Classification Tree for Effective and Efficient Marketing of Term Deposits

Andres Ramirez, Gizelle Lao, Ali Mirza, Rafael Muniz, and Trung Le

Business Insights & Analytics, Humber College Institute of Technology and Advanced Learning, Toronto, Canada

***Abstract* – An important application of machine learning models is predicting if clients will subscribe to term deposits or not before marketing campaigns to assist banks in improving revenue while performing cost-efficient marketing efforts, therefore increasing profits. Marketing campaigns for subscribing to term deposits have various independent factors to consider, such as client age, job, marital status, education level, day and month of contact, etc. In this paper, five algorithms – namely Linear Discriminant Analysis, K-NN, Naïve Bayes, Classification Tree, and Logistic Regression – have been tested to model relationships between independent factors for subscribing to term deposits. Test results show the best performance by the Classification Tree, which has an overall accuracy of 87%, and an AUC of 92% in predicting whether or not clients will subscribe to term deposits.**

***Index Terms*— Term deposits, Linear Discriminant Analysis, K-NN, Naïve Bayes, Classification Tree, Logistic Regression.**

I. INTRODUCTION

The banking sector is dynamic and needs to know its customers in the best way possible to apply resources effectively and efficiently. Banks need to sell financial products to leverage the money deposited from their clients to generate money in this operation. If banks can determine which customers are likely to purchase their products, they can target such customers with their marketing efforts, and avoid wasting resources on those who are unlikely to be interested. In this way, banks can improve their profits. Data analysis goes in that direction through machine learning models that use data to predict outcomes. In our study, we demonstrate that banks can apply machine learning to customer data and accurately predict whether customers will subscribe to their term deposit products or not. Using our recommended Classification Tree model shall support banks in optimizing their marketing operations and increasing profits.

In our study, we used the dataset by Moro et al. [1], which has data on telemarketing (direct marketing campaigns by phone calls) operations of a Portuguese bank and has previously been studied to predict whether potential clients would be willing to invest in a specific product called term deposit. The dataset contains factors such as age, marital status, education level, history of credit default, the existence of a housing loan, the presence of a personal loan, the number of contacts performed during the campaign, etc. These act as predictor variables and will be assessed with all other variables in the dataset to analyze different machine learning models, resulting in determining the most accurate prediction.

This paper is divided into the following parts: Section II discusses previous research on the business problem. Section III provides information about the processing performed to clean the data, preparing it to be used in the different machine learning models. Section IV shows the application of each algorithm (Classification Tree, Linear Discriminant Analysis, Logistic Regression, K-NN, and Naïve Bayes) and the discussion of the results given by each one. Section V provides the conclusion of the analysis and our recommendations.

II. LITERATURE SURVEY

Moro et al. [2] discuss that the effect of marketing campaigns on the general public has decreased due to the growing number of such campaigns over time, and how Business Intelligence and Data Mining techniques can enhance their effectiveness. They initially collected the data from a Portuguese bank with the business goal of finding a model that could explain whether a client would subscribe to a term deposit or not. We have adopted the same business goal.

A subsequent publication by Moro et al. [3] applied Decision Trees (DT), Support Vector Machines (SVM), and Neural Networks (NN) to answer the same question. In that study, NN achieved the best result with an AUC score of 0.8. However, our study did not include NN or SVM, but we utilized DT. We have also set AUC as the main performance metric, besides test accuracy and TPR as secondary metrics. Importantly, the dataset has been refined significantly over time, so performance comparison is not possible with studies that used a different version of the dataset.

