Classification on Bank Marketing Data Set to predict client's intention of a term subscription

Shuting Liao, Bingmiao Zhang, Ying Xie

April 13, 2017

Contribution:

- Shuting Liao: clean data, generating data, data reconstruction, conclusion, report revise.
- Bingmiao Zhang: clean data, data description, SVM coding, classification method report.
- Ying Xie: generating data, ensemble method coding, SVM report, plots coding.

Abstract

If a bank want to gain benefit, they should have a right taste of their clients' intention, in order to achieve this, they need to do plenty of research on the behavior of their clients. In this project, we are going to predict if the client will subscribe a term deposit or not by using different classification methods: random forest, Support Vector Machine, and Decision Tree are been mainly used. In particularly, this data is not so common since it contains many categorical variables which sets restriction on choosing methods. Besides, imbalance exists in this data. Under these unsatisfying circumstances, we try SMOTE algorithm to reconstruct the data. And then Ensemble Method is mainly applied. In addtion, we use two measurements, error rate and Area Under Curve, for each methods and choose the best method by comparing their evaluation.

Keywords: Classification, Area Under Curve, Imbalance, Ensemble Method, SMOTE.

1 Data Exploration

1.1 Data Description

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. There are 45211 examples and 17 inputs, ordered by date (from May 2008 to November 2010).

Variable description:

- age: clients' age (numeric)
- job : type of job (categorical: 'administration','blue-collar','entrepreneur','housemaid','management',' employed','services','student','technician','unemployed','unknown')
- marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
- education: education level (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','profess
- default: has credit in default? (categorical: 'no','yes','unknown')
- housing: has housing loan or not (categorical: 'no','yes','unknown')
- loan: has personal loan or not (categorical: 'no','yes','unknown')
- contact: contact communication type (categorical: 'cellular','telephone')
- month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
- day of week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

- duration: last contact duration, in seconds (numeric)
- campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
- pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
- previous: number of contacts performed before this campaign and for this client (numeric)
- poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

1.2 Statistics Summary

Table 1: Summary statistics of quantitative variables

Statistics	N	Min	Median	Mean	Max
age	45211	18	39	40.94	95
balance	45211	-8019	448	1362	102127
day	45211	1	16	15.81	31
duration	45211	0	180	258.2	4918
campaign	45211	1	2	2.764	63
pday	45211	-1	-1	40.2	871
previous	45211	0	0	0.5426	275

Table 1 provides a statistics summary of all continuous variables which can indicate that variables are on different scales, and the spreads are quiet large.

1.3 Visualization

Figure 1 gives a more visualized result, variable 'balance', 'duration', 'campaign', 'pdays', and 'previous' appear to be strongly right skew, which can infer the means are much greater than the median, this is the case because skewed-right data have a few large values that drive the mean upward but do not affect where the exact middle of the data is (that is, the median). Both Figure 1 and Figure 2 provide that there is no strong correlation within each variables.

Two plots below present a more detailed distribution between the predictor parameters and response y. There is a sign shows that it may exist imbalanced data, i.e. there are more 'no' than 'yes' in the response (y) variable.

