NUS
National University
of Singapore

CS5228 KNOWLEDGE DISCOVERY AND DATA MINING

Submitted by: Group 9

Name	Student ID
Ding Jinhua	A0148552B
Duan Junxu	A0163154L
Wang Yuying	A0174374Y
Wu Miao	A0091649U

1. Introduction

1.1 Background

Large amount data has been generated, collected and stored our daily life, like bank transactions, online shopping, flight booking and so on. Besides recording as history for future reference, useful information can be extract from the data. It is very difficult to extract knowledge by simple exploration or query on data when dataset is very large and complicated, while it is the most common case in the real word. Data mining techniques can be used to mine knowledge from raw data stored. Together with computer become cheaper and more powerful, data mining can be applied to various area to extract knowledge using to improve efficiency, reduce cost, or provide marketing strategy and so on.

1.2 Data Mining and Marketing Strategy

Data mining is usually used to extract the patterns and useful information from large dataset. It is widely used to analyze marketing response prediction, customer behavior patterns and other useful information. For example, data mining on the data records of marketing event happened before, successful cases and failed case can be analyzed separately or together to figure out reasons for success and the causes for failure, then for the next similar event, organizers can make use the knowledge to mined to predict marketing response, decide should the this event focus more on the customers with highly possibility to purchase the product to avoid the effort waste or should they pay more attention and avoid the causes to failure last time and persuade more customers to buy the product. Useful information can be mined from the data to provide more efficient, effective and smart strategies for marketing.

1.3 Research Question

In this project, we are mining knowledge from a dataset which is list of information of customers that participated in a marketing campaign of Portuguese Banking Institution. The campaign is about subscribing a term deposit product. Knowledge mined from the dataset can be used as a guide for future campaign to select the customers that is more likely to buy the product. Customers finally will be classified into two class “yes” (subscribe) and “no” (not subscribe), so it is a binary classification.

The flow of this report follows the data mining process flow. First, a data preprocessing has been done to understand data format and prepare the data for the following steps; then, a machine learning model has been trained and used to predict on the test dataset which also have been preprocessed by the same procedure as train dataset; next, an evaluation will be done on the predict result; finally, using the result to extract useful knowledge to provide guides for future marketing activities.

This is an academic group project aiming to guide students learn data mining process. A Kaggle competition has been organized to encourage students to get best model and most accurate prediction ensure the knowledge mined is correct and useful for future marketing. To get the best model, procedures mentioned above has been done iteration by iteration by trying different techniques and different combinations. This will lead student to research and learn more techniques. A powerful tool, R, is used in this project to facilitate data processing and machine learning process.

2. Data Preprocessing

In this section, the structure of this data and several techniques used to preprocess these raw data will be introduced.

2.1 Data Exploration

The data is available on the *Kaggle* website. There are two different data sets available, input variable and output variable. The input variables consist of “bank client data” and “other data”. The "bank client data” has 12 attributes, including *id, age, job, marital, education, default, housing, loan, contact, month, day* of week and duration. The "other data” has 9 attributes, including *campaign, pdays, previous, poutcome, employment variation rate, consumer price index, euribor 3-month rate* and *number of employees*.

Only 1 output variable, *y*, indicates whether the clients has subscribed a term deposit. The whole data is divided into training data and testing data. The size of training dataset is 30891 rows and the size of testing dataset is 10297 rows.

In the initial data observation, we try to distinguish different data types and find out what they represented respectively. Therefore, data is divided into the following four main categories, binary value, categorical value, numeric values and null value (unknown).

2.2 Data Cleaning

However, there are still some obvious problems for the whole dataset. The first one is that there are many records with missing values in the training dataset which are labeled "unknown". The missing data consists of about 26% of the data. The second one is that only 12% of the training data shows the output variable of "y", and it means that the data is imbalanced. The third one is that there are some outliers existing in the dataset, and we should minimize their effects on prediction by appropriate data cleaning measures. The last one is that the data dimension is too high, so feature selection is necessary to avoid overfitting.

2.2.1 Unknown Values

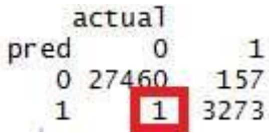
In the part, we try to handle missing values by some common analysis method. The first one is that omitting these records with missing values. The second one is that replacing “unknown” with the most frequent values or mean values. The last one is that using R function, *centralImputation(),* to assign a statistical-support value to each “unknown”.