In one study [4], Chen et al. used the current dataset to address the same business problem with Logistic Regression (LR), NN, Random Forest (RF), and K-NN. Based on the set of three metrics – AUC, test accuracy, and FPR at TPR = 0.99 – NN was selected best overall, although RF had the highest AUC of 0.94. In our research, we achieved the highest AUC of 0.92 with DT, which we refer to as Classification Tree, and Logistic Regression. Overall, the Classification Tree was a better performer based on higher test accuracy and TPR. Performance differences between the two studies can be attributed to at least three factors. Firstly, Chen et al. used more sophisticated algorithms. For example, RF is an advanced version of DT and it performs better [5]. Secondly, to address the imbalance between the target classes, Chen et al. resampled the dataset differently for algorithms. For example, they oversampled the data for LR and used mixed sampling for RF, to fine-tune results. However, we balanced the dataset by conducting under-sampling and used a single resampled dataset in all our models for consistency and fair comparison. Lastly, for missing values, Chen et al. performed imputations and partial deletions while we chose to remove records with missing values.

In our study, we achieved a comparable level of classification performance with much simpler methodologies that are easier to understand.

III. DATA PRE-PROCESSING

We explored the data to understand it better. The dataset contained 41,188 records and 21 variables, which consisted of 11 categorical and 10 numeric variables, as described in Table 1 [1].

Table 1: Attribute Descriptions

Attribute	**Description**
age	Age of the customer (numeric)
Job	Type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur',' housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
marital	Marital status (categorical: 'divorced', 'married',' single', 'unknown')
education	Education level (categorical: 'basic.4y',' basic.6y',' basic.9y',' high.school', 'illiterate',' professional.course',' university.degree', 'unknown')
default	Has credit in default? (Categorical: 'no', 'yes', 'unknown')
housing	Has a housing loan? (Categorical: 'no', 'yes', 'unknown')
loan	Has a personal loan? (Categorical: 'no', 'yes', 'unknown')
contact	Contact communication type (categorical: 'cellular', 'telephone')
month	Last contact month of the year (categorical: 'Jan',…, 'Dec')
day_of_week	Last contact day of the week (categorical: 'Mon',…,' Fri')
duration	Last contact duration, in seconds (numeric).
campaign	Number of contacts performed during this campaign and for this client (numeric, includes the last contact)
pdays	Number of days that passed by after the client was last contacted from a previous campaign (numeric)
previous	Number of contacts performed before this campaign and for this client (numeric)
poutcome	The outcome of the previous marketing campaign (categorical: 'failure',' nonexistent', 'success')
emp.var.rate	Employment variation rate - quarterly indicator (numeric)
cons.price.idx	Consumer price index - monthly indicator (numeric)
cons.conf.idx	Consumer confidence index - monthly indicator (numeric)
euribor3m	Euro InterBank Offered Rate - daily indicator (numeric)
nr.employed	Number of employees - quarterly indicator (numeric)
y	Target: has the client subscribed to a term deposit? (Binary: 'yes',' no')

While exploring the data, the classes within the target variable were found to be highly imbalanced, as depicted in Figure 1.

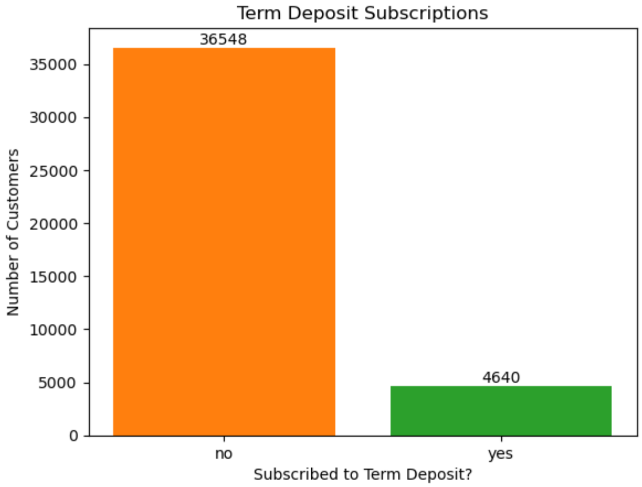

Figure 1. Class Imbalance in Target Variable

The ratio of non-subscribers ('no') and subscribers ('yes') was about 90:10, respectively. Therefore, balancing the classes was deemed important, to avoid the problem where a predictive model produced results that were disproportionately biased in favor of the majority class. However, data cleaning was required before that. Figure 2 depicts the distribution of missing values amongst the variables.