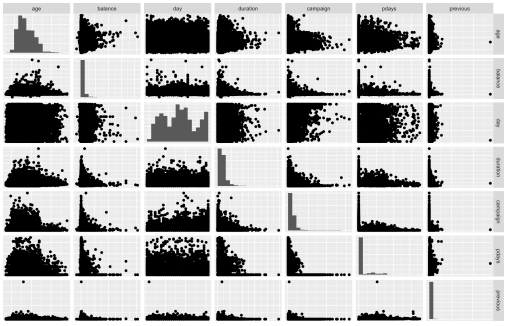

Figure 1: Pairwise screeplot for quantitative variables

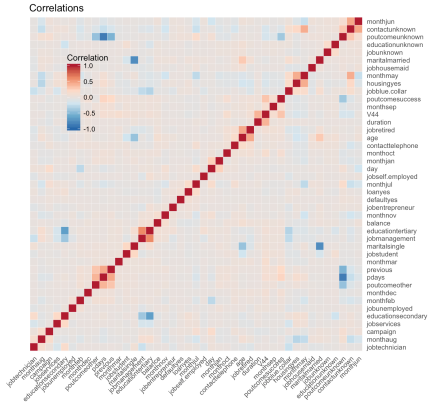

Figure 2: Correlation heatmap for whole data set

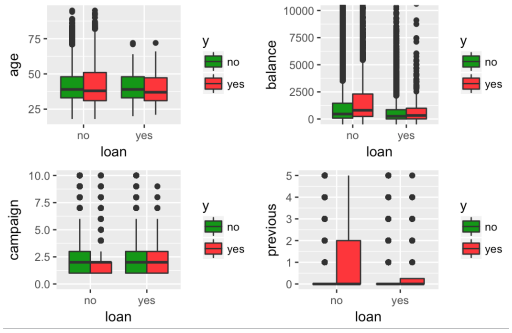

Figure 3: Boxplots for quantitative variables divided into response

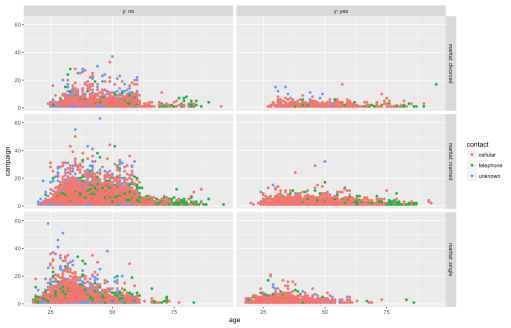

Figure 4: Scatter plot based on campaign vs. age by response and marital

2 Data Reconstruction

2.1 Imbalance in data

Since our data is about the direct marketing campaigns for a bank and it is obviously significant to make a right prediction on clients' intention if they are willing to subscribe a term deposit. It is okay if we think a client who is not interested in term deposit as the one is interested. However, if a client who intends to subscribe a term deposit is thought to have no interest, it may contribute to a big profit loss for a bank institution. Thus, the accuracy for prediction on class 'Yes' is more important than 'No'.

Unfortunately, according to the statistics summary, there seems to be an issue of imbalance in our data with 39922 cases in 'No' and 5289 cases in 'Yes'. A dataset is imbalanced if the classification categories are not approximately equally represented. Imbalance in data, especially in class, always contributes to worse prediction. This depressing performance appears when we fit the original data with random forest shows that the error rate for class 'Yes' is over 0.8.

2.2 SMOTE Method

SMOTE algorithm, an approach to the construction of classifiers from imbalanced datasets can help to solve this problem. The general idea of this method is to artificially generate new examples of the minority class using the nearest neighbors of these cases. Furthermore, the majority class examples are also under-sampled, leading to a more balanced dataset. By over-sampling the minority (abnormal) class and under-sampling the majority (normal) class can achieve better classifier performance (in ROC space) than only under-sampling the majority class.

Therefore, we will use this method to reconstruct our training set and try our models in the Section 4.

3 Performance Measures

In the field of classification, there are many statistics measurements. What we usually used is the error rate or overall accuracy, while doing the research, there is also a method called the Area Under Curve (AUC) of ROC. For extremely imbalanced data, the overall accuracy is often not a decent measurement of performance. A classifier that predicts every case as the majority class can still achieve high overall accuracy. Therefore, besides only check the overall accuracy, we are also going to use AUC as measurement method to evaluate our model and pay attention to the error rate for minority class (class 'Yes' in this dataset).

3.1 AUC

A confusion matrix in Table 2 (for a 2 class problem) is known to measure the performance of classifier algorithms. TN denotes the number of negative cases correctly classified (i.e. True Negatives), FP which means False Positive, is the number of negative examples incorrectly classified as positive, besides FN is the number of positive examples incorrectly classified as negative (False Negatives) and TP is the number of positive examples correctly classified (True Positives). It is easy to find that $ErrorRate = (FP+FN)/(TP+FP+TN+FN)$.

Table 2: confusion matrix

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

For a balabnced data, error rate is helpful to evaluate the performance. But it is more reasonable to use ROC curve to handle imbalanced data. ROC curves represents the family of best decision boundaries for relative costs of TP and FP. On an ROC curve the X-axis represents $FP = FP/(TN + FP)$ and the Y-axis represents $TP = TP/(TP + FN)$. Thus, the steeply farther the curve is from the line $y = x$, the better the performance is. Equavalently, larger Area Under the ROC Curve (AUC) represents better performance since essentially AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative example.

