Bank Marketing Classification model to predict if a customer would subscribe a term deposit

Group 11: Deepali Baswa, Manoj Bandi, Adharsh Madhusudan, Rhea Ghadge

Abstract

In today's highly competitive business landscape, effective marketing strategies are crucial for companies to promote and sell their products or services. Marketing encompasses a wide range of activities, including advertising, selling, and delivering products to customers or businesses. The marketing data of the bank is primarily focused on phone calls made to potential clients to assess whether they would subscribe to the bank's term deposit or not. The analysis of this data can provide valuable insights into the effectiveness of the bank's marketing strategies and help identify areas for improvement. This report analyzes the marketing data of a Portuguese banking institution that is primarily based on phone calls.

1. Introduction

1.1 Problem Statement

The problem addressed in this project is a binary classification task, where the objective is to predict whether a potential client contacted through a marketing campaign will subscribe to a term deposit offered by a Portuguese banking institution. The goal is to analyze the bank's marketing data, which primarily involves phone calls, and gain insights into the effectiveness of their outreach strategies. The primary aim is to provide recommendations for improving the bank's marketing efforts, thereby increasing the likelihood of successful subscription to their term deposit product.

1.2 Motivation

We wanted to take this topic as it allowed us to understand customer behavior and preferences in the banking industry, which can be applied to other sectors as well. By identifying the factors that influence customers' decision-making, we wanted to help the bank improve its marketing strategies, to increase customer acquisition and revenue.

Therefore, by analyzing the bank's marketing data, we can gain valuable insights into customer behavior and preferences, help the bank improve its marketing strategies, and ultimately, enhance its customer satisfaction and revenue. The project presents an exciting opportunity to apply data analysis techniques to real-world marketing problems and provide actionable insights to the banking institution.

1.3 Dataset

The dataset contains information on the direct marketing campaigns of a Portuguese banking institution. It includes 41,188 instances and 20 attributes, primarily related to bank client attributes,

the last contact of the current campaign, and social and economic context attributes. The associated task is classification, with the output variable being whether or not the client subscribed to a term deposit (binary: "yes" or "no"). The dataset attributes include age, job type, marital status, education level, credit in default, housing and personal loans, contact communication type, and outcome of previous marketing campaigns, among others.

2. Related Work

2.1 Background search:

In the case of our project, we conducted the following background research:
Banking industry: We researched the banking industry in Portugal to gain a better understanding of the market and its trends.

Data sources: We identified and studied the Bank Marketing dataset, which contains information on the bank's phone-based marketing campaigns. We reviewed the data dictionary and the data structure to gain a better understanding of the variables and their meanings.

Data analysis techniques: We researched various data analysis techniques used in marketing research, such as logistic regression, decision trees, and clustering. We evaluated the applicability of each technique to our dataset and identified the most suitable method for our project.

By conducting thorough background research, we were able to gain a better understanding of the industry, the marketing campaigns, the available data sources, and the data analysis techniques.

2.2 Relevant Project and works:

In the field of marketing data analysis, there have been several related works that have explored similar research questions and methodologies. Some of these works include:

"Customer segmentation in the banking industry using data mining techniques" by C. Romero and F. Ventura: This study used data mining techniques to segment bank customers based on their behavior, providing insights into their preferences and decision-making processes.

"Marketing analytics for customer acquisition: A case study in the banking industry" by A. Tripathi and M. Madaan: This study explored the use of marketing analytics for customer acquisition in the banking industry, providing insights into the factors that contribute to successful marketing campaigns.

These related works provide a foundation for our project and demonstrate the usefulness of data analysis techniques in understanding customer behavior and improving marketing strategies in the banking industry.

3. Methods

3.1 Logistic Regression - Baseline Model

Logistic Regression is a statistical method used for predicting binary outcomes, in which the response variable can take only two possible values. It is a widely used technique for classification problems in various fields, including business, finance, healthcare, and engineering.

In Logistic Regression, we use a transformation function called the sigmoid function, which maps any real-valued number to a probability between 0 and 1. The sigmoid function is expressed as:

$$\sigma(z) = 1 / (1 + e^{(-z)})$$

where z is a linear combination of the input variables and their respective coefficients. The sigmoid function ensures that the predicted probabilities are bounded between 0 and 1, which makes it suitable for binary classification problems.

The goal of Logistic Regression is to estimate the optimal set of coefficients that maximize the likelihood of the observed data. We can obtain these estimates using the maximum likelihood estimation method or gradient descent optimization. The likelihood function measures the probability of observing the data given the model parameters, and the maximum likelihood estimator maximizes this probability. Alternatively, the negative log-likelihood function is minimized using gradient descent optimization to estimate the parameters.

We chose logistic regression as our baseline model due to its interpretability and widespread use in binary classification problems.