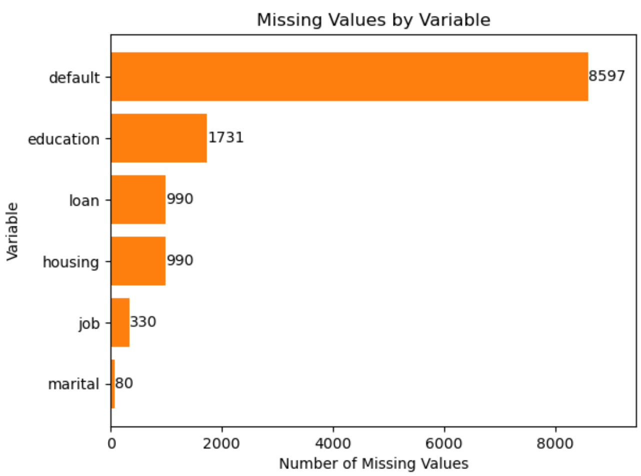

Figure 2. Missing Values by Variable

It was observed that in several cases it wasn't known if the customers had any credit default. Missing values were treated by omitting them from the dataset. Consequently, 10,700 records were dropped reducing the number of records to 30,488. The clean dataset was then balanced by under-sampling the majority ('no') class. For this purpose, the total number of records belonging to the 'yes' class was identified and an equal number of records from the 'no' class were randomly selected to form an equally balanced dataset as shown in Figure 3.

Figure 3. Class Balance After Resampling

Further data pre-processing, such as splitting of training and validation sets, conversion of categorical data to numerical values, rescaling of variables via normalization, etc., was done separately for machine learning algorithms before their training.

IV. MODEL TRAINING AND PERFORMANCE

Classification Tree

The motivation behind using the decision tree classifier was to discover decision rules that would help understand the data better. It would also provide insight into which variables were more important in terms of their predictive power. Table 2 shows the key parameters and their best values obtained via grid search.

Table 2: Classification Tree Parameters

Parameter	Value
max_depth	7
max_leaf_nodes	30
min_samples_leaf	20
min_samples_split	110

Since the goal was to have a predictive model that was understandable, the parameters that controlled the tree size, such as max-leaf nodes, were specified and then optimized to balance the model's complexity with performance. This produced a classification tree with a depth of 7 and an 'Area Under Curve' (AUC) score of 0.936 on the training data. For illustration purposes, a sample of the tree is shown in Figure 4 below.

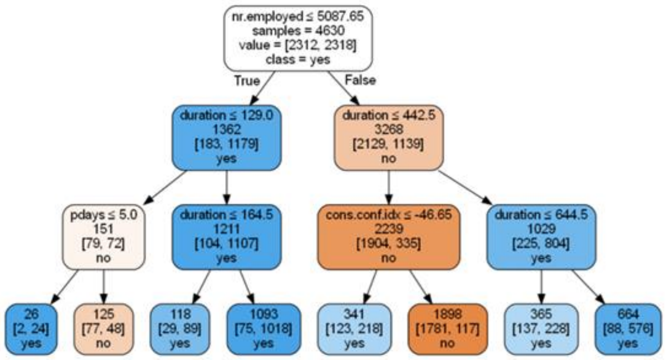

Figure 4. Classification Tree Diagram

The sample of the tree depicts that nr.employed (number of bank employees), duration (in seconds of the phone call with a customer), pdays (days since last contact), and cons.conf.idx (consumer confidence index) were important variables. However, the call duration and the consumer confidence index emerged as the variables that most effectively distinguished between the classes. Chen et al. [4] suggest that call duration could be a key factor as long conversations may indicate customers' interest in the product. According to the tree diagram, if a bank had a maximum of 5087.65 employees and the call with the customer lasted more than 129 seconds, then 87% of the time customers would invest in term deposits. On the other hand, if employees were above 5087.65, the call had to be more than 442.5 seconds to expect 78% of customers would subscribe. However, if the call was up to 442.5 seconds, there was only a 15% chance of a positive response. In that case, if the consumer confidence index was less than or equal to -46.65, the chances of a positive response increased to 69%. According to the Organization for Economic Co-operation and Development (OECD) [6] an index value of below 100 indicates a pessimistic attitude towards the economy and a tendency to save more. Therefore, a more negative value of the index suggests that customers would be looking for options to save money and would likely accept a term deposit offer.

