Marketing for the Banking System

1 Introduction

The modern banking landscape is evolving rapidly, and marketing strategies play a pivotal role in shaping the success of financial institutions. In this project, we delve into the realm of direct marketing campaigns conducted by a prominent Portuguese banking institution. The primary focus is on predicting whether a client will subscribe to a term deposit or not, utilizing a classification approach.

Objective

The overarching goal of this project is to harness the power of data-driven decision-making in the realm of banking marketing. By leveraging machine learning classification techniques, we aim to develop predictive models capable of discerning whether a client will subscribe to a term deposit based on a variety of features. The project is structured into distinct steps, each contributing to the overall understanding and effectiveness of the models.

Dataset Overview

The dataset provided for this project consists of two key components: a training set (trainset.csv) and a test set (testset.csv). The target variable, "Subscribed," serves as the focal point for our classification task, possessing binary values – 'yes' and 'no.' The features encompass a diverse range, including demographic information, employment details, contact preferences, and historical engagement with previous marketing campaigns.

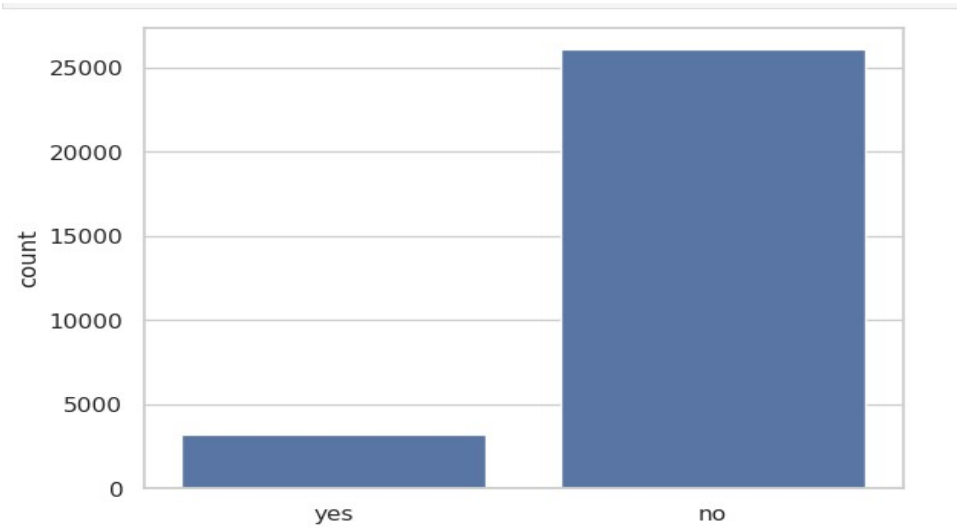
Steps

The steps involved in this project are

- Data Exploration
- Data Preprocessing
- Classification Learning Methods
- Model Testing And Reporting

2 Data Exploration

First of all I calculated the percentage of subscription in our classes and the result shown in figure



percentage of no subscription is 89.08134330907724
percentage of subscription is 10.918656690922756

Our classes are imbalanced, and the ratio of non subscription to subscription instances is 89/11. Also if we calculate the average of different features with respect to “Subscribed” feature, our calculations are

Subscribed	age	duration	campaign	pdays	nr.employed
no	40.186232	221.810086	2.807900	997.893538	5213.452004
yes	39.635795	629.597309	2.163642	909.438673	5139.377034