2.2.2 Outliers

In the part, we use 2 main methods to define outlier. At first, according to the convenient definition of an outlier, outlier are some points which falls more than 1.5 times the *interquartile range* above the *third quartile* or below the *first quartile* [5], we calculate the outliers for each numeric data and remove them. The second method is known as Pauta criterion or 3σ criterion. If a data value is in (μ-3σ, μ+3σ), the value should be considered as an abnormal value [3].

2.3 Imbalanced Dataset

The training data used in this project has 30891 entries which are labeled in two classes (y: yes, no). Number of entries labeled with “yes” is 3430, while number of entries with label “no” is 2746. The ratio between “yes” and “no” is about 1:8, so this is imbalanced dataset. Using the whole training dataset will cause the machining learning model biased towards “no” class. From table below, we can see the compare to FP, FN is quite small due to the imbalance property of the dataset.



```
      actual
pred      0     1
   0 27460   157
   1     1  3273
```

Figure 1. Confusion matrix using whole training dataset

During the project we have tried *over-sampling* and *under-sampling* techniques to balance the training dataset to get a better result.

2.3.1 Over-sampling

Over-sampling is a technique that generate more entries for minority class (“yes”) to balance the dataset. For over-sampling, we have tried ubSMOTE (x, y, …) function in “unbalance” library, which implements “Synthetic Minority Over-Sampling Technique” (SMOTE). SMOTE generate new entry based on nearest neighbors’ data. The result of using over-sampling is not good compared to under-sampling, this may due to most data attributes in the dataset is not numeric type.

2.3.2 Under-sampling

Under-sampling is a technique that remove entries of the majority class (“no”) from the training data set to balance number of classes used for training the machine learning. The entries have been removed is selected randomly. A uniformly balanced dataset, which contains same number of “yes” and “no” may not be the best dataset for training. By tune the ratio of “yes” over “no”, we found that for most of the models, ratio 1:3 will contribute best prediction result.

2.4 Feature Selection

In the part, we do feature selection based on two theories. Firstly, many classification methods perform better and predict more accurately if highly correlated attributes are removed [6]. Therefore, we use the R packages, Caret, to generate a correlation matrix and remove attributes with an absolute correlation of 0.75 or higher. The result is 19, 16 and 14. It means that it's suggested to remove *euribor* 3 *month rate, employment variation rate* and *previous* respectively.Confusion matrix are attached below.

