Comparing the Performance of Five Classification Models on an Imbalanced Dataset

Yan Song

Thompson Rivers University, 805 TRU Way, Kamloops, B.C., Canada

Abstract: **Imbalanced data is a common challenge in classification problems, particularly in areas where certain types of samples are rare and difficult to obtain, such as disease diagnosis, customer churn prediction, and fraud detection. This study aims to evaluate the performance of five conventional classification models on an imbalanced dataset of bank marketing, which only contains 10% positive samples. We revisit the basic knowledge of SVM, decision tree, random forest, boosting and Neural Networks. Without specially designed methods to handle the imbalance, we evaluate their performance on the dataset based on accuracy, sensitivity, specificity, and ROC curve. Our findings suggest that Neural Networks exhibit the highest sensitivity, while tree-based models achieve better AUC and accuracy. This implies that Neural Networks are particularly effective in addressing the challenges presented by imbalanced datasets. The source code of this study is available at: https://github.com/yan-song-211/5420-project.**

Index Terms: Binary classification, imbalanced data, model comparison.

1. Introduction

Marketing campaign is a strategic activity that is widely used across various industries to promote a particular product, service, or brand to a specific target audience. The primary objective of a marketing campaign is to generate interest, increase brand awareness, and ultimately boost sales. It has been used to promote bank deposit subscriptions for a long history. The success of a marketing campaign heavily relies on recognizing the audience's demographics, behaviors, and other characteristics. The ability to determine the probability of a promotion's success based on the target audience's attributes remains an essential and long-lasting problem and it casts challenges to the business and market sections.

With the rapid advancement of machine learning techniques, this problem in marketing can now be addressed using advanced algorithms and approaches. By leveraging the vast amounts of historical data on marketing campaign activities, it is now possible to identify patterns and relationships between audience attributes and the probability of successful sales using modern machine learning techniques.

However, there is a significant challenge associated with marketing campaign data in classification. The data is often imbalanced due to a low success rate in market campaign activities, which poses difficulties for accurate predictions. The imbalance in the data creates a bias towards the majority class and makes it harder to identify the minority class. This is because there are significantly fewer samples of the minority class compared to the majority class. As a result, it is challenging to correctly identify the minority class, and this poses significant challenges for traditional classification models.

The goal of this study is to evaluate the effectiveness of five widely-used and advanced classification models on an imbalanced bank marketing dataset, predicting the success of the campaign activity. The five methods utilized in the study are SVM [1], decision tree[2], random forest[3], boosting[4], and Neural Networks [5]. The dataset adopted in this study is the popular bank marketing dataset proposed in [6]. It collects records from direct marketing campaigns of a Portuguese banking institution based on phone calls. The main components of this project include:

1) Exploratory data analysis on the dataset

2) Applying the five classification models on the dataset to predict the success of the campaign
3) Comparing and discussing the performance of the classification models on the imbalanced dataset

The remaining parts of the manuscript starts with an introduction and an exploratory data analysis of the dataset followed by data manipulation. The paper then presents an overview of the five methods used in the study, along with their underlying principles. The subsequent sections describe the experimental results and conclude with a summary.

2. Data

2.1. Dataset overview

The bank marketing dataset[6] (https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#) is available in four different versions, each varying in the number of samples and features. For this study, we adopt the ”bank-additional” version, which contains 4119 samples and 20 variables, including 19 features and 1 output. This version is specifically designed to test more computationally demanding machine learning algorithms. The primary goal of the classification task is to predict whether a client will subscribe to a term deposit, indicated by the binary variable y with values ”yes” or ”no”. It is a mixed dataset, consisting of 19 features that include demographic attributes such as age, job, marital status, and education; economic-status-related attributes such as housing, loan, and employment variation rate; as well as campaign history features such as the previous number of contacts.

2.2. Exploratory data analysis

Firstly, as suggested by the author, we exclude the feature of ”duration” since it highly affects the output target and should be discarded if the intention is to construct realistic predictive models.

Of the 19 features, eight are numeric/quantitative, and their distributions are illustrated in Figure 1. We notice that there is no significant difference between the distributions of most of the quantitative features for the two categories, including age, campaign, consumer price index, consumer confidence index, and the number of contacts performed before. Although the ranges of the remaining three features are similar for the two categories, there are relatively notable differences between the averages.

