A data-driven approach to predict the success of bank telemarketing

Yiping Pan (yp2524)
Qing Shen (qs2206)
Shiqi Song (ss5885)

Abstract

In this report, we tried the methods used in Sergio Moro's[1] paper, which are Logistic Regression (LR), decision trees (DT) and neural networks (NN) and support vector machines (SVM). By computing the probability matrix, we derive the ROC and Lift curves. Evaluating the metrics of AUC and ALIFT from the curves, we made a comparison of those methods. In addition, we explored Random Forrest, Bagging, Boosting, and clustering. We also explored META and clustering to classify and improve the original probability matrix by assigning different weights to the probability matrices. It turned out that KNN has the worst result, while Bagging is the best. To improve the performance, we added PCA and resampling in our preprocessing part. In PCA, when selecting 5 features, AUCs of LR and KNN increase by 6% and 3% respectively, whereas DT decreases by 10%. Using resampling, AUC of RF is 0.906, increasing 6% and becoming the highest.

1.Introduction

1.1 Problem description

This problem is about making decisions in telemarketing. The bank calls its clients to sell long-term deposits. If the client decides to subscribe to the deposit, this contact process is successful. So, the result is a binary unsuccessful or successful contact. In reality, it costs money to call clients, so the bank wants to narrow down the data set of clients and choose those who are more likely to deposit. The problem now is about binary classification.

In the previous paper, they used data collected from a Portuguese retail bank with 52,944 records, 150 features. The data is unbalanced as only 12% of records have the successful result. They tried several data mining methods like Logistic Regression (LR), decision trees (DT) and neural networks (NN) and support vector machines (SVM) by using rminer package in R. The coefficients they use to determine which model is better are AUC, confusion matrix and ALIFT curve.

1.2 Our work

In this report, the data has 41188 records with 20 features, representing clients' information like marriage status and education. LR, DT, NN, SVM, which are analyzed in the previous paper are constructed. Apart from that, we explored tree-based models such as Random Forrest, Bagging, Boosting and clustering, which we have learned in class. Some further exploration and improvement are made in the preprocessing part. Principal Components Analysis (PAC) is used to descend dimensions, and resampling is used to generate new data. In order to be consistent with the paper, metrics used here are AUC, confusion matrix and ALIFT curve as well.

2. Code frame

Importing and ensuring the completeness of the data first, then by plotting some features such as job and age, we can inspect a sense of how a feature plays a role in subscription.

We split the data into training data, including the first 40188 records and testing data, which includes the last 1000 records.

We predict the result from different models by using the function of predict. After deriving the probability matrix of each model, we can plot the confusion matrix and the receiver operating characteristic (ROC) curves. The accuracy can be evaluated by the area of the receiver operating characteristic curve (AUC). The closer AUC is to 1, the better the model is. AUC of 0.5 is equal to a random classifier.