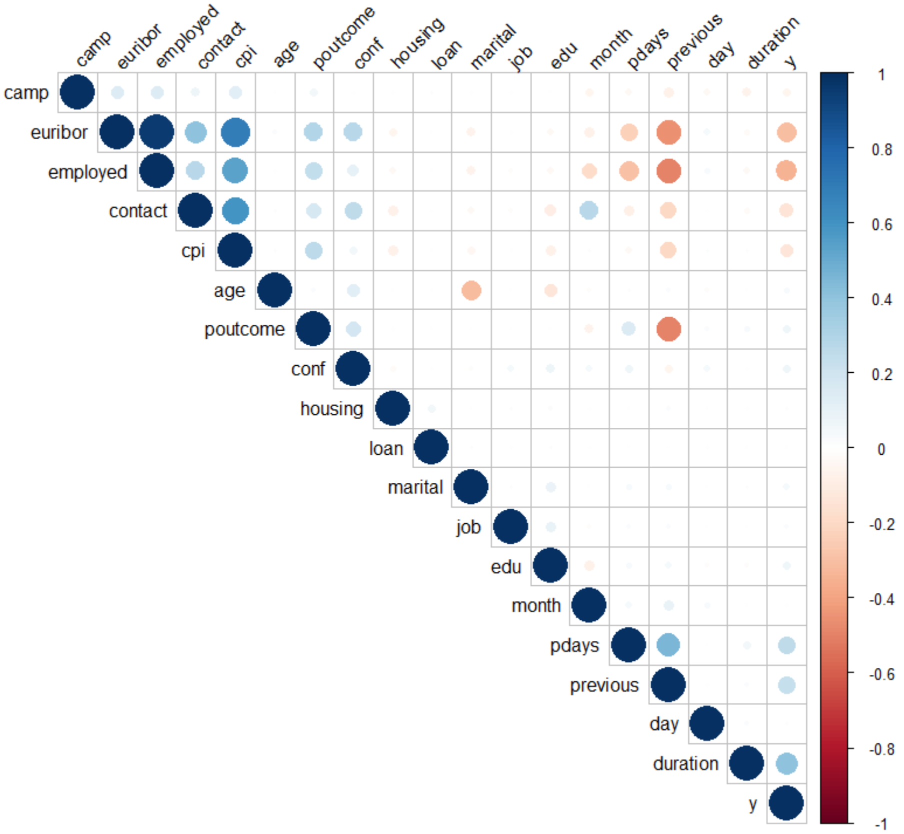
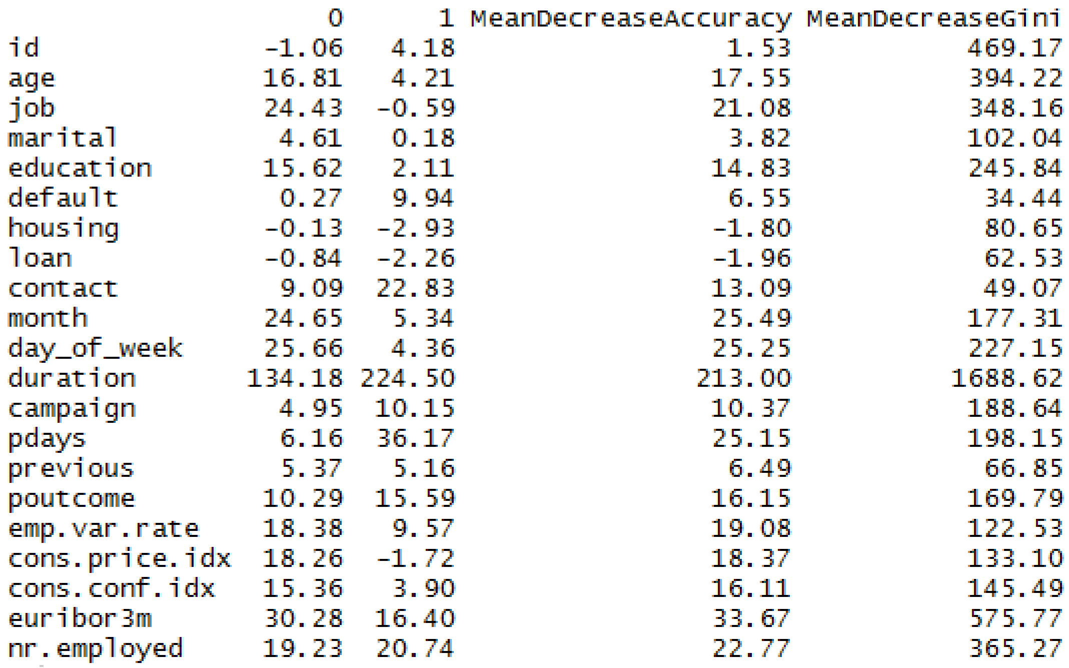


Figure 2. Correlation Matrix

The second method is that using the "importance" function of the "randomforest" package to identify the importance rank for attributes. The function will generate 2 variables, *MeanDecreaseAccuracy*, *MeanDecreaseGini*. Even if the two variables both represent the importance of attributes but from different perspectives, we only select *MeanDecreaseGini* as the main benchmark. The higher the *MeanDecreaseGini* is, the more effect the attribute has on the classification prediction. The result has been shown below, the most important attribute is duration with value of 1688.62 and the least important attribute is default with value of 34.44.



```
                      0      1 MeanDecreaseAccuracy MeanDecreaseGini
id                -1.06   4.18                 1.53           469.17
age               16.81   4.21                17.55           394.22
job               24.43  -0.59                21.08           348.16
marital            4.61   0.18                 3.82           102.04
education         15.62   2.11                14.83           245.84
default            0.27   9.94                 6.55            34.44
housing           -0.13  -2.93                -1.80            80.65
loan              -0.84  -2.26                -1.96            62.53
contact            9.09  22.83                13.09            49.07
month             24.65   5.34                25.49           177.31
day_of_week       25.66   4.36                25.25           227.15
duration         134.18 224.50               213.00          1688.62
campaign           4.95  10.15                10.37           188.64
pdays              6.16  36.17                25.15           198.15
previous           5.37   5.16                 6.49            66.85
poutcome          10.29  15.59                16.15           169.79
emp.var.rate      18.38   9.57                19.08           122.53
cons.price.idx    18.26  -1.72                18.37           133.10
cons.conf.idx     15.36   3.90                16.11           145.49
euribor3m         30.28  16.40                33.67           575.77
nr.employed       19.23  20.74                22.77           365.27
```

Figure 3. Calculating Importance of Each Attribute

3. Classification Methodology and Experiments

3.1 General Settings

Hold out validation method was implemented in this project to evaluate the performance of different models.