The histograms are plotted for the 11 qualitative features and the binary output in Figure 2. As we have mentioned, this dataset is imbalanced as the positive samples only account for 10.9%. We notice that there are trivial levels in three features. There is only one sample of ”yes” in ”default” and one sample of ”illiterate” in ”education”. There are 11 samples of ”unknown” in ”marital”. 20 levels account for only 3.9% of the total samples in ”pdays”.

2.3. Data manipulation

After the exploratory data analysis, for the sake of reducing the number of levels, we merge the single samples in ”default” and ”education” into the level of ”unknown”, and the 11 samples in ”marital” into the level of ”married” as it has the most samples. We also merge the 20 levels in ”pdays” into one level of 0.

3. Methods

In this section, we elaborate on the five classification models regarding the basic knowledge, including the objective functions and the optimization approaches.

3.1. SVM

Support Vector Machine (SVM)[1] is an effective binary classifier that has been adopted for various applications. Its objective is to identify the widest separating margin of the two categories. With non-linear kernels, SVM is able to deal with data non-linearly separable without simply enlarging the feature

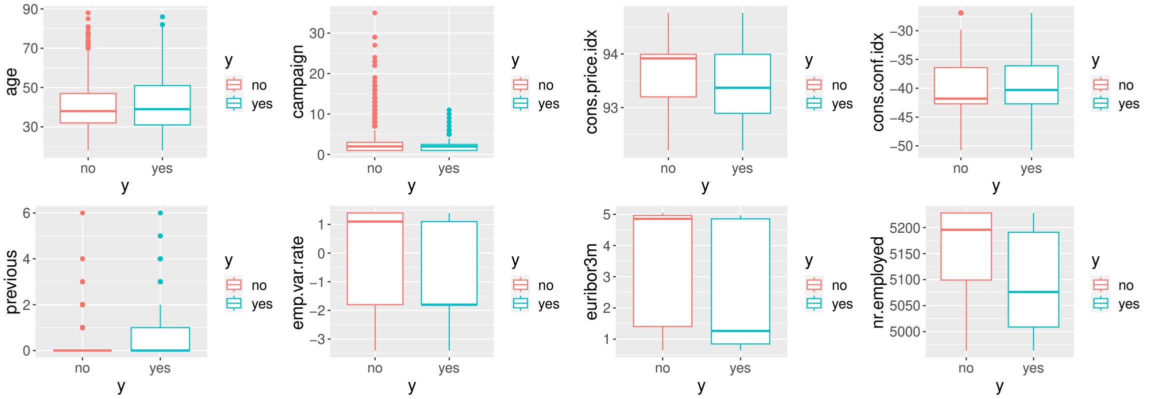

Fig. 1: Boxplots of eight quantitative features

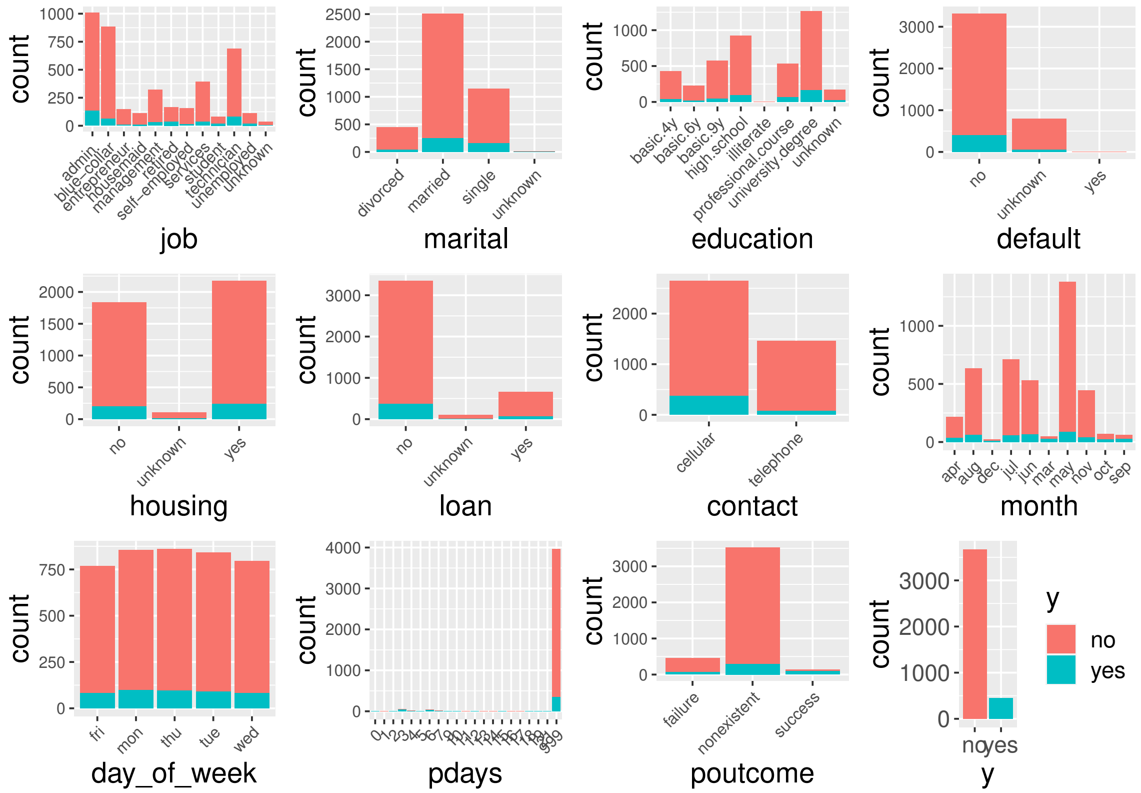

Fig. 2: Histograms of twelve qualitative features

space. The objective function is

$$maximize \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i k(x_i, x_j) y_j c_j$$

$$s.t. \sum_{i=1}^{n} c_i y_i = 0, 0 \le c_i \le \frac{1}{2n\lambda}$$

where $k(x_i, x_j)$ is the kernel function and λ is a hyper-parameter. Sub-gradient descent algorithm [7] is one of the mainly used approach for solving the optimization problem.

To optimize the performance of SVM on our dataset, we perform hyper-parameter tuning for C with a linear kernel using repeated 5-fold cross-validation. Additionally, we compare the performance of four different kernels, namely linear, sigmoid, radial, and polynomial, using ROC curves and AUC.

3.2. Decision tree