4 Procedure

Based on our major goal, we will focus on classification to predict that if a client would like to subscribe a term deposit. Therefore, we would try several classifiers and compare the results to choose the best one given our data. Given our data is a little special owning to many categorical variables, the main methods we would like to focus are Random Forest, Ensemble Method, SVM and Logistic Regression.

4.1 Generating test and training

Firstly, we notice that some categorical variables have multiple levels. Thus, when perfoming some method like svm and logistic regression which need to deal with numeric variables, we first transfrom the data matrix into a sparse data matrix in which all categorical variables are binary. But for other classifier such as random forest and tree, we choose to use the original data instead of the sparse matrix.

Second, since our data is a whole set, first we need to devide the data into training and test data. We randomly choose 20% from the data set as our test set while the rest is training set. Meanwhile, we keep the ratio of 2 classes the same in both training and test sets. Then we apply SMOTE method to imbalanced training set and get a new training data to avoid the imbalance. In the following steps, we will use balanced data set to predict imbalanced test set so that we can choose the best classifier for our data.

4.2 Restrictions

4.2.1 MDS/PCA

Generally, it is always a good choice to perform PCA and MDS for dimensionality reduction and data visualization. However, after exploring our data, we can find that the majority variables are categorical, especially when we transform the original data into sparse data matrix with 36 categorical variables and only 7 continuous variables. Besides, Given our major goal is prediction, it is not helpful to do PCA and MDS.

4.2.2 LDA/QDA

LDA and QDA should not be applied to this data. As we have claimed before, most variables of our data are categorical. It is known that performing LDA and QDA is under the assumption that assuming that the conditional probability density functions $p(\vec{x}|y = 0)$ and $p(\vec{x}|y = 1)$ are both normally distributed with mean and covariance parameters. Unfortunately, categorical variables do not satify this assumption and therefore, we have to disregard thses two methods.

4.3 Classification Method

4.3.1 Random Forest

We applied random forest method on the training data with only contains binary variable, and get the result as following:

Table 3: Random forest result on binary data

Error rate			AUC
Total	Yes	No	0.8527
0.1499	0.1438	0.1508	

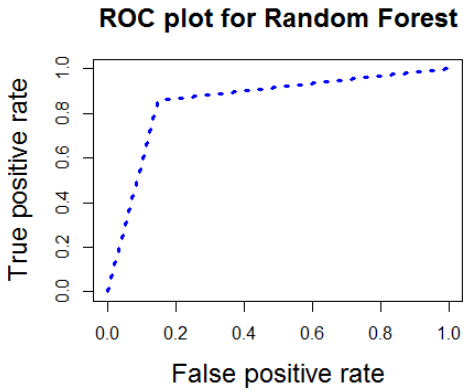

Figure 5: Roc plot for Random Forest

The result in Table 3 and Figure 5 show that both statistics measurements performed well with almost 86% of the classify accuracy. For ROC curve plot, the true positive rates are plotted against false positive rates. As we know, the closer the elbow point of ROC for a model comes to 1, the better it is. Here the ROC elbow point is near 0.86, so we can indicate that the model using random forest method is a good method, but not an excellent one.

4.3.2 Tree

Main idea: The goal for decision tree on classification case is to verify that each box should contain nodes mostly from the same class. To achieve this goal, first is to split the feature space into parts at each step, then repeat it for each branch, i.e. recursively partition the feature space into boxes, the final result will be a tree of splits, a partitioning of the variable space into boxes and assignment of a class label to each box.

The result in Table 4 and Figure 6 show the performance of decision tree method for predicting model. Here, the error rate for total and 'No' are quite small, but for predicting 'Yes', error rate is 0.20, which is larger than the error rate for 'Yes' in random forest method, this also drives a lower the AUC.

Table 4: Tree result on binary data

Error rate			AUC
Total	Yes	No	0.8159
0.1714	0.2006	0.1676	

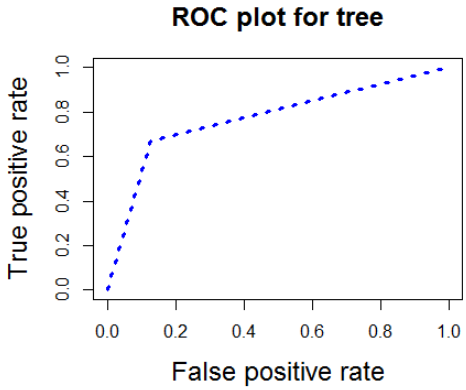

Figure 6: Roc curve for decision tree method

Tree size (total number of splits) is a measure of model complexity. If the size is too large, it will cause over-fitting, if the size is too small, it may miss important structure and/or not classify well. Most algorithms grow a large tree, then prune it. Figure 7 shows how the C_p value changes will effect the X-val Relative error and the size of tree. To get a relatively low error rate and a small size of trees at the same time, we choose C_p to be 0.003. Figure 8 present a tree diagram with $C_p = 0.003$. And using our tree model, we can get the boundary plot in Figure 9.