3.2 Basic Classification Algorithms

3.1.1 Logistic regression algorithm

Logistic Regression, is a generalized linear classification model. It is easy to implement and efficient to train.The results generated by LR tend to have a low variance and a high bias.Parameters tuning is not required for this algorithm.
In this project, we adjusted the threshold p to get the highest performance model.
Best Model:
Threshold: if Pred>0.211, class=”yes”,else ,class=”no”
Metthews Correlation Coefficient(MCC): 0.5582
Confusion Matrix

	True Positive	True Negative
Predicted Positive	864	755
Predicted Negative	346	8332

It is used as the base model for this binary classification problem here to have a rough idea of models' performance.

3.1.2 Decision tree algorithm(Regular)

R package party was used to build training model for regular decision tree algorithm. It is easy to understand and interpret. This advantage become more valuable when we need to summarize business insight from the model. However, it more prone to overfitting compared to other models.
Parameter Tuning:
The only flexible to tune this model is to adjust feature selection. The table below gives a summary of result for different feature selections.

Features	MCC
job,month,marital,age,var,contact,duration,pdays,poutcome,conf,euribor,employed	0.509
job,month,marital,contact,duration,pdays,conf,euribor	0.547
marital,contact,duration,pdays,conf,euribor	0.560

From the table above, we could conclude that decision tree tends to overfit the training data if there are too many features available. Proper selection of features will improve the test accuracy by controlling overfitting.
Best Model:
MCC:*0.560*
Confusion Matrix

	True Positive	True Negative
Predicted Positive	738	472
Predicted Negative	467	8620

3.1.3 Neural Network

A neural network model simulates how neutron transit signal in the biology. It is composed by input nodes, output nodes and hidden layer nodes. Multiple layers of nodes enable neural network to model complex relationships between input and output beside linearity. It is very easy to use. Sometimes actual relationship is simple, but using neural network to make a complex model which may cause overfitting and generate a bad result for test dataset.

Best Model:
threshold: if Pred>0.4, class="yes",else ,class="no"
MCC: 0.5846
Confusion Matrix

	True Positive	True Negative

Predicted Positive	1022	1055
Predicted Negative	188	8032

3.1.4 Others

Other models such as Support Vector Machine, Naïve Bayes algorithm was also implemented in this project. However, their result is much worse than the logistic regression model mentioned before. Therefore, their performance is not further discussed in this report.

3.2 Models Based on Ensemble Learning

3.2.1 XGBoost

XGBoost was invented by Tianqi Chen which helped him win the Higgs Machine Learning Challenge. It follows the principle of gradient boosting and uses a more regularized model formalization to control overfitting, which gives it better performance.

Moreover, it improves data structures for better utilization of process which makes training much faster than other models.

Parameters Tuning:

Different features combination was used to tune the models; however, the performance does not improve much.

Best Model:

Subsampling: yes entries vs. no entries =1:2

Hyper Parameters:max_depth=15, eta=0.12,nrounds=500,early_stopping_rounds=100

Threshold:pred <- ifelse(pred>0.5062,1,0),

MCC:0.624

Confusion Matrix

	True Positive	True Negative
Predicted Positive	1054	802
Predicted Negative	228	8140

3.2.2 Random Forest

From name we can know, “Random Forest” is machine learning model based on decision tree model. Decision tree model easily become overfitting when number of features or number of entries in the dataset is large, but random forest overcome this problem. Each decision tree in the forest is built on a subset of features and data entries [2]. During the prediction, each decision tree will do the classification (“yes” “no”) for the entry passed in, and then generate the final output based on most frequent classification result from all the decision trees in the model.