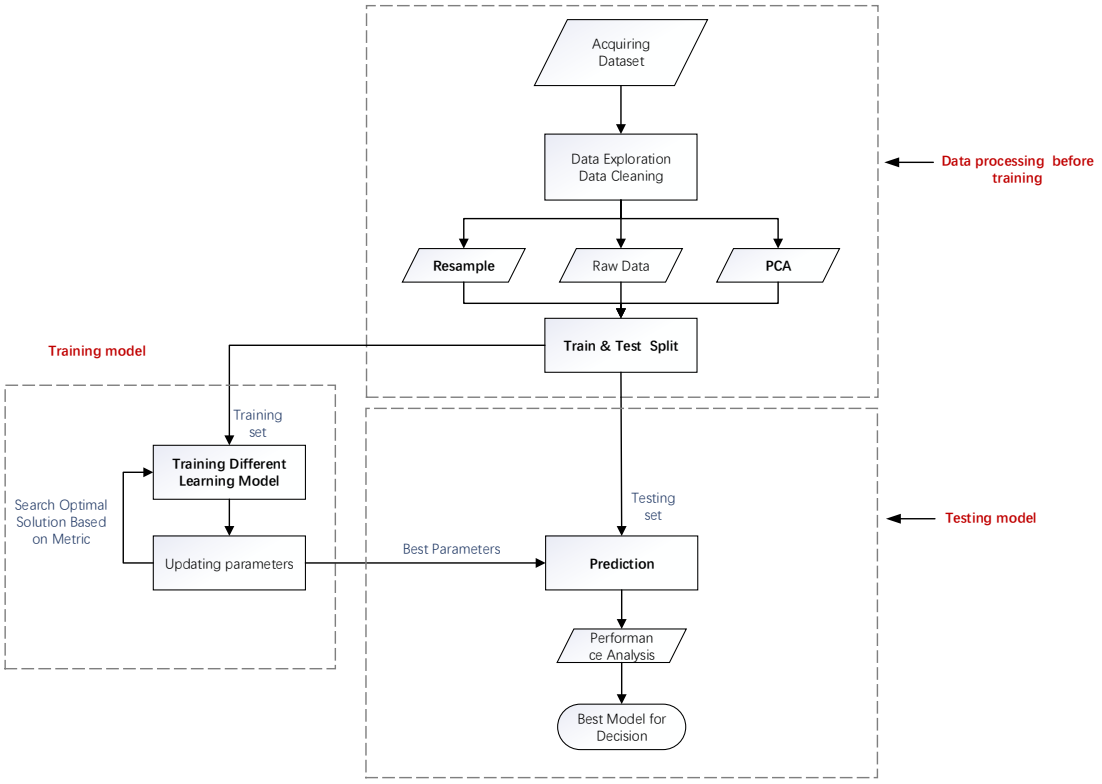

Figure 2.1 workflow

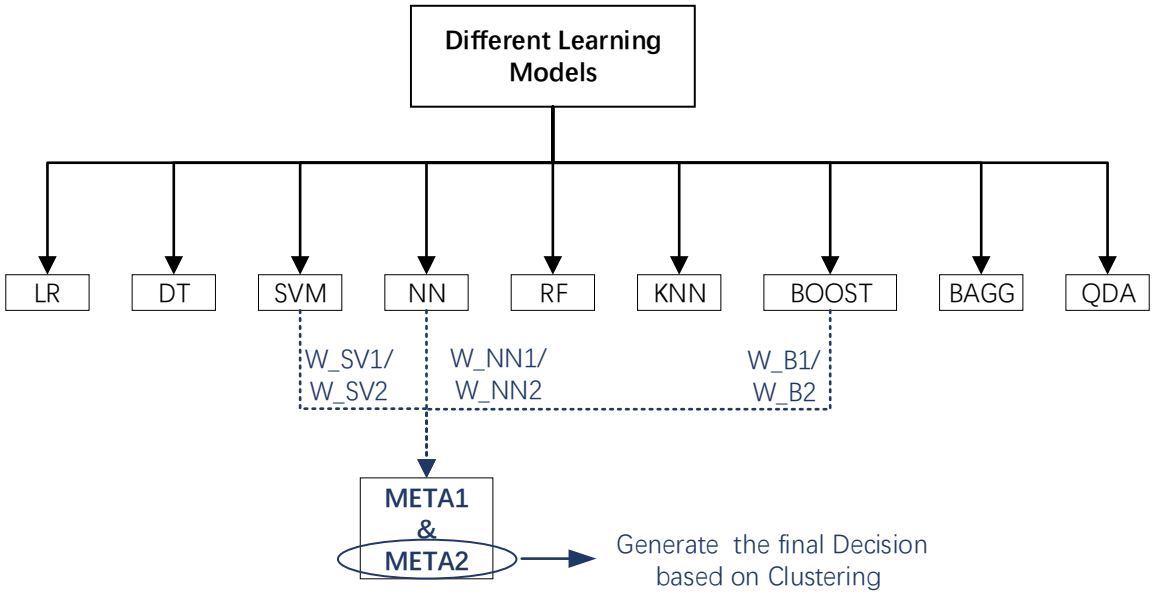

Figure 2.2 All Learning models

Having the probability matrix, we can also get the lift cumulative curve. Similarly, the ideal method should present an area under the LIFT (ALIFT) cumulative curve close to 1.0. A high ALIFT confirms that the predictive model concentrates responses in the top

deciles, while an ALIFT of 0.5 corresponds to the performance of a random baseline[1]. Repeat the above steps; we get the ROC and Lift curves of LR, DT, NN, SVM methods.

For the tree-based methods, instead of using predict function in the above section, we used mining function provided by rminer package to obtain the probability matrix. Then we explored RF, bagging, boosting, KNN, and their ROC and Lift curves, respectively.

In addition, we added PCA and resampling in the preprocessing part to compare whether these methods can improve the accuracy or not. The workflow is depicted in figure 2.1. Figure 2.2 shows all models we have used to accomplished classification.