The classification performance of the model is summarized in Table 3, Table 4, and Figure 5.

Table 3: Classification Report - Classification Tree

	precision	recall	f1-score
No	0.91	0.82	0.86
Yes	0.84	0.91	0.87

Table 4: Confusion Matrix - Classification Tree

Accuracy 0.87		Prediction	
		No	Yes
Actual	No	1274	273
	Yes	132	1409

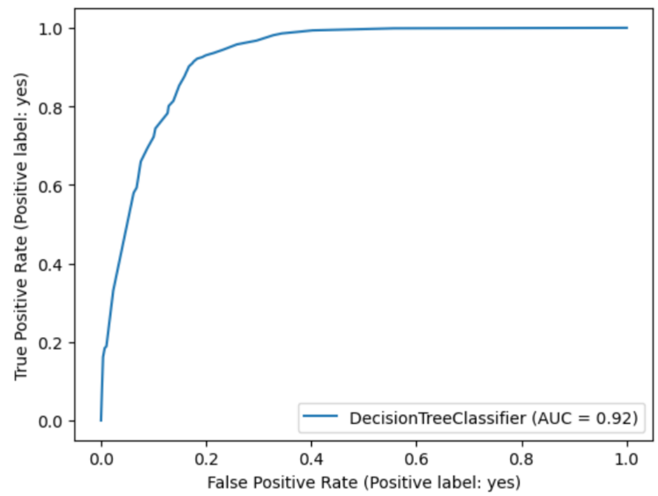

Figure 5. ROC Curve – Classification Tree

Overall, the model achieves all requirements: it facilitates understanding of decision rules but also performs well on predicting term deposit subscribers with a recall rate of 91% for the 'yes' class, an accuracy of 87%, and an AUC score of 0.92.

Linear Discriminant Analysis

Balakrishnama et al. [7] state that Linear Discriminant Analysis (LDA) is a technique that can be used for both data classification and dimensionality reduction. LDA can handle scenarios where the classes are unevenly distributed and have been tested with randomly generated data. The method increases the ratio of between-class variance to within-class variance, leading to better class separability and optimal performance on any dataset.

After running the LDA model to classify 'yes' and 'no' outcomes, a confusion matrix is used to illustrate the performance between prediction and actual result in Table 5.

Table 5: Confusion Matrix - LDA

Accuracy 0.83		Prediction	
		No	Yes
Actual	No	1303	244
	Yes	281	1260

Table 6: Classification Report - LDA

	precision	recall	f1-score
No	0.82	0.84	0.83
Yes	0.84	0.82	0.83

According to the Confusion Matrix of Linear Discriminant Analysis, there are 1,303 true negatives while the true positive is 1,260, representing the 'no' and 'yes' outcomes. In addition, the model has an accuracy of 83% in predicting the result above. On the other hand, the LDA model incorrectly predicted 525 out of 3,088 outcomes leading to a total error rate of 17%.

Precision is the ratio of true positives to the total number of predicted positives, while recall is the ratio of true positives to the total number of actual positives [8]. Based on the Classification Report of LDA, the precision and recall score for both 'no' and 'yes' outcomes are strongly balanced with 82% and 84%, achieving a higher score indicating the model is performing well.

Apart from the classification report, we can assess the classification performance of Linear Discriminant Analysis by plotting a ROC curve. The ROC curve represents the Receiver Operating Characteristic curve, which is a graphical representation of a classification model's performance across various classification thresholds [10]. The curve plots two key parameters, namely the True Positive Rate (TPR) and the False Positive Rate. A ROC curve that is closer to the top-left corner of the plot is indicative of a good classifier, as it suggests a high true positive rate and a low false positive rate.