Best Model:

threshold: if Pred>0.177, class=”yes”,else ,class=”no”
mcc: 0.6154
Subsampling: yes entries vs. no entries =1:3
Confusion Matrix

	True Positive	True Negative
Predicted Positive	982	802
Predicted Negative	228	8285

3.2.3 ADABoost

ADABoost is machine learning meta-algorithm. It can be used in conjunction with many other learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. Even if ADABoost is sensitive to noise and outliers, in some problems it can be less susceptible to overfitting than other learning algorithms. The individual learners can be weak, but if the performance of each learner is better than random guess even slightly, the final model can become a strong learner with the help of ADABoost.

Best Model:
Subsampling: yes entries: no entries=1:3
Hyper Parameters: type = "gentle", iter = 100, cp = -1, maxdepth = 15,maxcompete = 1,xval = 0
threshold: if Pred>0.55, class=”yes”,else ,class=”no”
MCC 0.6259
Confusion Matrix

	True Positive	True Negative
Predicted Positive	973	736
Predicted Negative	237	8351

3.2.4 CART

Classification and regression trees (CART) are a non-parametric decision tree learning technique that produces classification or regression trees, depending on the type of the dependent variable, categorical or numeric.

Decision trees are formed by a collection of rules based on variables in the data set. Rules are selected to get the best split to differentiate observations based on the dependent variable. Once a rule is selected and splits a node into two, the process repeats on each child node. Splitting stops when CART cannot detect any new gain, or some pre-set stopping rules are met.

Best Model:

Subsampling: yes entries vs. no entries =1:3
Hyper Parameters: minsplit=20,minbucket=20,maxdepth=10,xval=5,cp=0.0005,
method="class", parms = list(prior = c(0.75,0.25), split = "gini"
threshold: if Pred>0.61, class=”yes”,else ,class=”no”
mcc: 0.6051
Confusion Matrix

	True Positive	True Negative
Predicted Positive	921	699
Predicted Negative	289	8388

3.3 Further Ensemble

3.3.1 Voting Ensemble

Voting Ensemble is the easiest way to ensemble predictions from different models. This strategy’s performance generally improves when numbers of models increases or the correlations between different models decreases.
The different voting strategies that used in this project are listed in the table below:

Best Model 1:
Model Combination: 2*ADAboost, 1*Xgboost1, 1*Xgboost2, 1&CART1, 1*CART2, 1*randomForest
MCC: 0.6394
Confusion Matrix

	True Positive	True Negative
Predicted Positive	999	211
Predicted Negative	741	8346

Best Model 2:
Model Combination: 2*ADAboost, 2*Xgboost, 1*CART1, 1*CART2, 1*randomForest
MCC: 0.6404
Confusion Matrix

	True Positive	True Negative
Predicted Positive	996	214
Predicted Negative	730	8351

3.3.2 Stacking Ensemble

Stacked generalization was first introduced by Wolpert. This method uses a pool of base classifier and combines their predictions to reduce the generalization error.

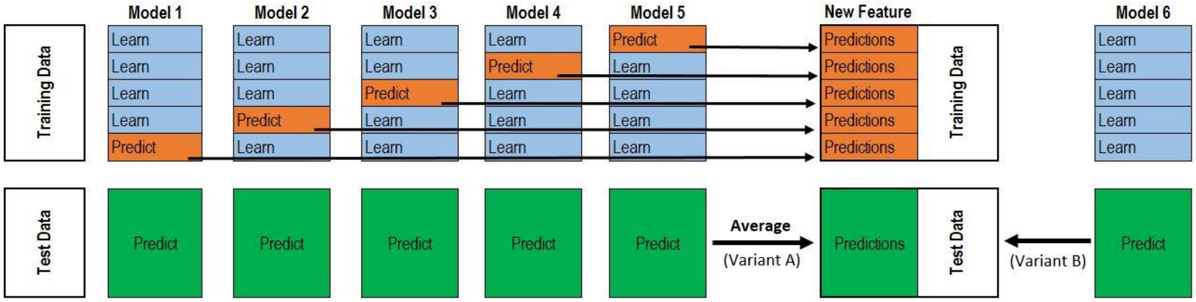


Figure 4. Flowchart of 2 layer 5-folds stacking

In this project, we implemented a 2 layer 5-folds stacking.In the first layer, three best available models from previous experiments, ADAboost, Xgboost and Random forest are used to generate the training data and testing data for second layer. Logistic Regression was used to train the second layer and threshold are carefully chose to get the best performance model.