3.Single Model Results

3.1 Confusion matrix and ROC

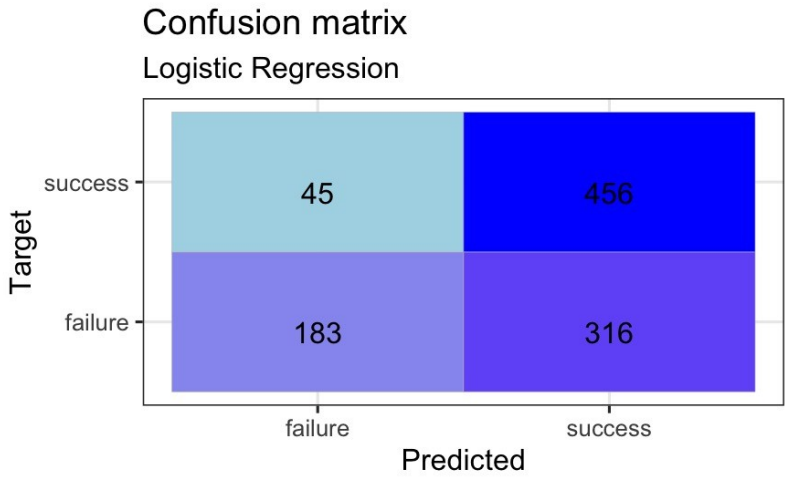

Figure 3.1.1 Confusion matrix of LR

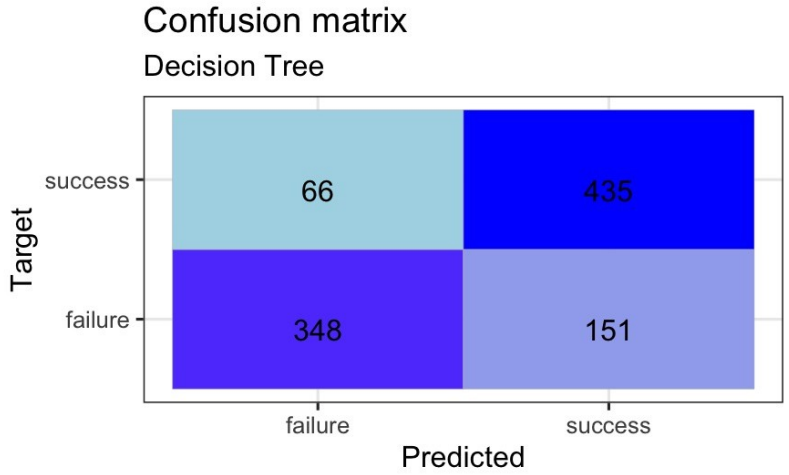

Figure 3.1.2 Confusion matrix of CT

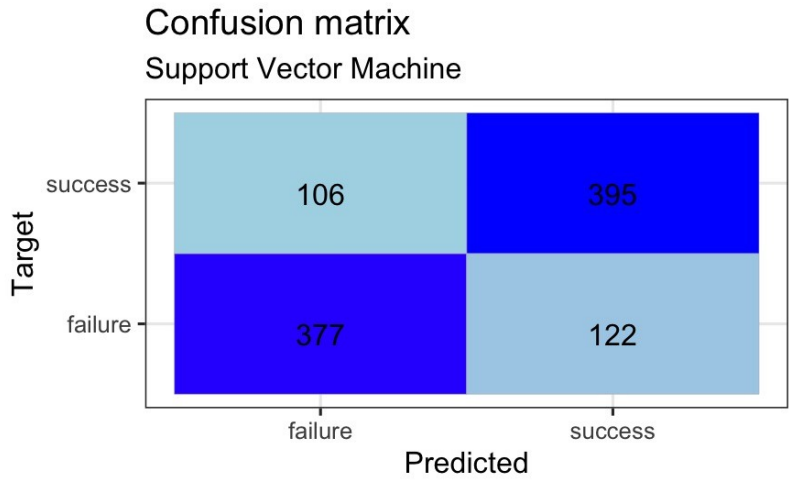

Figure 3.1.3 Confusion matrix of SVM

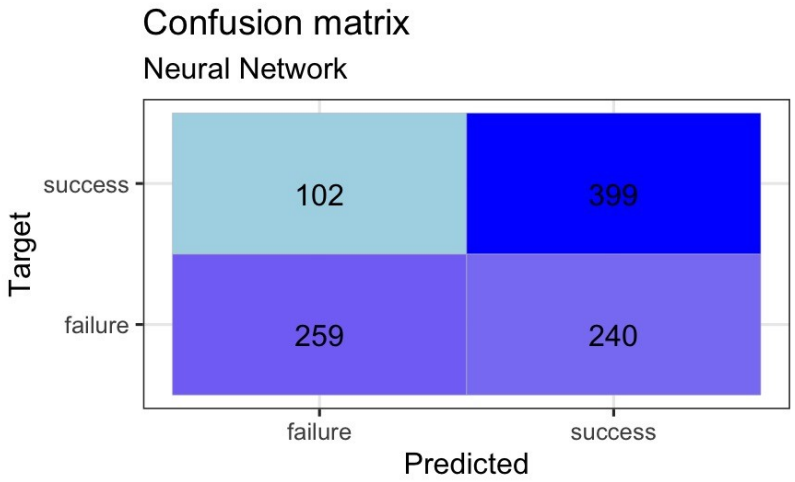

Figure 3.1.4 Confusion matrix of NN

In the confusion matrix, the darker the color is, the more samples are put into this region. We can see that the color in Neural Network is less distinct than SVM and DT. Since the parameters in our NN models are not fully tried and optimized due to the limit of time, the result of NN is not as good as the original paper. Generally speaking, KNN has bad performance in classifying imbalanced data. Accuracy is low for the rare class. Also, in our case, the distances between two different classes are not significantly larger than the distances of two samples inside one same class. For LR, the penalty is set as default; it may affect the result and may still result in overfitting.