3.2 KNN Model

K-Nearest Neighbor (KNN) is a non-parametric supervised learning method that can be used for both regression and classification problems. It is called a lazy learner algorithm because it does not immediately learn from the training set. Instead, it stores the data during the training time and does not perform any calculations. When it gets new data, it builds the model and classifies the data into a class that is similar to the new data. It works by identifying which data points are closest to a given query point. To do this, it uses distance metrics such as Euclidean distance, which measures the length of a line segment between two points. Decision boundaries are formed based on the distance metrics, which divide the query points into different areas.

In the context of the bank marketing dataset, KNN can be used as a classification algorithm to predict whether a potential client will subscribe to the bank's term deposit product. The algorithm works by first calculating the distance between the new data point (i.e., the potential client) and the existing data points (i.e., the clients in the training set).

The KNN algorithm then selects the subset of the training set that contains the k training samples closest to the new data point. The class of the new data point is then determined by taking the

majority vote of the class labels of the k nearest neighbors. For example, if the k nearest neighbors of a potential client are all labeled as having subscribed to the term deposit, then the algorithm would predict that this new client is likely to subscribe as well. The Euclidean distance can be used as the distance metric to calculate the distance between the new data point and the existing data points. By using the KNN algorithm in the bank marketing dataset, we can gain insights into which variables are most important in predicting whether a potential client will subscribe to the term deposit product.

3.3 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) works by finding a hyperplane in a high-dimensional space that separates the data into different classes. In the case of binary classification, the hyperplane divides the data into two classes: those that will subscribe to a term deposit and those that will not. The hyperplane is chosen based on the maximum margin, which is the distance between the hyperplane and the nearest data points from each class.

In practice, the data may not be linearly separable, which means that a straight line cannot divide the data into two classes. To handle this, SVMs use slack variables to allow for some misclassification while still trying to maximize the margin. This is called a soft margin classifier. The optimal hyperplane is found by solving a quadratic optimization problem.

To extend SVMs to nonlinear problems, a kernel function is used to transform the input data into a higher-dimensional space. In the bank marketing case, a radial basis function (RBF) kernel can be used to map the data into a higher-dimensional space where it can be separated by a hyperplane. The SVM algorithm will find the optimal hyperplane that maximizes the margin between the two classes of customers - those who subscribed and those who did not. The hyperplane can be linear or non-linear depending on the choice of kernel. Once the model is trained, it can be used to predict whether a new customer is likely to subscribe to a term deposit or not based on their attributes.

3.4 Random Forest Classifier

Random Forest is a machine learning algorithm that is used for both regression and classification tasks. A prediction is made using an ensemble learning technique that integrates different decision trees. The Random Forest approach builds a forest of decision trees for classification problems, where each tree is trained on a different portion of the input data and a random subset of the characteristics. The ultimate prediction is established by counting the votes cast by the majority of the forest's individual trees.

Random Forest has several advantages that make it a good fit for the Bank Marketing classification dataset as it can handle high dimensionality and noisy data. The Bank Marketing dataset is noisy at times and includes a lot of data instances. This type of data can be handled by Random Forest successfully since it may choose a subset of features for each individual tree, which lessens the influence of unnecessary features. Random Forest also reduces overfitting. Insufficient generalization to new data is the outcome of overfitting, which happens when a model is very

complicated and matches the training data too closely. By building numerous trees that are trained on various subsets of the data and then merging them to create predictions, Random Forest helps to reduce overfitting.

Overall, Random Forest is a powerful algorithm that can handle high-dimensional, noisy data and can provide insights into the importance of different features. These characteristics make it a good fit for the Bank Marketing classification dataset.

3.5 Gradient Boosting Classifier

Gradient Boosting is a popular machine learning algorithm used for both regression and classification problems. It is an ensemble learning method that builds a strong predictive model by combining multiple weak predictive models.

In Gradient Boosting, each new model in the ensemble is trained to minimize the error of the previous model. The key idea is to sequentially add new models to the ensemble, and adjust the weights of each instance in the training data to emphasize those that the previous models failed to predict accurately.

Gradient Boosting is particularly well-suited for classification problems, as it is able to handle both binary and multi-class classification tasks. It is also able to handle non-linear relationships between the features and the target variable, which can be useful when dealing with complex datasets.