Best Model:

Threshold:pred <-ifelse(pred>0.1165,1,0)

Subsampling:None

MCC: 0.621
Confusion Matrix

	True Positive	True Negative
Predicted Positive	978	767
Predicted Negative	232	8320

4. Evaluation and Discussion

4.1 Evaluation Methods

Accuracy

Classification Accuracy is the metric to measure the performance of a classification algorithm to test its capability to classify a entry into its respective group, which is binary classification in our case. It signifies the proportion of data entries of which the eventual class prediction proves to be correct.

Precision

Precision can be considered as a measure of the correctness of the positive prediction, which is the percentage of instances labelled as positive that are positive. Precision in other words is the fraction or percentage of predicted instances that are relevant by the classification algorithm.

Recall

Recall can be conceived as a measure of ability of the opportunity-catching, which is the percentage of positive instances that are actually labelled as positive. Recall in other words is the measurement of the capability of classification algorithm to actually turn those potential positive to become predicted positive instances.

F1 Score

F1 Score measures the classification accuracy by integrating both precision and recall as described above to compute a new score, it takes both precision and recall into consideration and output as an average of the precision and recall.

ROC

ROC, Receiver Operating Characteristic, curve is designed to measure discrimination, i.e. the ability of the model to correctly classify whether an instance would be predicted into the correct class. Compared with accuracy, it also considers the cost/benefit through the relationship of TPR/FPR and powers the decision makings in the classification models evaluation.

Ten-fold Cross-validation

The ten-fold cross-validation method is not a performance metric. It is a method of validation for the developed model whereby the train dataset is partitioned into 10 equal parts, nine separate parts of the dataset are then selected for training the model, while the one fold left is used for validating the developed model. This experiment is repeated 10 times to validate the proposed models and to test the consistency of the modelling results. In our case, we used five-fold cross-validation for better efficiency in the experiments.

4.2 Analysis of Model Results

In this session, we list the performance metrics including the MCC, test accuracy, TPR, FPR, specificity, F1 score and ROC for all models.

With results obtained in previous sessions, we observed that the train data with balanced of “yes” and “no” provides better classification results. Due to the very imbalanced distribution in the bank data, accuracy is not a good metric in measuring the performance in this project. With the various modelling results presented in the table below, ADABoost and Xgboost were found to have high recall rate and best MCC results in the single modelling. Ensemble modelling could further improve the MCC and lower the FPR to give better overall score in kaggle rankings, both ensemble modelling can improve the precision significantly. However, it’s observed that the recall was lower for ensemble voting although the MCC was higher than the rest.

Table 1: Comparison of the performance metrics with different models

	TPR/Recall	FPR	Accuracy	MCC	F1 Score	Precision
Random Forest	0.812	0.088	0.900	0.615	0.656	0.550
Logistic Regression	0.714	0.089	0.893	0.558	0.611	0.534
CART	0.761	0.077	0.904	0.605	0.651	0.569
Neural Network	0.845	0.116	0.879	0.585	0.622	0.492
ADABoost	0.804	0.081	0.906	0.626	0.667	0.569
Xgboost	0.871	0.104	0.893	0.624	0.657	0.527
Ensemble_voting 1	0.574	0.025	0.908	0.639	0.677	0.826
Ensemble_voting 2	0.577	0.025	0.908	0.640	0.679	0.823
Ensemble_stacking	0.808	0.084	0.903	0.621	0.662	0.561

The main goal of this project is to get the highest MCC from different models to achieve the best classification prediction results, as suggested in the Kaggle competition. From the table listed above, we can see ADABoost performs best and scored highest in the MCC results in the single modelling. The ROC curves computed in the figure below also presented that the AUC for ADABoost is highest among the experimented six models.