Based on Figure 6, when the True Positive Rate (TPR) is 0.8, the model correctly identifies 80% of positive outcomes as 'yes'. However, at this threshold, the corresponding False Positive Rate (FPR) is 0.2, reflecting 20% of the 'no' outcomes that are incorrectly classified as 'yes'. As the threshold is increased closer to 1.0 (100%) for true positive identification, the false positive rate increases to approximately 35%.

Furthermore, the performance of the model can also be evaluated using the Area Under the Curve (AUC), which measures the degree of separability between the classes and reflects the model's capability to differentiate between them [9]. A higher AUC score signifies a better ability to accurately predict 0 as 0 and 1 as 1. This LDA model has an AUC score of 0.91 indicating a 91% likelihood of correctly distinguishing between 'yes' and 'no' outcomes.

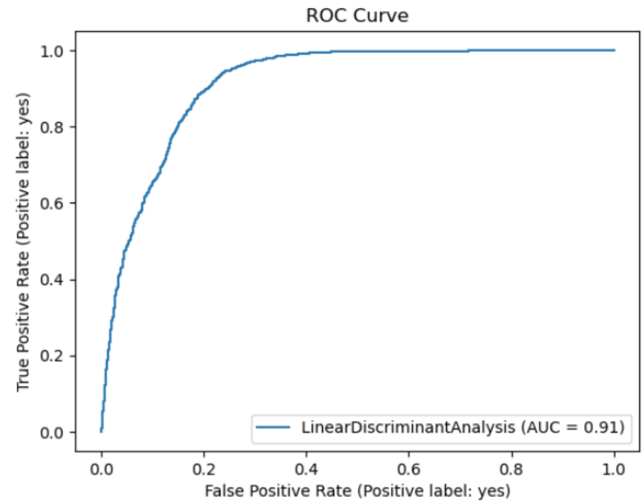

Figure 6. LDA - ROC Curve

Logistic Regression

In this case, the dataset was preprocessed by one-hot encoding of categorical features, scaling the data, and splitting it into training and testing sets. Hyperparameters were tuned using grid search, and the resulting best model was evaluated using accuracy score, classification report, confusion matrix, and ROC curve analysis.

Hyperparameter tuning is an essential step in developing a machine learning model that can generalize well to unseen data. In this code, GridSearchCV was used to find the best hyperparameters for the logistic regression model [11]. The hyperparameters tested were the penalty ('l1' or 'l2'), the inverse of regularization strength (C), the optimization solver ('liblinear'), and the maximum number of iterations taken for the solver to converge (max_iter). The best hyperparameters were found to be *C=1, penalty='l2', solver='liblinear', and max_iter=100.*

After tuning the hyperparameters, the best model was created and evaluated using various metrics. The accuracy score was 0.8510, which means that the model correctly classified 85.10% of the records. The classification report in Table 7 shows that the model performed similarly well for both the positive and negative classes, with precision, recall, and F1-score all between 0.84 and 0.87. This suggests that the model is well-balanced.

Table 7: Classification Report - Logistic Regression

	precision	recall	f1-score
No	0.86	0.84	0.85
Yes	0.84	0.87	0.85

The confusion matrix in Table 8 further confirms the model's performance, showing that the majority of class ‘no’ was correctly classified, with 1,293 true negatives, 1,335 true positives, 254 false positives, and 206 false negatives. The overall AUC score of the ROC curve is 92%, indicating that the model has good predictive power in distinguishing between the positive and negative classes.

Table 8: Confusion Matrix - Logistic Regression

Accuracy 0.85		**Prediction**	
		No	Yes
Actual	No	1,293	254
	Yes	206	1,335

Overall, this logistic regression model performs well in predicting whether customers will subscribe to term deposits or not, with an accuracy score of 85.10% and an AUC score of 92% (see Figure 7). The model's balanced performance, as indicated by the classification report, suggests that it is a reliable predictive model.