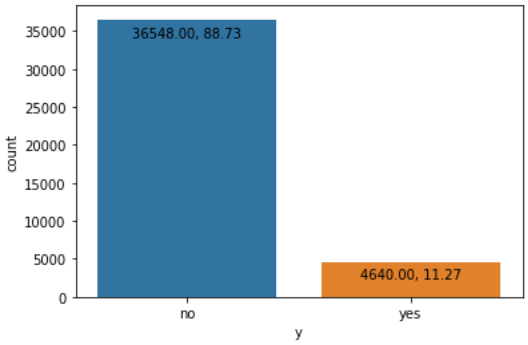
In the case of a bank marketing classification dataset, Gradient Boosting could be a good fit because the data is likely to be complex and have non-linear relationships between the features and the target variable (i.e., whether a customer will subscribe to a term deposit or not). The dataset may also have class imbalance, with a relatively small proportion of positive examples (customers who subscribed to a term deposit) compared to negative examples (customers who did not subscribe). Gradient Boosting can be particularly effective in addressing these challenges, by handling non-linear relationships and by weighting the training examples to ensure that the model learns to predict the positive examples as well as the negative examples. Additionally, Gradient Boosting is known to perform well on a wide range of classification tasks, making it a good choice for many types of data.

4. Experimental Results

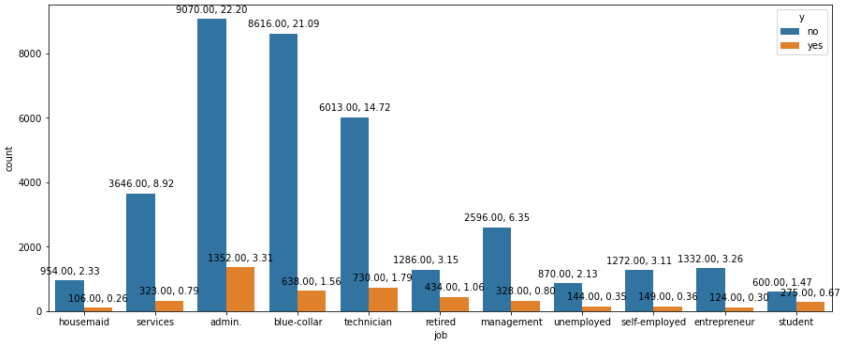
4.1 Exploratory Data Analysis

Univariate Analysis and Segmented Univariate Analysis

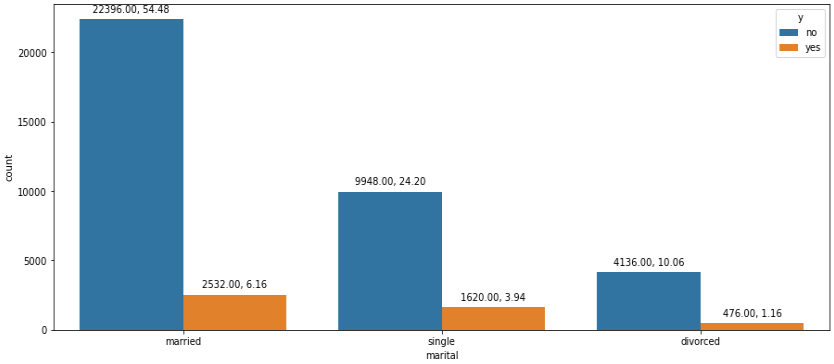
a. Class Distribution: We first saw the distribution of positive and negative classes across our dataset to see if our data was balanced or unbalanced.



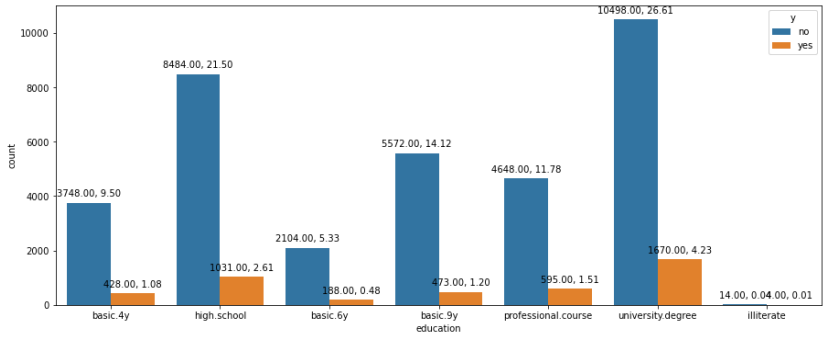
b. Categorical Variables: We compared the distributions of the two classes across different occupations i.e. Job, Marital Status, Education, Housing, Current Loan etc.
Job:



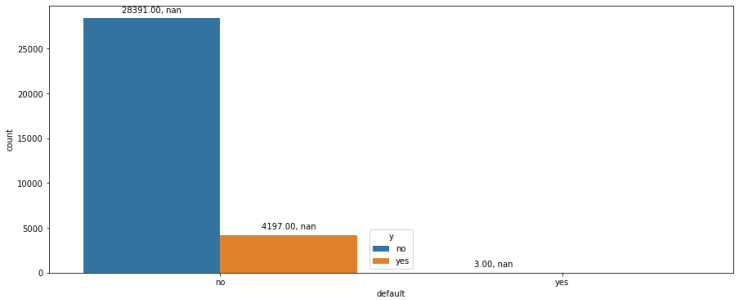
Marital:



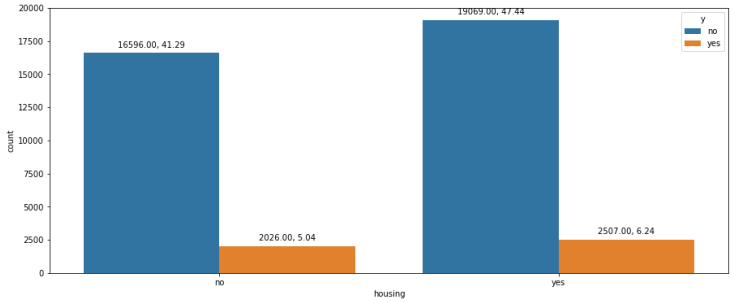
Education:



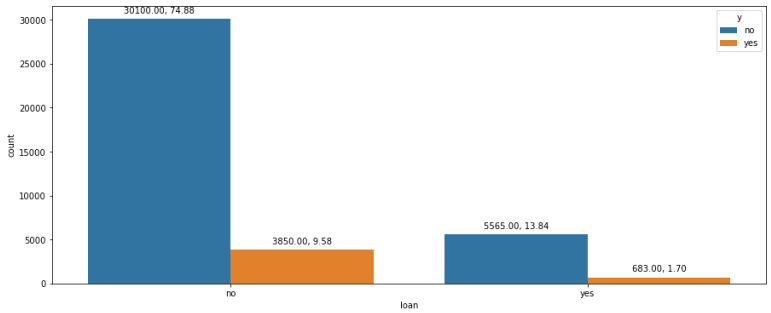
Default:



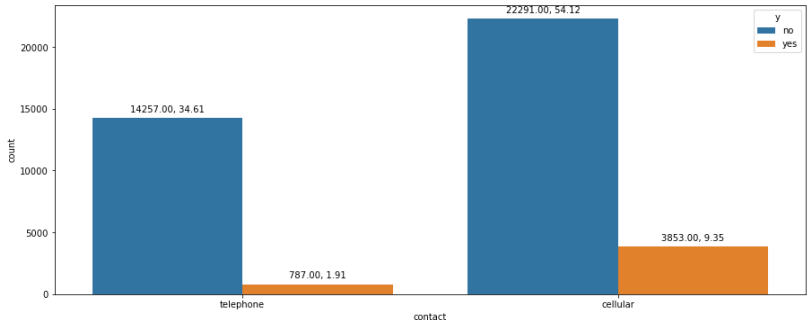
Housing:



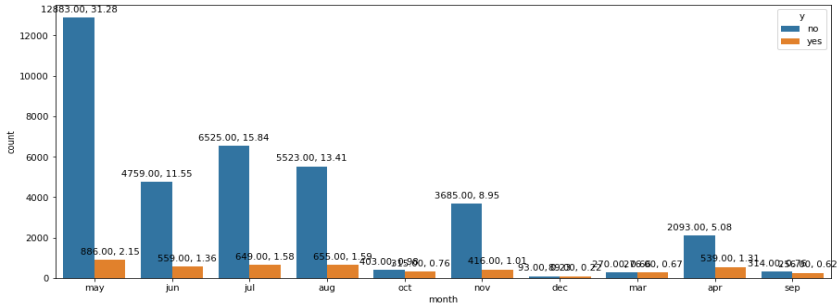
Loan:



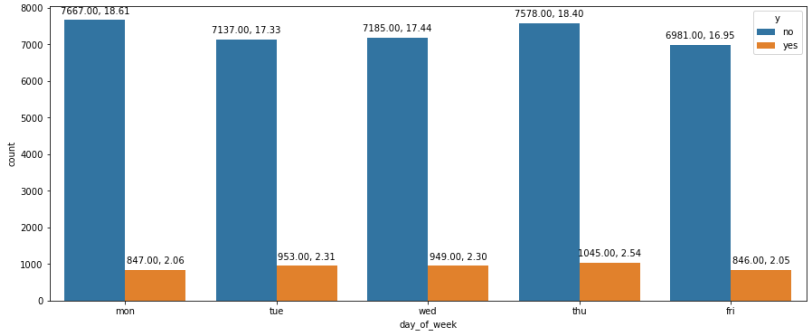
Contact:



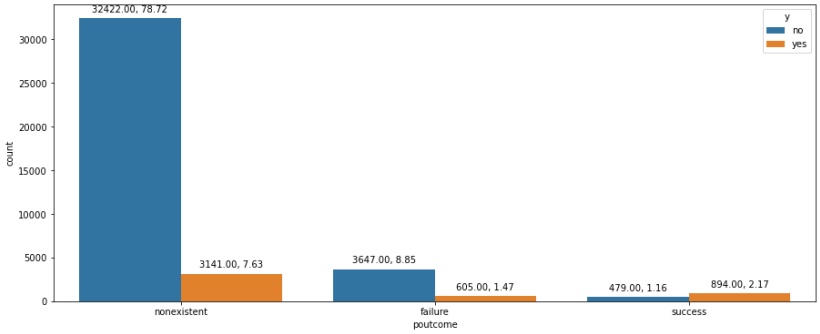
Month:



Day of the week:



P outcome:



4.2 Insights from categorical variables (based on univariate analysis)

1. Job: Highest Number (around 25%) of applications are from the admin type of job.

2. Default: Default variable has no impact on the client subscribing for term deposit. As we can see with no input, the client took the term deposit and the client having credit is not taking a term deposit. So we will drop this feature.

3. Marital: Around 60% of clients who were approached were married.

4. Education: Clients with university degrees and high school were approached more as compared to others and they have a higher success rate as well. (in terms of term deposit number)

5. Housing: Housing loan does not have much effect on the number of term deposits purchased.

6. Loan: We approach around 84% of clients with not having a personal loan.

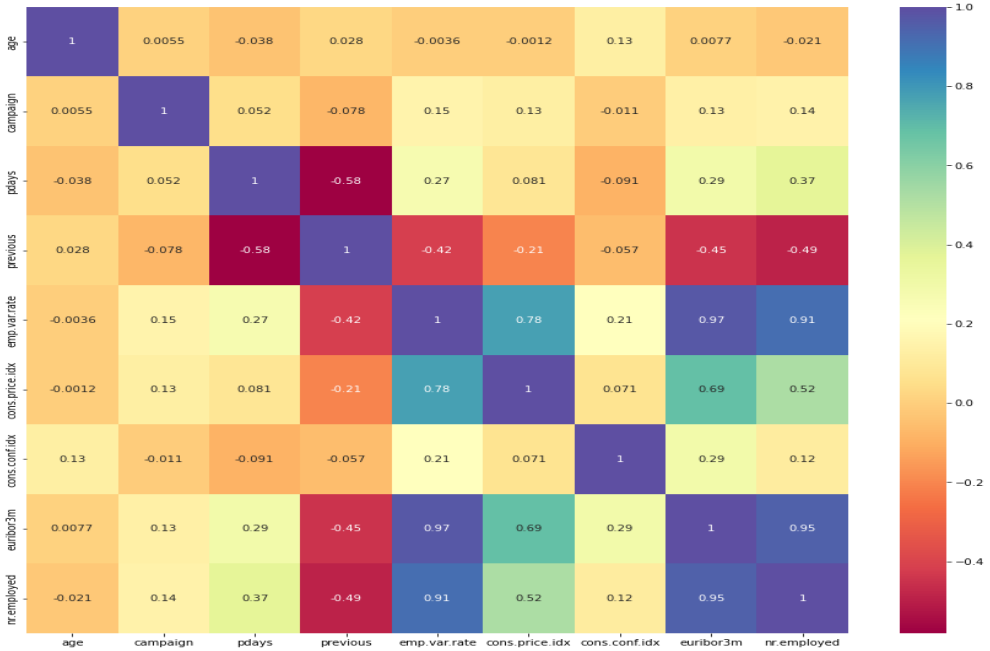
7. Contact: Around 64% calls are from cellular.

8. Month: Around 33% were approached in may and in January, February we don't have data or no one was approached. Success rate was almost the same in June, July and August.

9. day_of_week: We have 5 days collected values. There is no significant difference in the number of clients approached and number of people subscribed.

10. P outcome: If a client took the term deposit last time then there is higher chances of that client subscribing to it again.

Numerical Values: To identify which attributes are highly correlated and affect the target variable.



Positive high correlation between:

- emp.var.rate' and 'nr.employed'
- emp.var.rate' and 'euribor3m
- euribor3m' and 'nr.employed'

Note : Euribor is the acronym for the Euro Interbank Offered Rate. This is the interest rate at which credit institutions lend money to each other, which is often referred to as **"the price of money"**.