3.2 ROC and LIFT curve

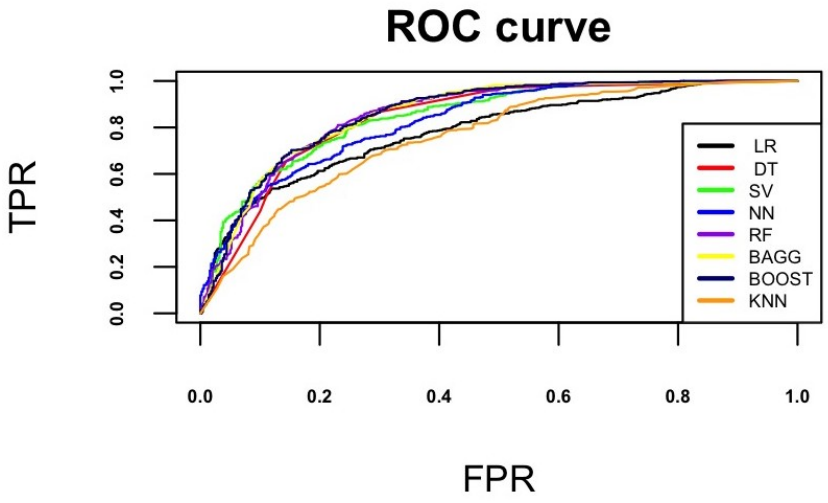

Figure 3.2.1 ROC curve

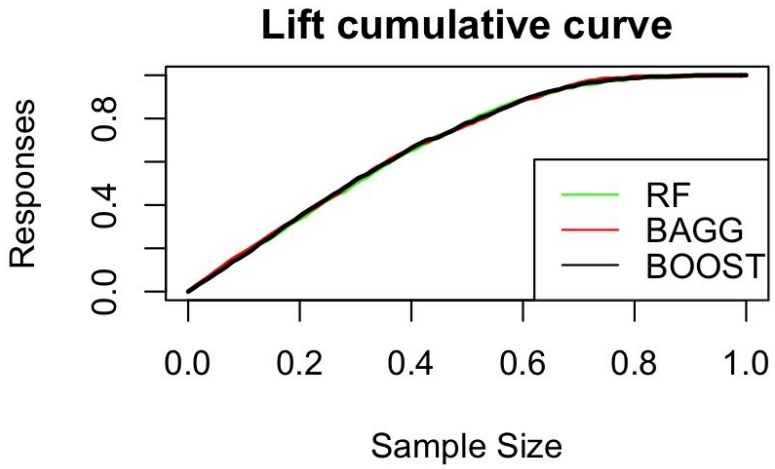

Figure 3.2.2 LIFT curve of tree-based models

For the ROC and Lift, the curves on the upper left position have better performance than the lower right ones. We can see that tree-based methods have higher AUC; their curves are relatively higher. And the Lift curves of tree-based methods are in close approximation. Of all the methods, boosting has the best performance, which is different from the original paper. Because boosting can adaptively change the weights of training samples and learn from different classifiers, then uses those classifiers to create a linear combination, achieving better performance than DT. Other methods have their own advantages in different parts.