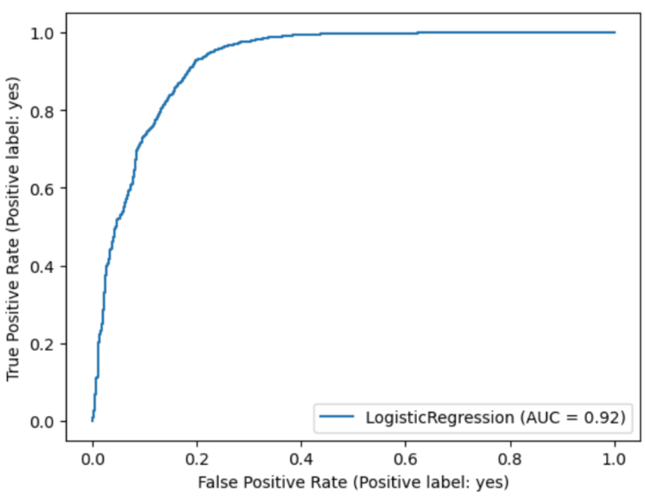

Figure 7. ROC Curve - Logistic Regression

K-Nearest Neighbors (K-NN)

One of the key points in the K-NN model is knowing the ideal number of "K" to use. This parameter is used to choose the number of nearest neighbors of the data being analyzed, and the model will determine its classification based on the biggest amount of a specific class among those nearest neighbors.

For the analysis done in this case, K was tested in a range of values (from 1 to 14) to check the best accuracy result. Table 9 below shows that K=9 had the best accuracy result. Therefore, K=9 was used to train the model.

Table 9: Finding Optimal K

k	**accuracy**
1	0.81347
2	0.79534
3	0.83841
4	0.82772
5	0.84132
6	0.83679
7	0.84294
8	0.84424
9	**0.84521**
10	0.84197
11	0.84488
12	0.84132
13	0.84165
14	0.84197

After training the model, it was tested on the validation set to get the classification predictions. The results are shown in Table 10 below.

Table 10: Confusion Matrix - K-NN

Accuracy 0.85		**Prediction**	
		No	Yes
Actual	No	1,262	285
	Yes	193	1,348

The above table shows that the model had an accuracy of 85%. Going further in the analysis of the confusion matrix, it has 1,348 true positives, which is the number of correct predictions that the client will invest in term deposits, and 285 false positives reflecting the number of incorrect predictions that the client will invest in term deposits. Also, there are 1,262 true negatives which show the number of correct predictions that the client will not invest in term deposits, and 193 false negatives which are the predicted number of clients that will not invest in term deposits, but in reality, they did. In addition, Figure 8 shows that the AUC is 91%, which is a high likelihood of correctly distinguishing between ‘yes’ and ‘no’ outcomes.

Table 11: Classification Report - K-NN

	precision	**recall**	**f1-score**
No	0.87	0.82	0.84
Yes	0.83	0.87	0.85

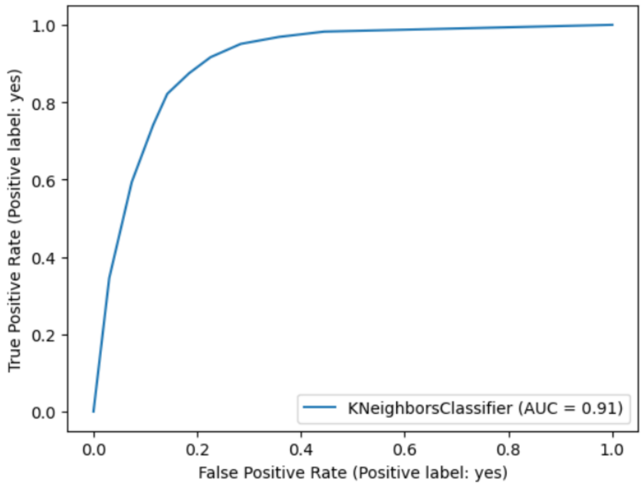

Figure 8. ROC Curve - K-NN

Naïve Bayes

Naïve Bayes is a categorical classification algorithm, often used as a benchmark against other classification algorithms [12]. That means it can be compared to other models to determine if they are useful or not.