With this we can say that 'emp.var.rate' (employment variation rate) and 'nr.employed' (number of employees) are positively correlated with euribor. So we will drop **'emp.var.rate'** and **'nr.employed'** as **'euribor'** and also give us the price of money in current market.

	cons.price.idx	previous	cons.conf.idx	euribor3m	age	pdays	campaign
count	38245.000000	38245.000000	38245.000000	38245.000000	38245.000000	38245.000000	38245.000000
mean	93.570313	0.170009	-40.541164	3.623298	39.860871	963.531651	2.566662
std	0.576367	0.487169	4.623200	1.730226	10.289488	184.295254	2.767473
min	92.201000	0.000000	-50.800000	0.634000	17.000000	0.000000	1.000000
25%	93.075000	0.000000	-42.700000	1.344000	32.000000	999.000000	1.000000
50%	93.444000	0.000000	-41.800000	4.857000	38.000000	999.000000	2.000000
75%	93.994000	0.000000	-36.400000	4.961000	47.000000	999.000000	3.000000
max	94.767000	7.000000	-26.900000	5.045000	98.000000	999.000000	43.000000

4.3 Insights from continuous variables

1. Campaign: If the number of contacts performed during this campaign and for this client become more than 23 then there is a very high possibility that the client will not subscribe for a term deposit. Even if we contact a person more than 8 times, the probability is still low. **We will drop this column as it will not be known beforehand. But it is important to see that we should not contact any client more than 8 times during any campaign.**

2. Consumer price index: If this value is high then the probability of the client not subscribing is slightly higher.

3. Previous: If we contact a client before a campaign then there are high chances that client will subscribe. We converted all the values above 2 to 2 based on the given data
4. Euribor 3 month rate: If this rate is high there is high chances of clients not subscribing to term deposit.
5. Pdays: If we start contacting clients 1 month before the campaign there is a high probability of that client subscribing.

4.4 Hot one encoding:

Hot one encoding is a technique used to convert categorical data into a format that can be used by machine learning algorithms. In the bank marketing dataset, there are several categorical variables such as contact, poutcome, job, month, marital, and education.

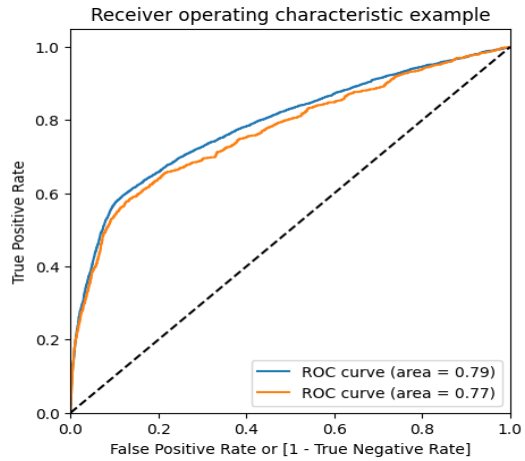
The dummy variable encoding is used to convert these categorical variables into numerical variables that can be processed by machine learning algorithms. It involves creating a new binary variable for each category in the original variable.

The function get_dummies from the pandas library is used to get the dummy variables for the categorical variables in the bank marketing dataset. In this code, the function is applied to the columns 'contact', 'poutcome', 'job', 'month', 'marital', and 'education'.

To avoid the dummy variable trap, which occurs when there is perfect multicollinearity among the dummy variables, the first column is dropped using the drop_first = True parameter. This first column can be derived using the other columns, so it is redundant and not necessary for the analysis.

5. Model Results

5.1 Logistic Regression




```
[ ]  print(classification_report(y_test, y_test_pred_lr))
```

	precision	recall	f1-score	support
0	0.91	0.98	0.95	6843
1	0.63	0.22	0.33	806
accuracy			0.90	7649
macro avg	0.77	0.60	0.64	7649
weighted avg	0.88	0.90	0.88	7649

The precision score of 0.91 for class 0 indicates that the model correctly identified 91% of the predicted instances of class 0. The recall score of 0.98 means that the model was able to identify 98% of the actual instances of class 0. The F1-score of 0.95 is the harmonic mean of precision

and recall for class 0. This suggests that the model performs well in identifying customers who did not subscribe to the term deposit.

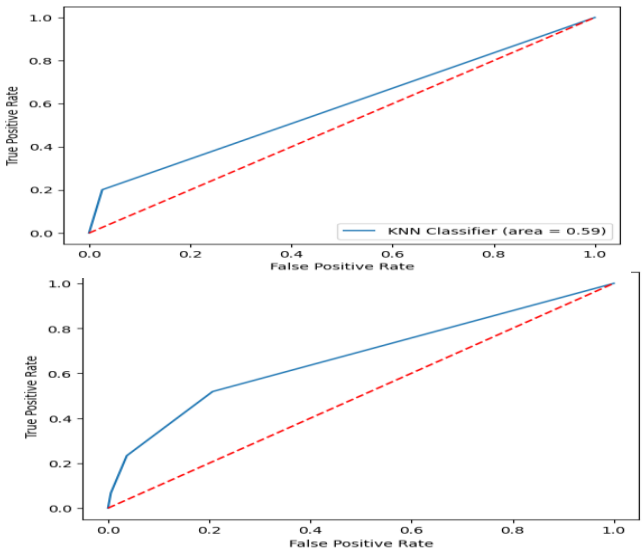
However, for class 1, the precision score of 0.62 indicates that the model correctly identified only 62% of the predicted instances of class 1. The recall score of 0.24 means that the model was only able to identify 24% of the actual instances of class 1. The F1-score of 0.35 is the harmonic mean of precision and recall for class 1. This suggests that the model struggles to identify customers who did subscribe to the term deposit.