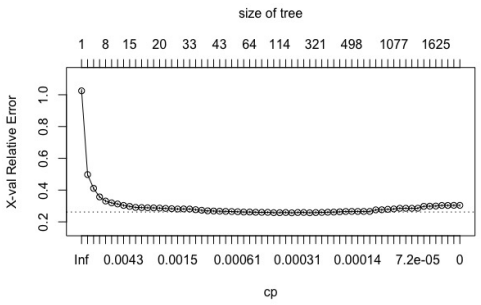

Figure 7: error rate vs. cp

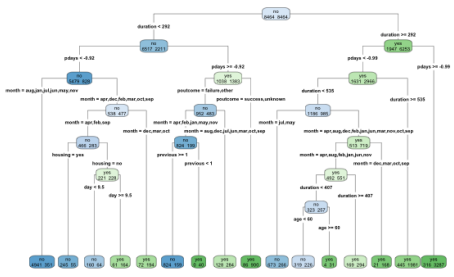

Figure 8: Tree diagram

4.3.3 Boosting

Main idea: First, given some sample points weights (initially all equal) that measure how much the point affects the classifier, then fit a classifier and adjust weights to give more weight to the points that were misclassified. Next repeat the last step M times. Finally, take

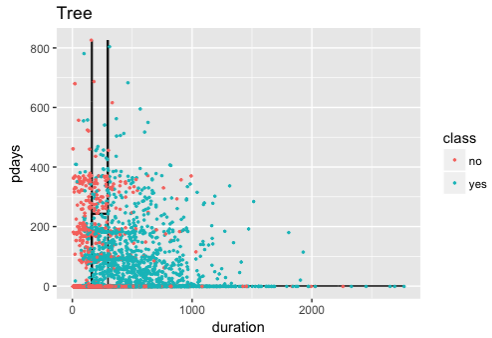

Figure 9: Tree Boundary

a weighted vote of the resulting M classifiers, with more weight given to better classifiers. In our project, we are going to use a R package called Adaboost.

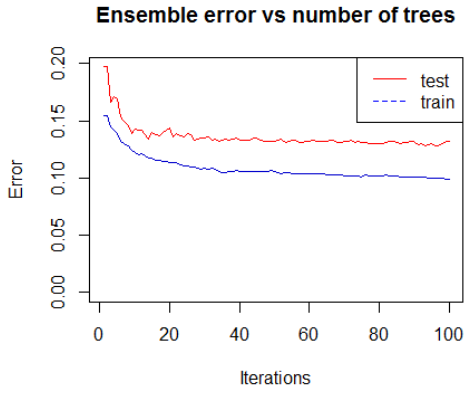

Figure 10: Shows the error evolution of Adabag Boosting method

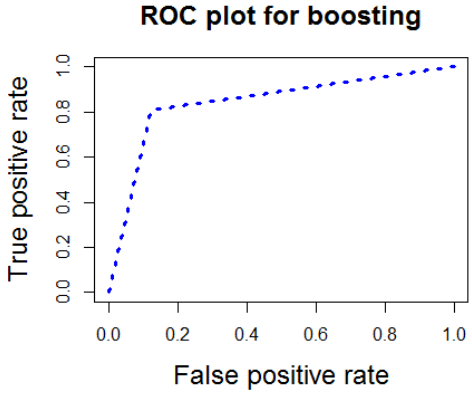

Figure 11: ROC curve for Boosting

Figure 9 is calculating the error evolution of an AdaBoost.M1 for a data frame as the ensemble size grows, according to the plot, we can decide the ensemble size for the best boosting model. As we can indicate that both test error and train error drop at size ≈ 10, and become much more stable after size $= 40$. After choosing the size, we get the result in Figure 10 and Table 5. The AUC value is similar to random forest with smaller error rate, overall better than decision tree.