The confusion matrix in Table 12 and classification report in Table 13 shows the model's performance on the validation set.

Table 12: Confusion Matrix - Naïve Bayes

Accuracy 0.71		**Prediction**	
		No	Yes
Actual	No	1,223	324
	Yes	557	984

Table 13: Classification Report - Naïve Bayes

	precision	**recall**	**f1-score**
No	0.69	0.79	0.74
Yes	0.75	0.64	0.69

Table 12 above shows a total of 3,088 predictions. There are 1,223 true negative responses, which reflect that the model correctly predicted clients that did not subscribe to term deposits (no). The 984 true positives (yes) are correct predictions for clients that actually subscribed to term deposits. The 557 false negatives are the incorrect predictions that clients will not subscribe to term deposits when in fact they did. The 324 false positives indicate incorrect predictions that a client will subscribe to a term deposit when in fact they do not. The accuracy of the models is 0.71, meaning that the overall accuracy of correctly predicting the outcome of a client's decision is 71%. Again, Naïve Bayes is a benchmark model, so other classification models need to have higher accuracy in order to be useful.

Figure 9 depicts the ROC curve. It is important to consider that the curve is closer to the upper-left corner reflecting a higher TPR, lower FPR and greater area under the curve (AUC), which measures the model's ability to distinguish between positive and negative cases [10]. The AUC is 77%, indicating a significantly better likelihood of correctly distinguishing between 'yes' and 'no' than random guessing. Other models in this study will require at least an AUC of 77% to be considered useful.

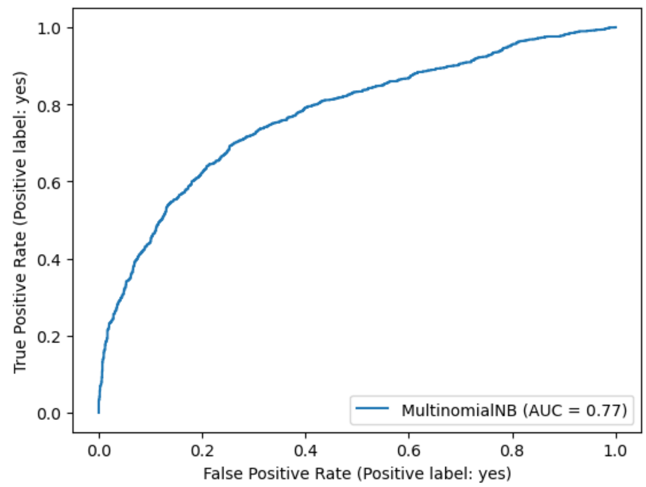

Figure 9. ROC Curve - Naïve Bayes

Although a key aspect of our research was to identify customers that will subscribe to term deposits (positive cases), it is important that our models can determine non-subscribers (negative cases) as well, for higher overall accuracy. The concept of cost-sensitive learning [13] is evident in the models as it determines that customers who do not subscribe to term deposits, the bank can direct the allocation of its telemarketing resources towards customers with a higher likelihood of subscribing.

V. CONCLUSION

In this paper, we applied and analyzed the effectiveness of five different machine learning algorithms – Classification Tree, Linear Discriminant Analysis, Logistic Regression, K-NN and Naïve Bayes – for predicting whether or not clients will subscribe to term deposits based on data collected from a Portuguese bank's telemarketing campaign. As per Table 14 and

Figure 10, our analysis shows that the Classification Tree algorithm performed the best with an overall accuracy of 87% and an AUC of 92% among other models.