The accuracy score of the model is 0.90, which indicates that the model correctly predicted the class label for 90% of instances in the dataset.

The macro-average F1-score is 0.65, which is the average of the F1-scores for each class label. The weighted-average F1-score is 0.89, which is the average of the F1-scores weighted by the number of instances in each class. These scores show that the model performs better for class 0 than for class 1.

In summary, the logistic regression model performs well in identifying customers who did not subscribe to the term deposit but struggles to identify customers who did subscribe. The precision and recall scores for class 1 are much lower than for class 0, indicating that the model is not able to distinguish between these customers as well.

5.2 KNN




```
Classification report:
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      6843
           1       0.48      0.20      0.28       806

    accuracy                           0.89      7649
   macro avg       0.69      0.59      0.61      7649
weighted avg       0.87      0.89      0.87      7649

Confusion matrix:
 [[6664  179]
 [ 644  162]]
```

The precision score of 0.91 for class 0 (the customers who did not subscribe) means that out of all the predicted instances of class 0, 91% were actually correct. The recall score of 0.97 means that out of all the actual instances of class 0, 97% were correctly predicted by the model. The F1-score of 0.94 is the harmonic mean of precision and recall for class 0.

For class 1 (the customers who did subscribe), the precision score is 0.48, which means that out of

all the predicted instances of class 1, only 48% were actually correct. The recall score of 0.20 means that out of all the actual instances of class 1, only 20% were correctly predicted by the model. The F1-score of 0.28 is the harmonic mean of precision and recall for class 1. The macro-average F1-score is 0.61, which is the average of the F1-scores for each class label. The weighted-average F1-score is 0.87, which is the average of the F1-scores weighted by the number of instances in each class.

Overall, the model performs well in predicting customers who did not subscribe to the term deposit (class 0) but struggles to predict customers who did subscribe (class 1). The precision and recall scores for class 1 are much lower than for class 0, indicating that the model is not able to distinguish between these customers as well. The confusion matrix shows that the model incorrectly predicted 644 instances of class 1 as class 0

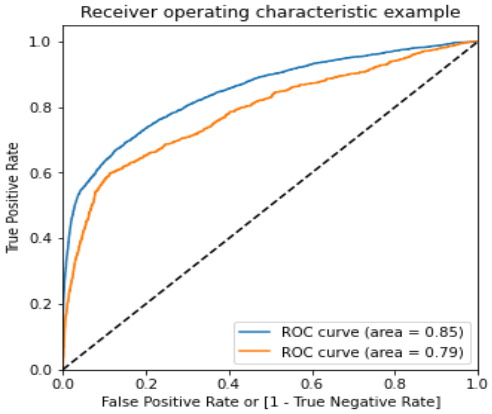
5.3 SVM

```
Classification Report:
              precision    recall  f1-score

           0       0.91      0.99      0.95
           1       0.66      0.20      0.31

    accuracy                           0.90
   macro avg       0.78      0.59      0.63
weighted avg       0.88      0.90      0.87

Accuracy:  0.8985999838148417
```

The SVM model performs well in predicting instances of class 0 with high precision and recall, but not as well for instances of class 1, as seen from the low recall and F1-scores. The accuracy score of the model is 0.8986, which means that it correctly predicted the class label for 89.86% of instances in the dataset. The macro-average F1-score is 0.63, and the weighted-average F1-score is 0.87, indicating that the model performs better on class 0 due to its higher number of instances in the dataset.
The macro-average F1-score is 0.63, which is the average of the F1-scores for each class label. The weighted-average F1-score is 0.87, which is the average of the F1-scores weighted by the number of instances in each class.

In summary, the SVM classifier performs well in predicting customers who did not subscribe to the term deposit, but has a lower performance in predicting customers who did subscribe, as indicated by the lower precision, recall, and F1-score for class 1.