Decision tree[2] is one of the tree-based models. For classification, decision tree is called classification tree which splits one of the two previously identified regions in the feature space in an iterative way. It uses a top-down, greedy approach known as recursive binary splitting, selecting a predictor x_j and a cutpoint s in each step such that splitting the space into two regions R_1 and R_2 which satisfy:

$$minimize \sum_{i:x_i \in R_1(j,s)} 1 - \max_k(\hat{p}_{1k}) + \sum_{i:x_i \in R_2(j,s)} 1 - \max_k(\hat{p}_{2k}) \tag{1}$$

where p_{1k} and p_{2k} denote the proportions of training observations in the two regions that are from the kth class.

In order to decrease the variance of the model and prevent overfitting, researchers use tree pruning technique to obtain a smaller subtree satisfying:

$$minimize \sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} [1 - \max_k(\hat{p}_{mk})] + \alpha|T| \tag{2}$$

where T is a subtree and $|T|$ is the number of its terminal nodes. α is hyper parameter tuned by cross validation.

We first construct a decision tree and use 10-fold cross-validation to find the number of misclassifications as a function of α. Then we prune the full tree with the best number of terminal nodes obtained in the cross-validation.

3.3. Random Forest

One of the drawbacks of decision trees is their sensitivity to changes in the tree structure, which can lead to instability in their predictions. To mitigate this issue, several ensemble methods have been proposed to aggregate the results of multiple trees. One such method is the random forest[3], which aims to decorrelate the decision trees built on bootstrapped training samples by using only a subset of predictors in each splitting. This approach helps to reduce the correlation between the decision trees, thereby increasing the diversity of the trees and reducing the risk of overfitting.

The number of predictors considered in each subset during the splitting process is a crucial hyper-parameter in the random forest model. When this parameter is set to the total number of predictors, the model becomes equivalent to bagging. We use 5-fold cross-validation to find the best value of this parameter. Additionally, we investigate the impact of the number of trees on Out-Of-Bag (OOB) error with different values (2, 4, 6) of this parameter.