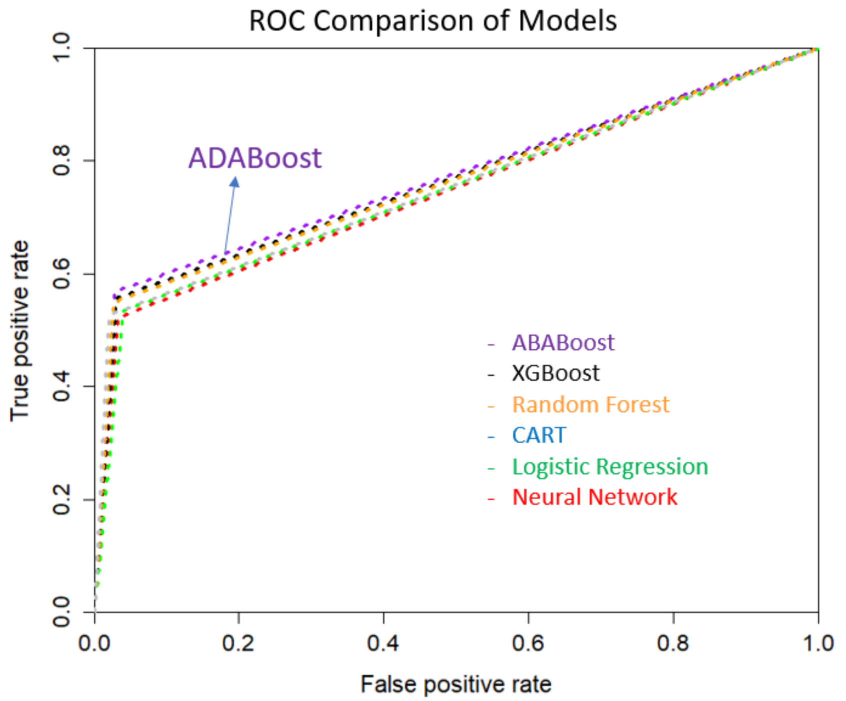


Figure 5. ROC Comparison of the six models discussed above, with latching the best MCC point

With the use of voting and stacking, we exploited further potential of the classification methods by combining and integrating different models to get better prediction results as listed in the table above.

4.3 Comparison & Discussion of Models

4.3.1 Model Comparison

Besides the standard model evaluation metrics, we believe it's important to explore further to determine how to choose the best practical implementation of the classification models. Thus, we also take a deep dive into the results of the models to investigate how it performs when the TPR is very high, or close to 1, by checking the FPR value [8]. By doing so, we could better understand the behavior and performance these models in the practical user environment.

Ultimately, the model built in this project could be used for the bank to predict and identify those customers who are potentially subscribing the term deposit, while trying to keep a minimal impact on those customers who are not interested which is to lower FPR with high TPR. With this strategy, the model could help to reduce the workload and costs by focusing on the fine-tuned targeted group who would eventually subscribe to the deposit. In this case, we highlight a measurement of the power of a model the false positive rate at true positive close to 0.95 (i.e. 95% of the customers who indeed have interests are covered). This metric is to compare those models which can identify almost all the potential customers by how much they mistakenly classify people who will not subscribe a term deposit into class "yes".

Table 1: Comparison of added metrics of models

	Random Forest	Logistic Regression	CART	Neural Network	Xgboost	ADABoost
Accuracy	0.900	0.893	0.904	0.879	**0.906***	0.893
MCC_*max*	0.615	0.558	0.605	0.585	0.624	**0.626***
FPR at TPR≅0.95	0.985	0.2709	0.2419	0.6254	**0.1797***	0.9982

With our targeted metric of the FPR with TPR ≅0.95, we can see some models resulted in very high FPR, sometimes even higher than TPR, when about 95% of the actual positive instances are covered. Xgboost model, on the other hand, dominates the performance in this measurement with a significant advantage in minimizing the cost while maximizing the benefit.

4.3.2 Limitation and Discussion in the Modelling

In the exploration of classification modelling, we also learnt some areas where could be further improved and researched.

- Several techniques to preprocess the data has been tried, including outliers eliminating, continuous number categorization, missing value handling, data sampling (like sample high 0-score entries) and so on, but later we found that adoption of those many those techniques did not improve our prediction result. In the final code, we only implemented the feature selection and focused on the modelling manipulation to improve the prediction results. Better data preprocessing techniques to further improve the results could be exploited.

- Due to the lack of experience in data mining and machine learning, our team spent large amount of time on surveying machine learning algorithms to find the best-fitting model for the project dataset. In the first half of this project, we used the fundamental algorithms that has been taught in the lectures, but results are not as good as expected. Later, we started to use more advanced algorithms, such as ADAboost, Xgboost which gave the best prediction. Due to result focused and time limit, some algorithms hadn't not been researched in depth to its full capability.
- The Stacking algorithm we introduced in 3.3.2 is an algorithm that has large potential but not fully utilized. This algorithm using prediction result of various machine learning models as input data to construct a secondary model, and much better result could be expected compared to base single modelling. More in-depth investigation for better prediction could be conducted.