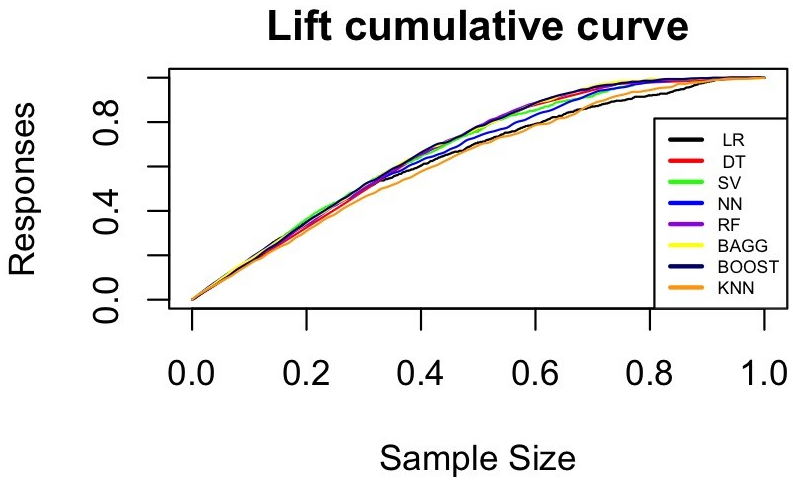

Figure 3.2.3 LIFT curve of tree-based models

Table 3.2.1 AUC of all models

LR	DT	SV	NN	RF	BAGG	BOOST	KNN
0.783	0.841	0.848	0.829	0.858	0.859	0.862	0.759

From the lift curve, we can see that tree-based methods have similar AUC and ALIFT. Similarly, as the result of AUC, KNN and LR have lower ALIFT, and tree-based methods are the best.

4. Additional Work

4.1 Blending Models

Motivation: in the previous steps, we tried many models and got AUC metric score of each instance for each model. Some models are better than others, however, we may not want to immediately give up the models that are not able to rank the highest. The reason is that these models are designed in different ways, therefore, each model will have the chance to overwhelm others in certain parts of data. In general, there is no absolute best model that can perform best in every case.

Method: Hence, we want to collect the information from all different kinds of models and make use of them. The method is to blend the predicted probabilities. After prediction on the test set, for each model, we will get predicted probabilities for 1000

instances. To blend them together, we are going to find weights for all the models. Then the weighted averaging will be performed.

Preparation: There is one more step for preparation before starting blending, which is model selection. We need to select a group of models that are most uncorrelated. The reason is that we cannot get more useful information from models that are in the same type. To explain more clearly, the purpose of blending is to take use of useful information from different models without adding much noise into the final model. Different types of models will have different behaviors. We aim to compensate the weak part of certain kinds of models with other models' support. Similar and correlated predictions cannot compensate with each other.

Preparation 1 - model selection by correlation matrix: We calculated the correlation matrix of the predicted probabilities of KNN, SV, DT, BAGG, RF, BOOST, LR, and NN. Then we use HAC (Hierarchical Agglomerative Clustering) to find strongly correlated models. The result is shown in fig.4.1.1. By HAC, without too much surprise, we found Boosting, Random Forest, Bagging, and Decision Tree are strongly correlated, which come from tree-based methods. So, we will use the best of them, Boosting, to perform the blending.

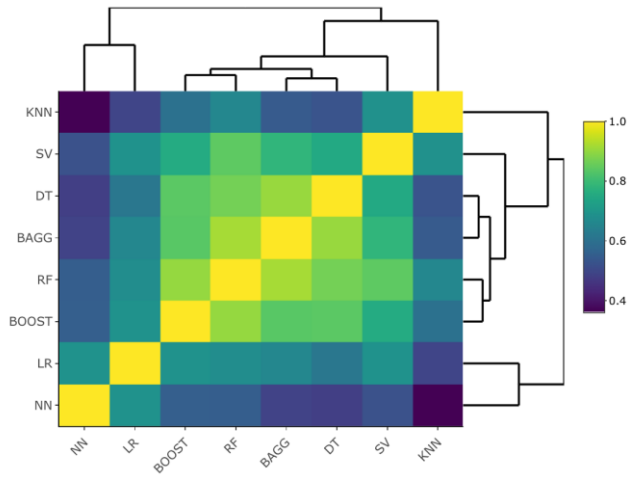

Figure 4.1.1 correlation matrix of all models

Preparation 2 - model selection by AUC: In addition, since the AUC metric (fig 4.1.1) of each model in our case should be close to each other (the maximum difference is 0.85 with 0.82), we will not assume the lower AUC models will influence the higher AUC models. If the AUC difference is very high, for example, 0.9 and 0.6, the model with 0.6 AUC might introduce more noise into blending instead of advantages. So, we will not use LR and KNN here.