3.4. Boosting

Boosting is a generalized scheme of ensemble learning. Here, we limit it in the context of tree-based model[4]. Different from bagging and random forest, boosting works in a sequential way, meaning the tree is growing based on the previous one. Essentially, in each step of tree-growing, the new tree is fitted to predict the classification error generated from the previous step, rather than the class label. In regression problem, classification error is substituted by residual. It is updated with a shrinkage parameter α in the bth iteration:

$$y_i \leftarrow y_i - \alpha \hat{f}^b(x_i) \tag{3}$$

In this way, it is slowly improved in areas where it does not perform well previously by integrating the new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \alpha \hat{f}^b(x) \tag{4}$$

Finally, the boosted model is obtained by:

$$\hat{f}(x) = \sum_{b=1}^{B} \alpha \hat{f}^{b}(x) \tag{5}$$

We use grid search to optimize four parameters: the total number of trees (10, 50, 200), the maximum depth of each tree (1, 3,5), the minimum number of observations in the terminal nodes of the trees (10, 15, 20), and the shrinkage parameter (0.01, 0.1 0.3).

3.5. Neural Networks

Neural Networks[5] are a powerful non-linear modeling approach that, like tree-based methods, map a set of input features to an output response. They have an unique architecture comprising multiple layers, each consisting of several interconnected neurons. The output of each neuron is a linear combination of the outputs from the previous layer, which is then passed through a nonlinear activation function:

$$u_j^{(k)} = \sum_{i=1}^{n} w_{ij}^{(k-1)} A_i^{(k-1)} + w_{0j}^{(k-1)}$$
$$A_j^{(k)} = g(u_j^{(k)})$$

where $u_j^{(k)}$ is the input of the jth unit in kth layer, $w_{ij}^{(k)}$ is the weight from the ith unit in the kth layer to the jth unit in the $(k+1)$th layer. $A_i^{(k)}$ is the output of the ith unit in the kth layer, and $g(\cdot)$ is the specified activation function.

The activation function plays a critical role in the overall structure of neural networks. Without an appropriate activation function, a neural network would reduce to a linear combination, regardless of the number of layers or neurons. In other words, the activation function introduces non-linearity into the model, enabling it to learn complex relationships between input and output.

There are various techniques for fitting neural network models, but one of the most widely used is gradient descent. This approach begins by initializing all the parameters, including the weights, randomly. It then iteratively updates these parameters with small changes to minimize the objective function, which in turn reduces classification error, until a stopping criterion is met. Although relatively straightforward, gradient descent has proven to be an effective method for optimizing neural networks.

We do not conduct an exhaustive hyper-parameter tuning for the Neural Network model, including the number of layers and units, activation function, learning rate, etc, due to time limit. We use a network with 2 hidden layers of 5 units and 2 units, respectively. In addition, as there are only two activation function choices available in the ”neuralnet” package, we chose the ”logistic” function.

The source code of this study is available at: https://github.com/yan-song-211/5420-project.

4. Results

In this section, we first provide an overview of our experiment setup and evaluation methodology. Then, we present the results for each individual model. Finally, we compare the overall performance of the five models and discuss their ability to handle the issue of imbalanced data.

4.1. Experiment setup and evaluation

To evaluate the effectiveness of the classification models, we conduct binary classification on the Bank Marketing Dataset [6]. We randomly divide the dataset into two parts, including a training set with 80% samples and a testing set with 20% samples. The experiments are implemented by R, including the packages of ”e1070”, ”tree”, ”randomforest”, ”gbm”, and ”neuralnet”. .

For a problem of binary classification, evaluating the performance of a model is crucial. Two commonly used evaluation metrics are the Receiver Operating Characteristic (ROC) Curve [8] and Accuracy. The ROC curve is a graphical representation of the true positive rate (TPR) against the false positive rate (FPR) at different decision thresholds. The Area Under Curve (AUC) is a single number that summarizes the

