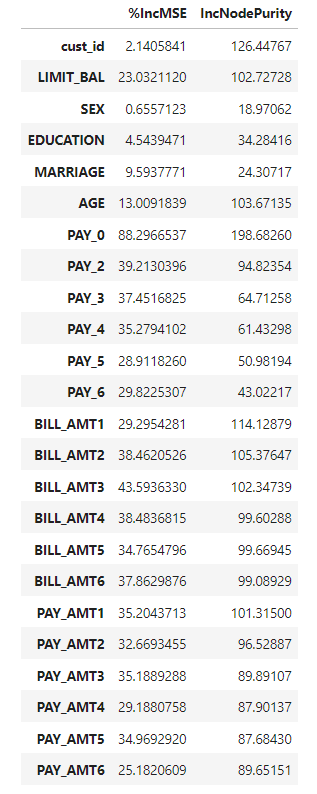
1. Experimental Setup
   1. Variable selection

In order to choose the variables to fit in the model, there are in total 3 variable selection methods, forward selection, backward selection and mixed selection. Forward selection is to select the variables one by one in order to have the lowest residual sum of squares (RSS). Backward selection is to select all the variables at the beginning and decrease those which are not statistically significant which means they have a large p-value (i.e. normally higher than 0.05), it stops until all the remaining variables have the p-value below the threshold.   
  
The following summarized variable selection method used in this study:

Logistic regression: Backward selection. All variables to fit in the model first, then remove those which are not statistically significant by considered the P-Value of each of the variables.  
  
***<Before variable selection> <After variable selection>***

Boosting tree: Backward selection. All variables to fit in the model first, then select the top 5 features with the highest relative influence. (Cust\_id is excluded as it is not one of the features)

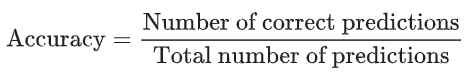
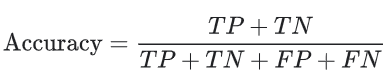


Random forest: Backward selection. All variables to fit in the model first, then select the top 5 features with the highest feature importance. (Cust\_id is excluded as it is not one of the features)  


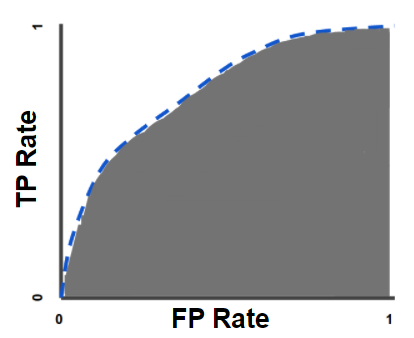
* 1. Cross-validation method  
     In order to validate the parameters used in the algorithm, cross-validation method is used. For logistic regression and KNN algorithm, 10 fold cross validation is performed.

For KNN, random forest and boosting tree, it narrows the train-test accuracy and AUC difference, which having a less over-fitting effect.

* 1. Evaluation metric  
     The following are the evaluation metrics to analyze the algorithms:  
     1. ACC – It is to measure the accuracy of the model. The evaluation result can be calculated by using the following formula:

  
  
where TP = True Positives, TN = True Negatives, FP = False Positives and FN = False Negatives.

* + 1. AUC – It is to measure the ability of a classifier to distinguish between classes (i.e. default and not default credit payment)  
       It is the area under the ROC curve.



1. Results & Conclusion

Refer to the result below, for the selected 5 algorithms. Logistic regression has the highest AUC and ACC and have very mild overfitting issue. Therefore, it is a better model to choose for predicting the default payment of the credit card customer.

The Best ACC & AUC After Performing Feature Selection & Cross-Validation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ACC\_Train | AUC\_Train | ACC\_Test | AUC\_Test |
| Logistic regression | 0.8077 | 0.7641 | 0.8062 | 0.7548 |
| Decision tree | 0.8177 | 0.6400 | 0.8157 | 0.6331 |
| Boosting tree | 0.8259 | 0.6582 | 0.8170 | 0.6418 |
| Random forest | 0.9721 | 0.9402 | 0.8057 | 0.6375 |
| KNN | 0.8077 | 0.7558 | 0.7953 | 0.7107 |

From the above results, it shows that logistic regression has the highest AUC and ACC and have very mild overfitting issue. Decision tree, boosting tree and KNN hve a little bit lower of AUC, but also have a mild overfitting issue as the evaluation metric difference between train and test is small. For random forest, it has a high accuracy in train dataset and great difference of ACC & AUC result between train & test dataset, it indicates an overfitting issue. Even though performed feature selection and cross-validation, it only narrows the difference of the evaluation metrics between train and test set, but the overfitting issue still exists.

**Appendix I: Reference**

1. Class materials from Statistical & Machine Learning
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). An Introduction to Statistical Learning: With Applications in R. Springer Publishing Company, Incorporated
3. Understanding Logistic Regression in Python Tutorial  
   <https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python>
4. Decision Tree Classification Algorithm  
   <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>
5. Gradient Boosting Decision Tree Algorithm Explained  
   https://towardsdatascience.com/machine-learning-part-18-boosting-algorithms-gradient-boosting-in-python-ef5ae6965be4
6. Understanding Random Forests Classifiers in Python Tutorial  
   https://www.datacamp.com/community/tutorials/random-forests-classifier-python
7. KNN Classification Tutorial using Scikit-learn  
   <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
8. Classification: ROC Curve and AUC  
   https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
9. Classification: Accuracy  
   https://developers.google.com/machine-learning/crash-course/classification/accuracy