Table 14: Model Performance Summary

Model	**Accuracy**	**AUC**
Linear Discriminant Analysis	0.83	0.91
K-NN	0.85	0.91
Naïve Bayes	0.71	0.77
Classification Tree	**0.87**	**0.92**
Logistic Regression	0.85	0.92

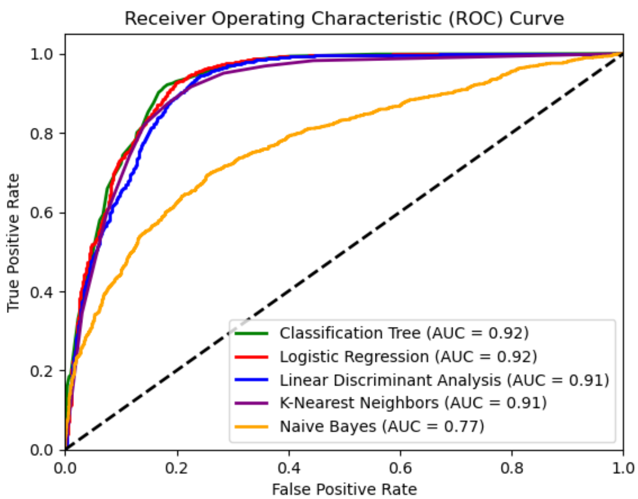

Figure 10. ROC Curve - All models

Our findings shall provide valuable insights to banks in designing effective marketing strategies to increase revenue while minimizing costs, therefore improving profits. By leveraging machine learning algorithms, banks can efficiently create effective market segmentations and identify potential clients who are more likely to subscribe to term deposits, thereby reducing their marketing costs and increasing their return on investment.

Future research can explore the effectiveness of other machine learning algorithms and feature engineering techniques to improve the accuracy of predictions. Additionally, incorporating the latest data and feedback from clients can enhance model performance and provide more accurate predictions for banks to make informed decisions.

REFERENCES

[1] S. Moro, P. Cortez, and P. Rita, "Bank Marketing Dataset," UCI Machine Learning Repository, 2014. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#

[2] S. Moro, R. Laureano, and P. Cortez, "Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology," in Proceedings of the European Simulation and Modelling Conference - ESM'2011, P. Novais et al. (Eds.), Guimaraes, Portugal, October, 2011, pp. 117-121.

[3] S. Moro, P. Cortez, and P. Rita, "A Data-Driven Approach to Predict the Success of Bank Telemarketing," Decision Support Systems, vol. 62, pp. 22-31, Jun. 2014.

[4] J. Chen, Y. Han, Z. Hu, Y. Lu, and M. Sun, "Who Will Subscribe A Term Deposit? " 2014. [Online]. Available: http://www.columbia.edu/~jc4133/ADA-Project.pdf

[5] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," International Journal of Computer Science Issues (IJCSI), vol. 9, no. 5, pp. 272, 2012.

[6] OECD, "Consumer Confidence Index (CCI) (Indicator)," doi: 10.1787/46434d78-en, accessed on 05 April 2023, https://data.oecd.org/leadind/consumer-confidence-index-cci.htm#

[7] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," Institute for Signal and Information Processing, vol. 18, pp. 1-8, 1998.

[8] Statology, "How to Create a Classification Report in Scikit-Learn," Feb. 11, 2021. [Online]. Available: https://www.statology.org/sklearn-classification-report/

[9] S. Narkhede, "Understanding AUC-ROC curve," Towards Data Science, May 21, 2018. [Online]. Available: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

[10] Hand, D.J., Till, R.J. A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Machine Learning 45, 171–186 (2001). https://doi.org/10.1023/A:1010920819831

[11] Scikit-learn, "3.2. Tuning the hyper-parameters of an estimator," [Online]. Available: https://scikit-learn.org/stable/modules/grid_search.html#tuning-the-hyper-parameters-of-an-estimator

[12] S. Yang, "Introduction to Naïve Bayes Classifier," Towards Data Science, 2019. [Online]. Available: https://towardsdatascience.com/introduction-to-naïve-bayes-classifier-fa59e3e24aaf

[13] Ling, C. X., & Sheng, V. S. (2008). Cost-sensitive learning and the class imbalance problem. In C. Sammut & G. I. Webb (Eds.), Encyclopedia of machine learning (pp. 231-235). Springer. https://doi.org/10.1007/978-0-387-30164-8_151