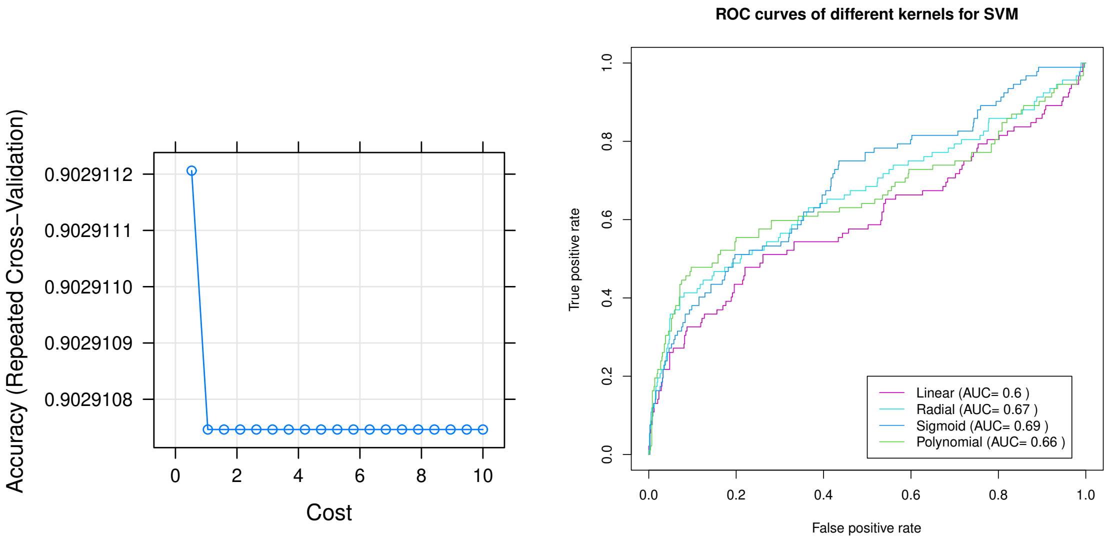

Fig. 3: Left: Hyper-parameter tuning for C; Right: ROC curves of different kernels for SVM

ROC curve's performance, with a value of 1 indicating perfect performance and a value of 0.5 indicating random guessing.

Accuracy is another evaluation metric that measures the proportion of correctly classified samples, i.e., the total number of samples correctly classified divided by the total number of samples. When considering only positive samples, accuracy becomes sensitivity, and when considering only negative samples, it becomes specificity. Since the dataset is imbalanced and contains only 10.9% positive samples, it is crucial to consider sensitivity and specificity along with accuracy.

4.2. Results of SVM

Hyper-parameter tuning is conducted with a linear kernel using repeated 5-fold cross-validation. The accuracy is evaluated for different values of C, and the results are presented in the left graph of Fig. 3. We observe that the highest accuracy is achieved at $C = 0.52$. The results of comparing the four kernels are depicted in the right graph of Fig. 3. Our analysis reveals that the sigmoid kernel outperforms the other kernels with an AUC of 0.69 on our dataset.

4.3. Results of tree-based methods

In this section, we conduct experiments using decision tree, random forest and boosting. The full decision tree and the pruned tree are displayed in Fig.4.The pruned tree is obtained by applying 10-fold cross-validation to determine the optimal tree complexity, resulting in a pruned tree with only three terminal nodes.

For the random forest model, we use cross-validation to select the optimal number of variables randomly sampled as candidates at each split. The best value is found to be 2, as shown in the left graph of Fig.5, where we plot the Out-of-Bag (OOB) error against the number of trees for different values of the parameter.

The training process of grid search for boosting is illustrated in the right graph of Fig. 5, with different choices of max tree depth, indicating that a max tree depth of 1 yields the best performance. After the grid search, total number of trees is set to 50, the maximum depth of each tree is set to 1, the minimum number of observations in the terminal nodes is set to 10, and the shrinkage parameter is set ot 0.1.

4.4. Comparison and discussion

We plot the ROC curves of the five models in Fig.6. There is no significant difference among the five curves. Random forest and boosting perform best of AUCs of 0.76 and 0.75 respectively. On the other