5.4 RFC



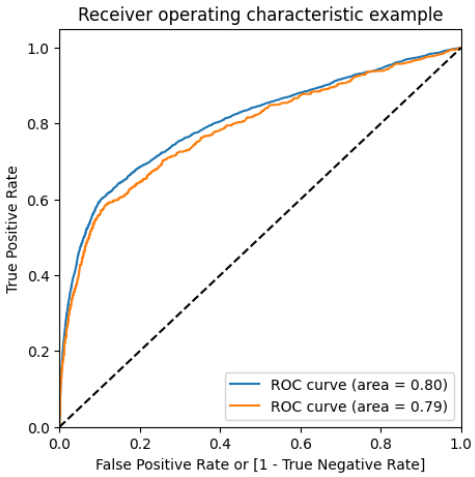

```
[ ]  print(classification_report(y_test,  y_test_pred_rf))
```

	precision	recall	f1-score	support
0	0.92	0.98	0.95	6843
1	0.62	0.24	0.35	806
accuracy			0.90	7649
macro avg	0.77	0.61	0.65	7649
weighted avg	0.89	0.90	0.89	7649

An accuracy of 0.904 indicates It is a good sign that the accuracy of the testing data is similar to the accuracy of the training data, as this indicates that the model is generalizing well to new data. An AUC of 0.85 means that the model has an 85% chance of correctly ranking a random positive example higher than a random negative example on the training data.An AUC of 0.79 means that the chance of correctly ranking a random positive example higher than a random negative example on the testing data. It is a good sign that the AUC of the testing data is not much lower than the AUC of the training data, as this suggests that the model is not overfitting to the training data.

Overall, the Random Forest model has a high accuracy on both training and testing data, which is a good indication that it is performing well. The AUC scores are also relatively high, indicating that the model is effective at distinguishing between positive and negative examples. However, the AUC score of the testing data is lower than that of the training data, which suggests that there may be some overfitting. Further evaluation and tweaking of the model may be necessary to improve its performance.

5.5 Gradient Boosting



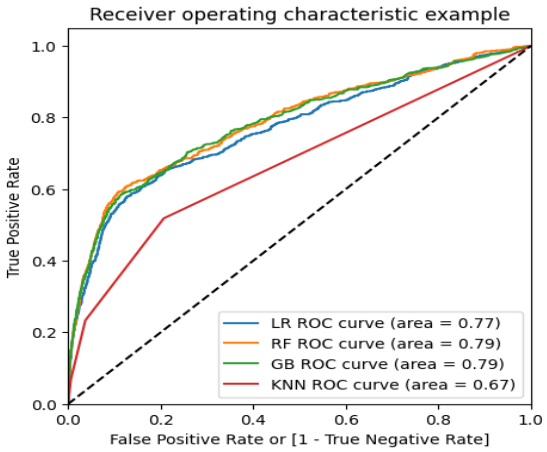

```
[ ]  print(classification_report(y_test, y_test_pred_gb))
```

	precision	recall	f1-score	support
0	0.92	0.99	0.95	6843
1	0.65	0.22	0.33	806
accuracy			0.91	7649
macro avg	0.78	0.60	0.64	7649
weighted avg	0.89	0.91	0.88	7649

The accuracy of the training data is 90.3%, which means that the model is able to correctly classify most of the examples in the training set. The accuracy of the testing data is slightly higher at 90.5%, indicating that the model is generalizing fairly well to new, unseen examples. An AUC value of 0.80 for the training data and 0.79 for the testing data suggests that the model is able to distinguish between the positive and negative classes reasonably well.

However, the fact that the training accuracy is higher than the testing accuracy and the AUC of the training data is higher than the AUC of the testing data suggests that the model may be overfitting to the training data. This means that the model may be fitting the noise in the training data rather than the underlying pattern, which could lead to poor performance on new, unseen data. To address this issue, the model may need to be regularized or the dataset may need to be expanded with more diverse examples.

6. Model Comparison



	Logistic Regression (LR)	Random Forest (RF)	Gradient Boosting (GB)	KNN	SVM
Accuracy of Training Data	89.9	91.7	90.2	89.6	90.07
Accuracy of Testing Data	90.48	90.41	90.57	90.4	89.85
ROC Curve Area	0.77	0.79	0.79	0.67	

ROC (Receiver Operating Characteristic) Curve is a widely used metric for evaluating the performance of binary classification models. The ROC curve is a plot of the true positive rate (TPR) against the false positive rate (FPR) for different classification thresholds. The TPR is the proportion of actual positive cases that are correctly identified as positive by the model, while the FPR is the proportion of actual negative cases that are incorrectly identified as positive by the model.

The values above represent each model's area under the ROC curve (AUC), which measures how effectively the model performed overall in classifying between positive and negative classes. The AUC values range from 0 to 1, with 1 denoting the best model and 0.5 denoting the random model. When comparing the two models, we can see that Random Forest has the higher AUC value, which is 0.79 and Logistic Regression, which has an AUC value of 0.77. This shows that, among the two models, Random Forest performs the best overall.

But it's vital to keep in mind that the AUC values don't tell the entire narrative; depending on the specific problem and the trade-offs between false positives and false negatives, it's crucial to take into account other metrics such as precision and recall.

Based on their AUC values, we could conclude that Random Forest is the better option among the two models, however the optimal model ultimately relies on the specific problem at hand and the business requirements.