4.3.4 Bagging

Main idea: Bagging or bootstrap aggregation averages a given procedure over many samples, to reduce its variance. We need to find sample with replacement from the training data $(x_1, y_1), ..., (x_n, y_n)$ to obtain a bootstrap sample $(x_1^*, y_1^*), ..., (x_n^*, y_n^*)$. Then construct a new

Table 5: Boosting result on binary data

Error rate			AUC
Total	Yes	No	0.8513
0.1271	0.1769	0.1205	

tree T_1. Next, repeat everything B times, obtaining B trees $T_1, ..., T_B$. Finally, given a new point x, classify by majority vote among $T_1(x), ..., T_B(x)$.

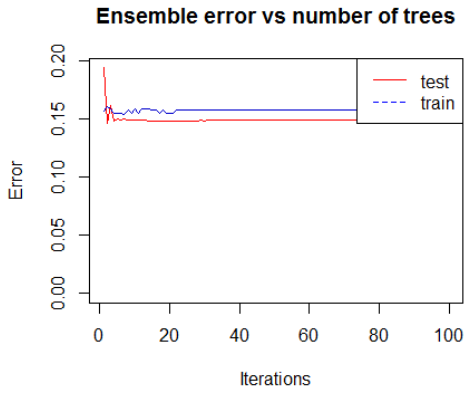

Figure 12: Shows the error evolution of Bagging method

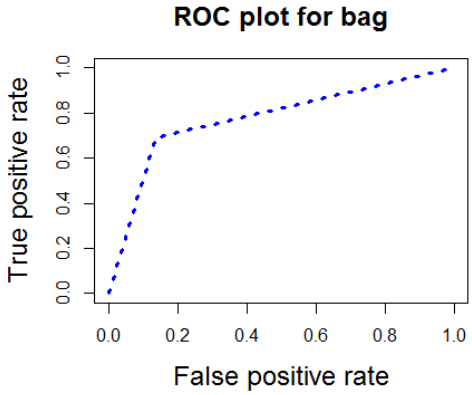

Figure 13: ROC curve for Bagging

Table 6: Bagging result on binary data

Error rate			AUC
Total	Yes	No	0.7834
0.1441	0.3113	0.1220	

Similarly, Figure 11 is calculating the error evolution of an bagging classifier for a data frame as the ensemble size grows, according to the plot, we can decide the ensemble size for the best boosting model. As we can indicate that test error has a quick drop at size ≈ 5, and stay almost the same error rate after size ≈ 7. After choosing the size, we get the result in Figure 12 and Table 6. The AUC value is fall below 0.80, which may cause by the large error rate for 'Yes' with value of 0.3113. Based on this result, Bagging may not be a good choice for prediction on this particular data set.

4.3.5 SVM

For SVM with different choices of kernels (linear, polynomial, Gaussian kernels), the training, testing errors and AUC are shown in Figure 14, Figure 15, Figure 16 and Figure 17. From the plots, we can find that all training error, testing error and AUC are sensitive to the choice of kernels.