5. Conclusion

5.1 Summary of the Project

From this project, we have learnt several techniques related to data mining and machine learning, such as data preprocessing techniques and machine learning models. Various classification algorithms including the traditional models like Neural Network, Random Forest, CART and Logistic Regression etc. and the state-of-art machine learning models like XGBoost and ADAboost were investigated and implemented throughout the project, the strength and weakness were studied and analyzed to understand the best fitting solution under the real-world environment. Through these problems finding and solving journey, we have researched, applied and learnt a lot of data mining and machine learning techniques, which will be very helpful in our future study and work. With the utilization of the advanced data mining technique and machine learning algorithms, it helps to reduce the cost while improve the benefits and efficiencies in today's organizations, such as the underlying bank data discussed in this project.

5.2 Recommendations for Bank

According to the data from this case, data of the term deposit status with "yes" label consists of 1/9 of the training data, which indicates that the success rate of making a deal of term deposit with customers by the marketing campaign is not high. Therefore, based on the analysis above, we try to provide some suggestions for the bank marketing campaign.

Firstly, bank should construct an efficient and complete customer information system. For example, job, marital status, education degree, financial status, previous and loan status have different influences on the result of marketing campaign. Bank could build the related relational database including these data with different weights calculated to classify and find out the potential buyers to increase the marketing success rate.

Secondly, nowadays, telemarketing is very common in bank industry, insurance industry and financial industry. However, most people really do not like telemarketing. From the “importance” analysis above, it’s discovered that these data, duration, contact, day and month, have relatively high MeanDecreaseGini, and should be taken consideration into a marketing campaign implementation. In our opinions, bank should train its staff to increase their telemarketing skills. For example, staff can control the duration as well as find the best call time in order not to make customers bored or unhappy. Likewise, we also believe that if the bank can regard customers with high degree, no loan and no default as target customers, it’s possible to increase the amount of successful transaction and net profits.

5.3 Recommendations for Future Studies

Meanwhile, this is a popular project of data mining, and it has very high research value for the future researchers of machine learning or data mining, so we also have some suggestions for the future research,

Firstly, there is a very important conclusion. Data with feature selection and without feature selection both has certain abilities to predict the result accurately. The key is to select the most suitable model with the most compatible feature selection method. Appropriate feature selection could simplify the model structure and reduce data dimension to increase the run time and the model accuracy.

Secondly, different models may have different advantages on prediction. For example, comparing Random Forest with ADAboost, randomForest has higher recall (less FN) (0.812>0.804) in this case, and ADAboost has higher precision (less FP) (0.57>0.55). It means that the two models may have low correlation and have the value of combination. Therefore, voting based on the combination of different models with the optimized weights and threshold could smooth the whole prediction process and get a better prediction result, and the method is also the key of our solution.

References:

[1] Mlwave.com. (2017). Kaggle Ensembling Guide | MLWave. [online] Available at: https://mlwave.com/kaggle-ensembling-guide/ [Accessed 12 Nov. 2017].
[2] Random Forest, [online] Available at: https://en.wikipedia.org/wiki/Random_forest
[3] Zhang, J., Pechenizkiy, M., Pei, Y., & Efremova, J. (2016, April). A robust density-based clustering algorithm for multi-manifold structure. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 832-838). ACM.
[4] Chen.T. (2017). dmlc/xgboost. [online] Available at: https://github.com/dmlc/xgboost [Accessed 12 Nov. 2017].
[5] Wikipedia. (2017) Outlier [online] Available at: https://en.wikipedia.org/wiki/Outlier [Accessed 12 Nov. 2017].
[6] Brownlee. (2014). feature-selection-with-the-caret-r-package. [online] Available at: https://machinelearningmastery.com/feature-selection-with-the-caret-r-package/ [Accessed 12 Nov. 2017].
[7] Analytics Vidhya. (2017). How to build Ensemble Models in machine learning? (with code in R). [online] Available at:https://www.analyticsvidhya.com/blog/2017/02/introduction-to-ensembling-along-with-implementation-in-r/ [Accessed 12 Nov. 2017].
[8] Chen.J (2014). Who Will Subscribe A Term Deposit?[Online] Available at: http://www.columbia.edu/~jc4133/ADA-Project.pdf