Table 4.1.1 LIFT curve of tree-based model

LR	DT	SV	NN	RF	BAGG	BOOST	KNN
0.783	0.841	0.848	0.829	0.858	0.859	0.862	0.759

The models that we selected are: NN, BOOST, SV.

4.1.1 META1 – AUC weighted

We will use AUC as the criteria to combine the predicted probabilities and then get the new predicted probabilities.

For each model, its weight is defined as:

$$Weight = \boldsymbol{Normalize}(\frac{AUC - 0.5}{1 - 0.5})$$

A new column of prediction is then obtained, shown in tab.4.1.1 and 4.1.2.

Table 4.1.2 LIFT curve of tree-based model

	LR	SV	NN	BOOST	KNN
40189	0.886	0.984	0.918	0.625	0.604
40190	0.366	0.374	0.399	0.481	0.536
40191	0.237	0.036	0.019	0.237	0.118
40192	0.651	0.171	0.373	0.546	0.227

Table 4.1.3 New values

META
0.822
0.479
0.149
0.408

We compare the ROC performance of the new META model in fig.4.1.2.

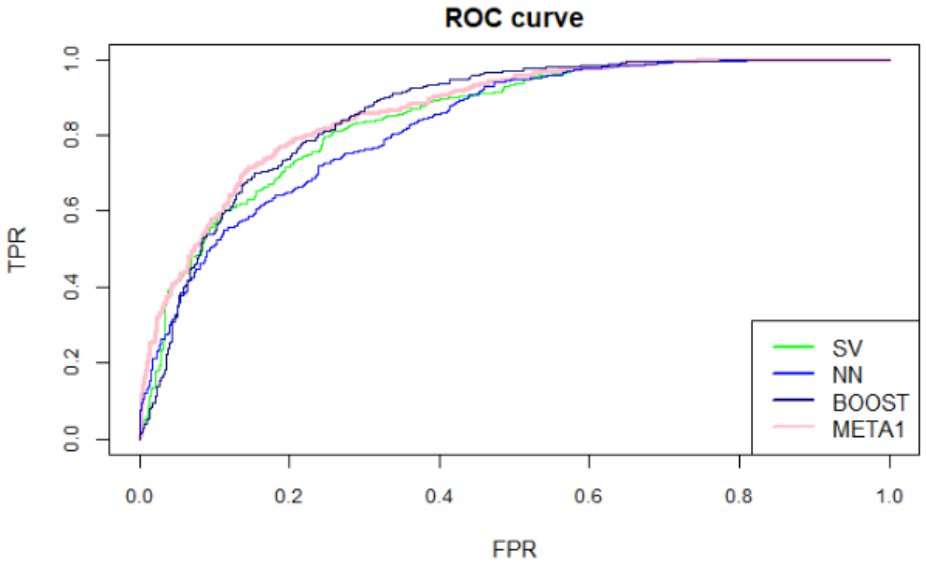

Figure 4.1.2 ROC curve using META

The pink curve in the plot shows that: the META1 model is not dominating among all the models, but it dominates in some certain TPR, FPR ranges. Specifically, at the region where FPR<0.3 and 0.6<TPR<0.8, representing good FPRs and good TPRs, the META1 model performs very well.

4.1.2 META2 – Clustering weighted

The implementation of weighted averaging is exactly the same as the previous META1. However, in META2, the difference is we are going to decide weights by K-means clustering – an unsupervised learning method.

Table 4.1.4 Probability matrix using clustering

	LR	SV	NN	BOOST	KNN	Cluster
40189	0.886	0.984	0.918	0.625	0.604	1
40190	0.366	0.374	0.399	0.481	0.536	0
40191	0.237	0.036	0.019	0.237	0.118	0
40192	0.651	0.171	0.373	0.546	0.227	0

As shown in tab.4.1.4 and table 4.1.5, firstly, we will perform K-means clustering on the predicted probabilities of five models, and the clusters are denoted as 0 or 1, which is exactly matched to our classification target, y.