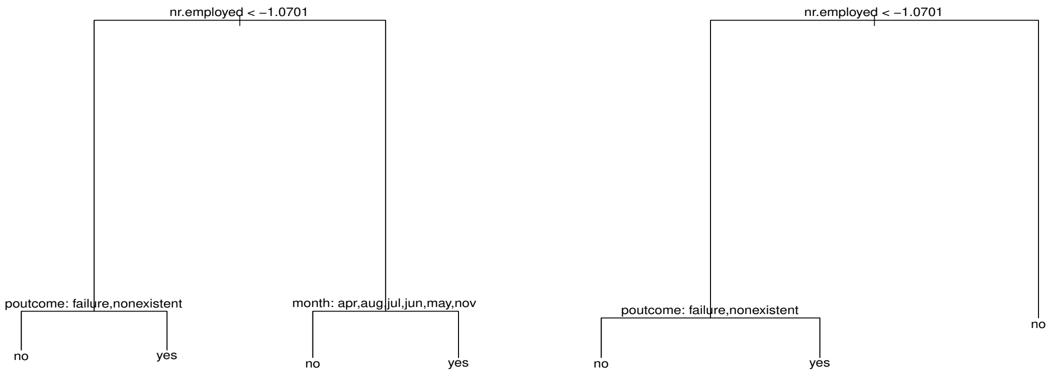
nr.employed < −1.0701
poutcome: failure,nonexistent
month: apr,aug,jul,jun,may,nov
no
yes
no
yes
nr.employed < −1.0701
poutcome: failure,nonexistent
no
no
yes

Fig. 4: Left: Decision tree ; Right: Pruned tree with 3 nodes

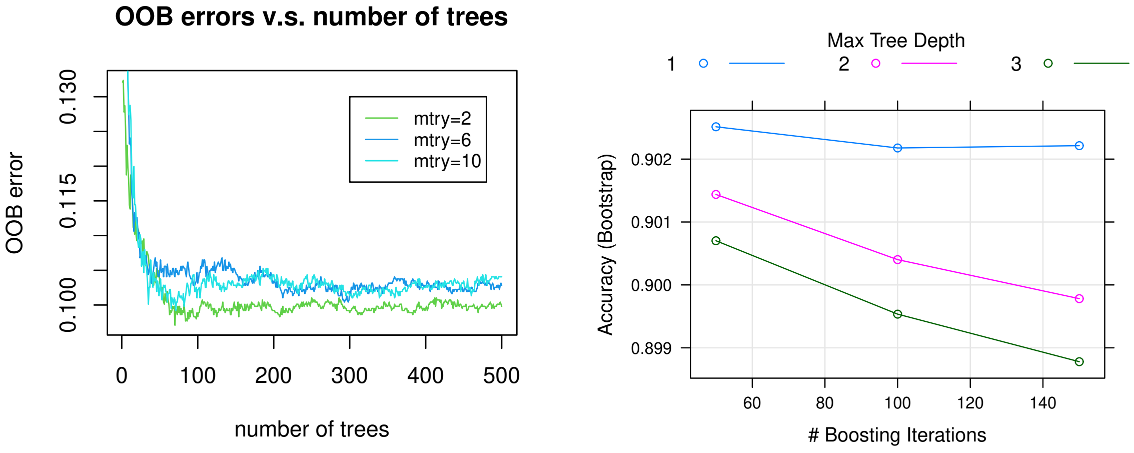

Fig. 5: Left: OOB error versus number of trees for random forest; Right: Accuracy versus number of Boosting iterations

hand, decision tree and Neural Network models have similar performance with an AUC of 0.71. The SVM model performs the worst among the five models, with an AUC of 0.69.

Table I presents the accuracy, specificity, sensitivity, and *p*-value of different models, including two kernels in SVM and the decision trees before and after pruning. Our results show that the boosting model achieves the highest accuracy of 0.8919, outperforming the Neural Network by nearly 0.02. Among all models, random forest has the highest specificity. However, as our dataset has only 10% positive samples, sensitivity is a more appropriate measure for evaluating model performance. Interestingly, our findings reveal that the Neural Networks achieve the highest sensitivity and *p*-value, suggesting that it has the lowest bias towards the majority class and performs best for overcoming imbalance during model fitting.

It is important to note that we do not conduct an exhaustive hyper-parameter tuning for the Neural Network model. However, our results demonstrate that a Neural Network model without fine-tuning already outperforms traditional models in addressing the problem of imbalance in classification.