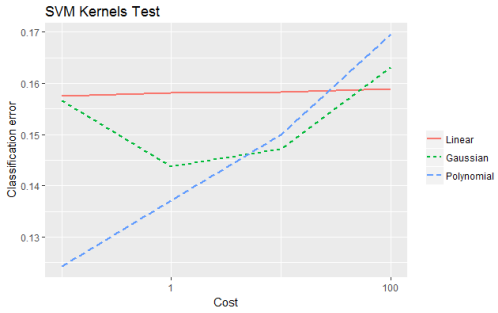

Figure 14: Test error for SVM

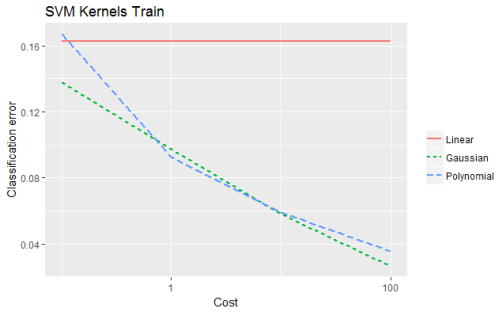

Figure 15: Training error for SVM

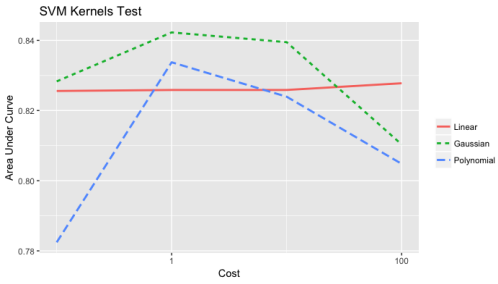

Figure 16: AUC for SVM test

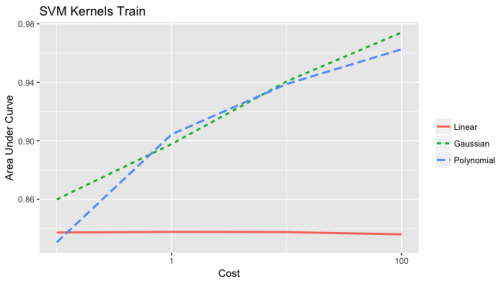

Figure 17: AUC for SVM train

Table 7: SVM result on binary data

Error rate			AUC
Total	Yes	No	0.8255
0.1505	0.1807	0.1446	

From the plots, we can see that for SVM with linear kernel, training error, testing error and AUC are all not sensitive to the choice of cost (the tuning parameter controlling the total amount of slack allowed). For SVM with polynomial and Gaussian kernel, training error, testing error and AUC are sensitive to the choice of cost. According to Figure 15, among SVM with different kernels, SVM with Gaussian kernel has a slightly better result in terms of AUC. Therefore, the best choice of kernel for SVM model is Gaussian kernel.

For SVM with Gaussian kernel, and different choices of gamma (0.01, 1, 10, 100, 1000), the testing error and AUC are shown in Figure. We can see that for SVM with Gaussian kernel, both testing error and AUC are sensitive to the choice of gamma. According to Figure 19, when gamma is 0.01 and cost is 10, SVM with Gaussian kernel achieves highest AUC. Therefore, the best choice of gamma and cost are 0.01 and 10 respectively.

The result in Table 7 shows the performance of SVM with best choice of each parameter (Gaussian kernel, gamma is 0.01, and cost is 10). Here the error rate for total and 'No' are small, but for 'Yes', the error rate is about 0.18, which is larger than the error rate for 'Yes' in random forest. The boundary plot is also shown in Figure 20.