Table 4.1.5 Weight of 40190

Weighs		
0.46	0.38	0.16
0.38	0.35	0.27
0.38	0.41	0.21
0.17	0.31	0.53

Then we defined the weight calculation formula. The explanation is that if the predicted probability of a model is closer to the clustered type, it will be allocated a larger weight; on the contrary, if the probability is far from its cluster, it will be given a smaller weight. For each row:

$$\boldsymbol{Score_i} = -3|Cluster_i - \boldsymbol{Probability_i}|$$

$$\boldsymbol{Weight}_i = Softmax(\boldsymbol{Score_i})$$

i: the row index

$\boldsymbol{Probability}_i$: 3-dimensional row vector representing probabilities in each row.

$\boldsymbol{Score_i}$: 3-dimensional row vector representing how close the probability is to its cluster.

$\boldsymbol{Weight}_i$ 3-dimensional row vector representing weights assigned to each row

The ROC curve results are shown in fig.4.1.3.

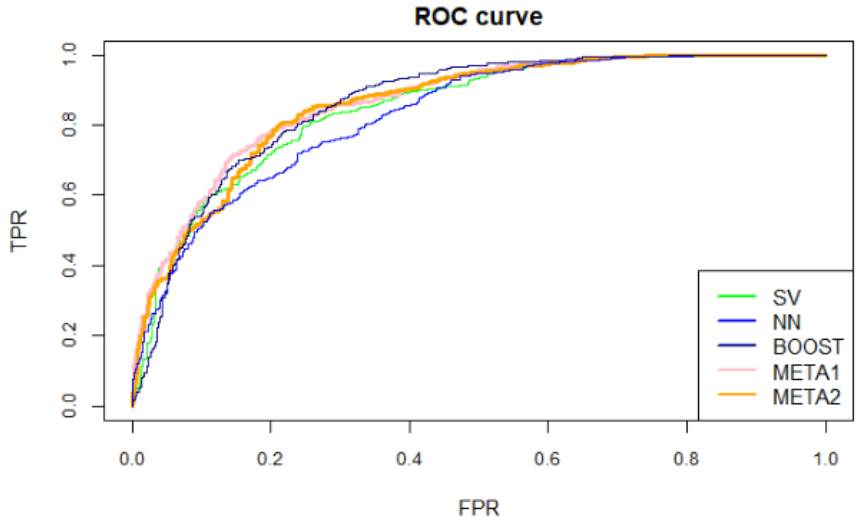

Figure 4.1.3 ROC curve using META2

For META2, we found it also has its advantages. In the specific area where FPR is around 0.25 and TPR is around 0.8, the META2 behaves the best. However, other parts it may not compete with META1 or BOOST.

Summary: the performance of META1 and META2 is not dominating all the time but at certain ranges of TPR and FPR they perform pretty well. This is meaningful because usually, the business decision will be done with a selection of relatively high TPR and low FPR. META1 and META2 dominate in the most important region (TPR>0.6, FPR<0.3).

4.2 Preprocessing – PCA and Resample

4.2.1 PCA

Method:

We implemented Principal component analysis (PCA) tried in original models. First, we used dummy.data.frame() function to transfer all contents into numerical values. By converting all observations into a set of values of linearly uncorrelated variables. Then we selected five and twenty components with the largest proportion of variance

Results:

We have feature numbers n=5 and n=20. The AUC scores with PCA are shown in table 4.1 and table 4.2, respectively. We can find that the AUC of LR improved from 0.783 to 0.8316/0.8497 whereas AUC of DT decreased from 0.841 to 0.751/0.751, the results of other models don't change significantly.

Table 4.2.1 AUC without PCA

LR	DT	SV	NN	RF	BAGG	BOOST	KNN
0.783175	0.8411554	0.8482514	0.8290153	0.8579294	0.8593434	0.8618174	0.759167

Table 4.2.2 AUC with PCA (n=5)

LR	DT	SV	NN	RF	BAGG	BOOST	KNN
0.8316	0.750915	0.8324	0.8061952	0.8436494	0.8558694	0.860213	0.80812

Table 4.2.3 AUC with PCA (n=20)

LR	DT	SV	NN	RF	BAGG	BOOST	KNN
0.8497	0.750915	0.8139633	0.8220313	0.8471534	0.8557754	0.860213	0.724619

Analysis:

Since PCA convert all features into a new space, we can get different AUC values in LR and DT. For RL, new data can be approximated to a linear model better, and thus ACU is relative higher. For KNN, PCA effectively reduces the distance among different class when n = 5 so AUC increase from 0.759167 to 0.80812.