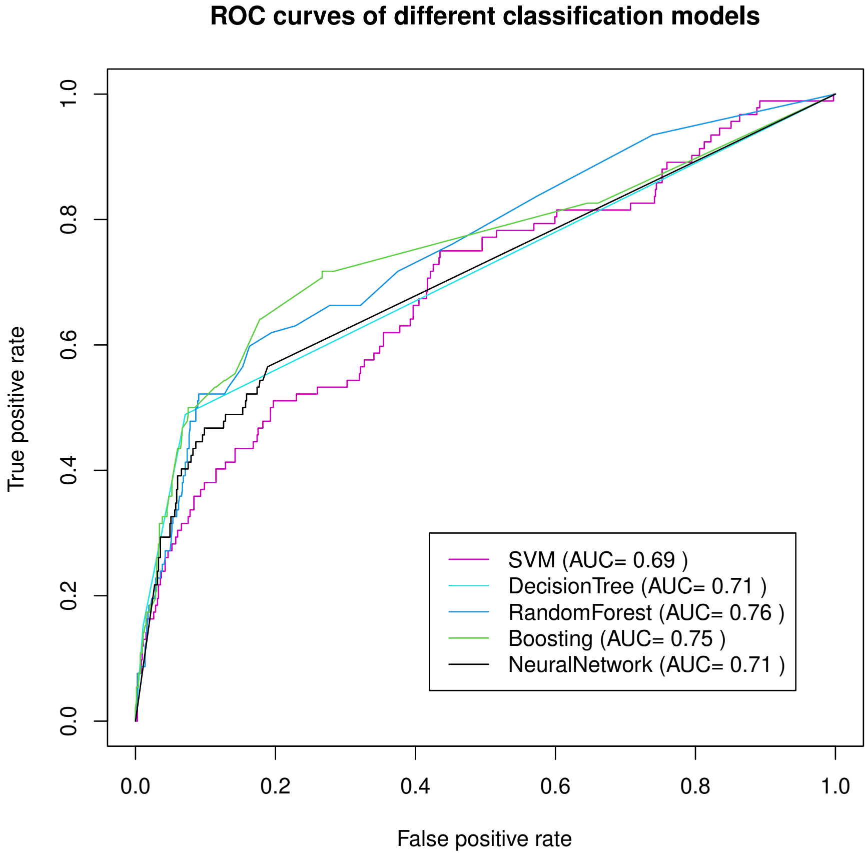

Fig. 6: ROC curves of the five models

TABLE I: Comparison of different models

Method	Accuracy	Specificity	Sensitivity	p-value
svm (linear)	.8812	.9726	.1630	.7315
svm (sigmoid)	.8906	.9863	.1304	.4396
Decision tree (full)	.8894	.9822	.1522	.4836
Decision tree (pruned, 3 nodes)	.8955	.9891	.1522	.2745
Random forest	.8906	**.9918**	.0870	.4396
Boosting	**.8919**	.9850	.1522	.3962
Neural network	.8724	.9576	**.1957**	**.9302**

5. Conclusions

This study aims to investigate the performance of five traditional classification models on an imbalanced dataset. We apply SVM, tree-based methods, and the Neural Networks on the Banking Marketing dataset, which only contains 10% positive samples. We investigate the impact of parameter tuning using cross-validation. We compare their performance in terms of ROC curve, accuracy, sensitivity and specificity. The results reveals that tree-based methods have a higher AUC and accuracy, while Neural Network exhibits the highest sensitivity. It suggests that Neural Networks are particularly adept at overcoming the challenges posed by imbalanced datasets.

References

[1] Noble WS. What is a support vector machine? Nature biotechnology. 2006;24(12):1565-7.

[2] Song YY, Ying L. Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry. 2015;27(2):130.

[3] Breiman L. Random forests. Machine learning. 2001;45:5-32.

[4] Drucker H, Cortes C. Boosting decision trees. Advances in neural information processing systems. 1995;8.

[5] Gurney K. An introduction to neural networks. CRC press; 1997.

[6] Moro S, Cortez P, Rita P. A data-driven approach to predict the success of bank telemarketing. Decision Support Systems. 2014;62:22-31.

[7] Shalev-Shwartz S, Singer Y, Srebro N. Pegasos: Primal estimated sub-gradient solver for svm. In: Proceedings of the 24th international conference on Machine learning; 2007. p. 807-14.

[8] Janssens ACJ, Martens FK. Reflection on modern methods: Revisiting the area under the ROC Curve. International journal of epidemiology. 2020;49(4):1397-403.