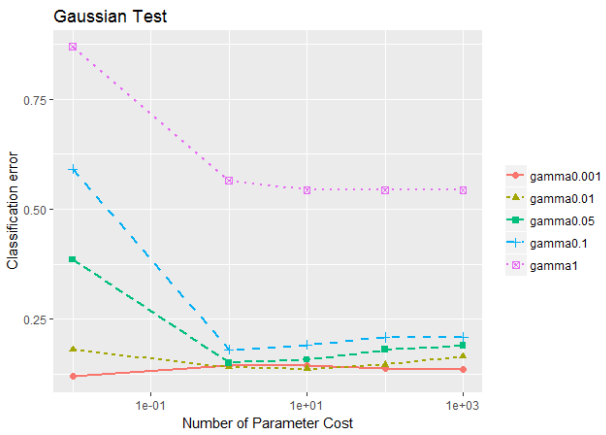

Figure 18: Test error for Gaussian kernal with different gamma and cost values

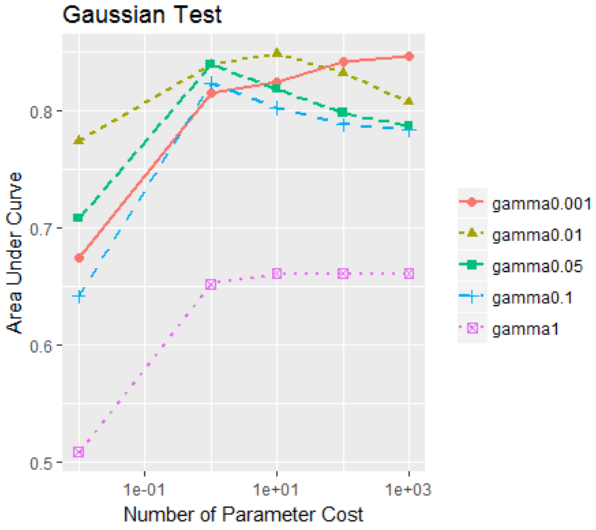

Figure 19: AUC for Gaussian kernal with different gamma and cost values

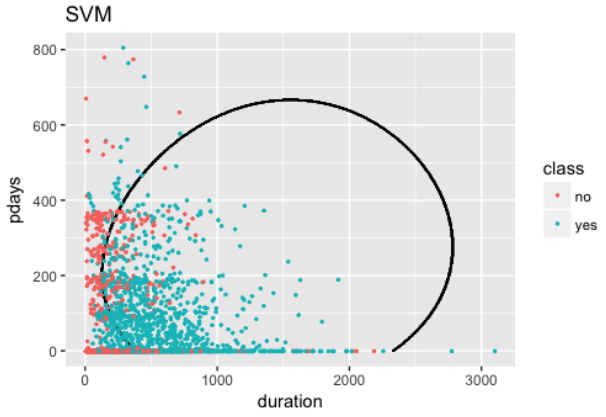

Figure 20: SVM Boundary

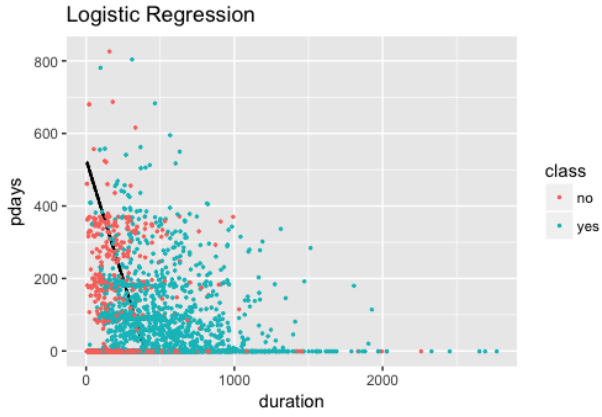

Figure 21: Logistic Regression Boundary

4.3.6 Logistic Regression

Main idea: This model arises when one wants to model the posterior probabilities of the K classes through linear functions of x. And then model the log-odd by a linear function with logistic function $\mu(x) = \frac{1}{1+\exp(-\eta(x))}$.

By applying Logistic Regression, we can compute the error rates for each class and AUC in Table 8. We can see that the overall error rate is around 0.1454. The error rate for 'Yes' is 0.2706 while 0.1289 for 'No'. Besides, the AUC we observed is 0.8003.This result is worse than SVM based on the AUC and error rate for 'Yes' generally. The boundary plot is shown in Figure21.

Table 8: Logistic Regression result on binary data

Error rate			AUC
Total	Yes	No	0.8003
0.1454	0.2706	0.1289	

5 Conclusion

Comparing the error rate for each class we have above in each method, the general rate for Boosting, 0.1271, is smallest while the one for Tree, 0.1714, is the largest. As for the class 'Yes', its error rate performs the best in Random Forest with 0.1438 and the worst is 0.3113 by using Bagging. The error rate for class 'No' in Boosting is 0.1205 while in Tree the error rate for class 'No' is 0.1676, which is the worst. As we once said in Section 2.1, the accuracy for prediction on 'Yes' is much more important and thus we need focus more on the error rate for 'Yes'. The results above indicate that Random Forest is the best for prediction on 'Yes'. Therefore maybe this method is our best choice. But we should not forget to take AUC into consideration.

As for AUC, it is generally between 0.78 to 0.86. The largest AUC is 0.8527 in Random Forest while the smallest one is 0.7834 in Bagging. We can also have a direct glance at Figure 22 which shows differnet Roc curves for different method. The plot and the AUC

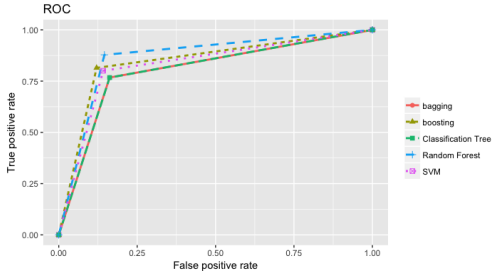

Figure 22: Roc curves for different models

value also imply that Random Forest performs the best since its Roc curve is the steepest and it owns the largest AUC value.

In conclusion, based on the error rate for 'Yes' and AUC we have computed, to predict on the purpose of clients on term deposits, Random Forest is the best method to help the bank institution.

References

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique*, (Journal of Artificial Intelligence Research 16, 2002).

[2] *Random Forest*, available at $https://en.wikipedia.org/wiki/Randomforest$.