4.2.2 Resample

Motivation:

As plotted in fig.4.2.1, we found the training dataset is highly imbalanced. It contains 36049 samples with label "no" but only 4139 with label "yes". The imbalanced data may influence the training process because models will probably pay more attention to those samples with label "no". As a consequence, our trained models may perform weakly on those test data with potential label "y".

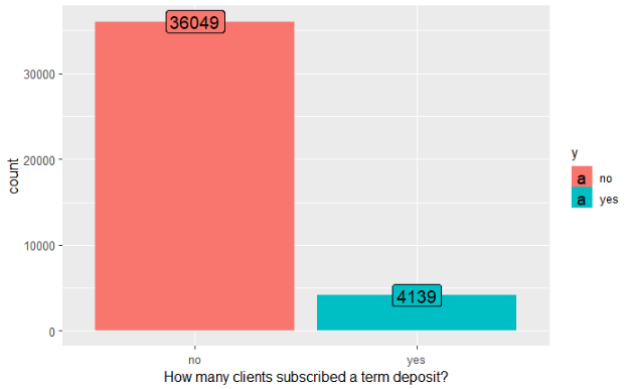

Figure 4.2.1 Imbalanced distribution of original data

Method:

In order to make the training set balanced, we oversample the data by generating more data from the original training samples, which have labels "1". We used MWMOTE algorithm to intelligently generate new data. The final training set contains samples with the same amounts of labels “no” and “yes”.

Result:

After testing, we found most models perform worse with the resampled data, except ensembled tree-based models. Random Forest model is used to show the result. The ROC curves are plotted in fig.4.2.2.

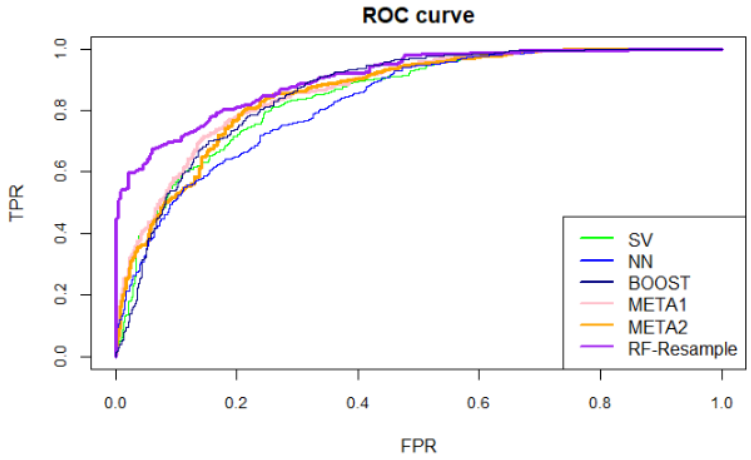

Figure 4.2.2 ROC curve with RF Resample

It is interesting that Random Forest with resampled data, i.e., balanced data, overwhelms other previous models. The most important advantage is: within the range TPR<0.5, the FPR keeps almost 0. With the range 0.6<TPR<0.8, this result is significantly better than others. The explanation is that Random Forest makes use of part of sampled data to training weak trees and the weak decision trees are based on separating data. Therefore, the sampling it does will significantly influence every single weak tree. With a balanced dataset, Random Forest will have more chance to learn all the information from different labels of data. The difference is that other models will not gain more information in the training process and will even deal with generated data as noise data.

5. Conclusion

In the project, we successfully reproduced the experiment conducted by the original research paper and compared different models, including LR, DT, SV, and NN. We provided ROC, confusion matrix and LIFT analysis on the prediction results from models.

The decision support system can be constructed basing on our models. Then, we add several more models, including Random Forest, Bagging, Boosting, and KNN. Boosting is found to be the best. In addition, new designed META models are performed. We blended the previous models together by weighted averaging, using AUC weights and Clustering weights separately. Both models performed well in the end. Furthermore, from another angle, we started from preprocessing, we performed PCA and resampling methods. PCA partly improved the original models slightly. However, resampling highly improved Random Forest method and we finally get a very strong prediction from this model. The best model can strongly support the bank telemarketing problem.

In the future, more work can be done by adding more base models, try better model blending, and more combinations of methods that we have used in this project.

Reference

[1]Sergio Moro, Paulo Cortez, Paulo Rita A data-driven approach to predict the success of bank telemarketing [J].Decision Support Systems 62 (2014) 22–31.