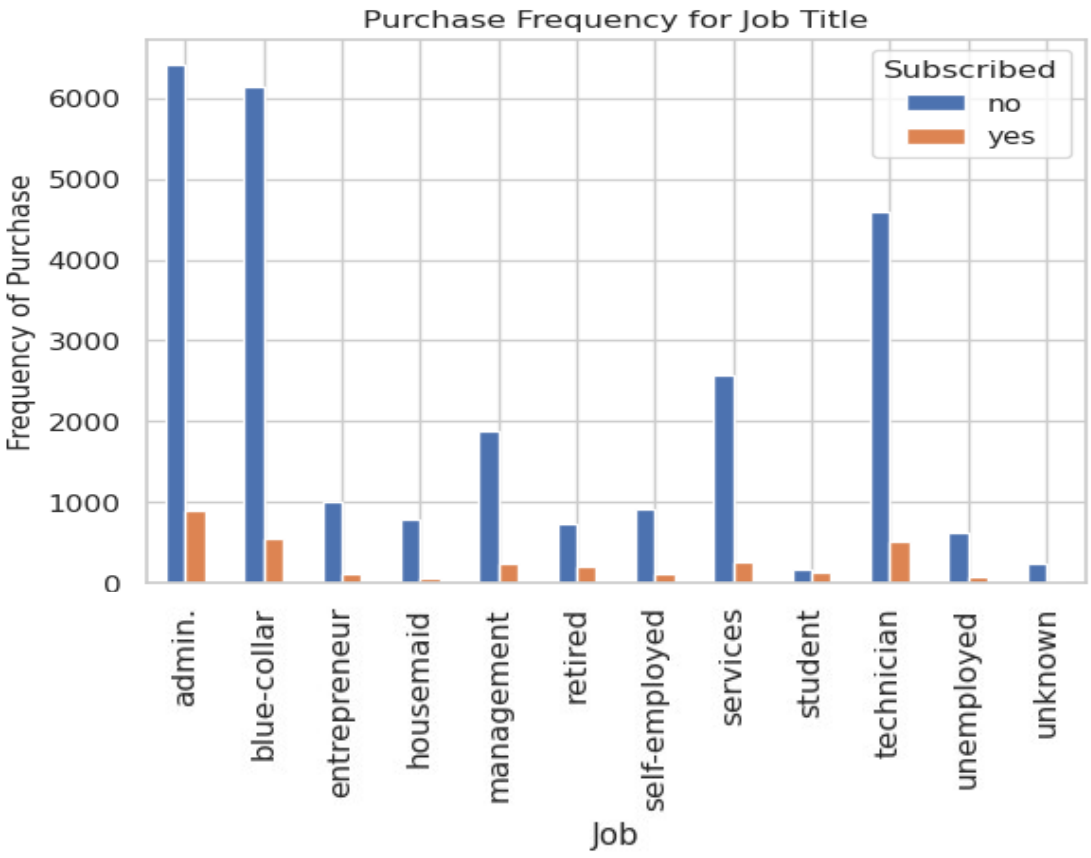
Mean of different features

Some of the observations taken from the above figure is

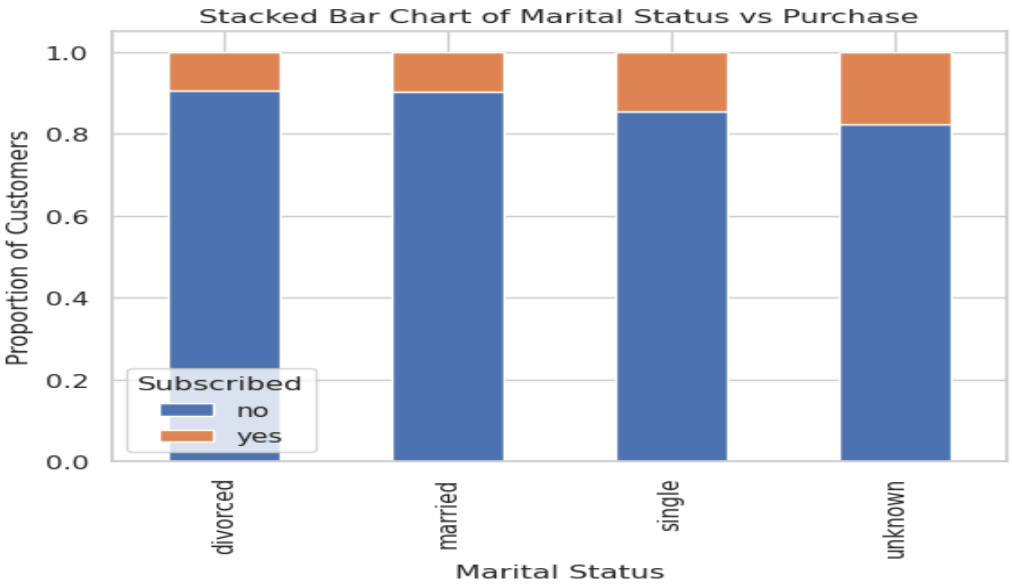
- The average age of the customers who bought the term deposit is higher than that of the customers who did not
- The pdays (days since the customer was last contacted) is understandably lower for the customers who bought it. The lower the pdays, the better the memory of the last call and hence the better chances of a sale.
- Surprisingly, campaigns (number of contacts or calls made during the current campaign) are lower for customers who bought the term deposit.

We can calculate categorical means for other categorical variables such as education and marital status to get a more detailed sense of our data.

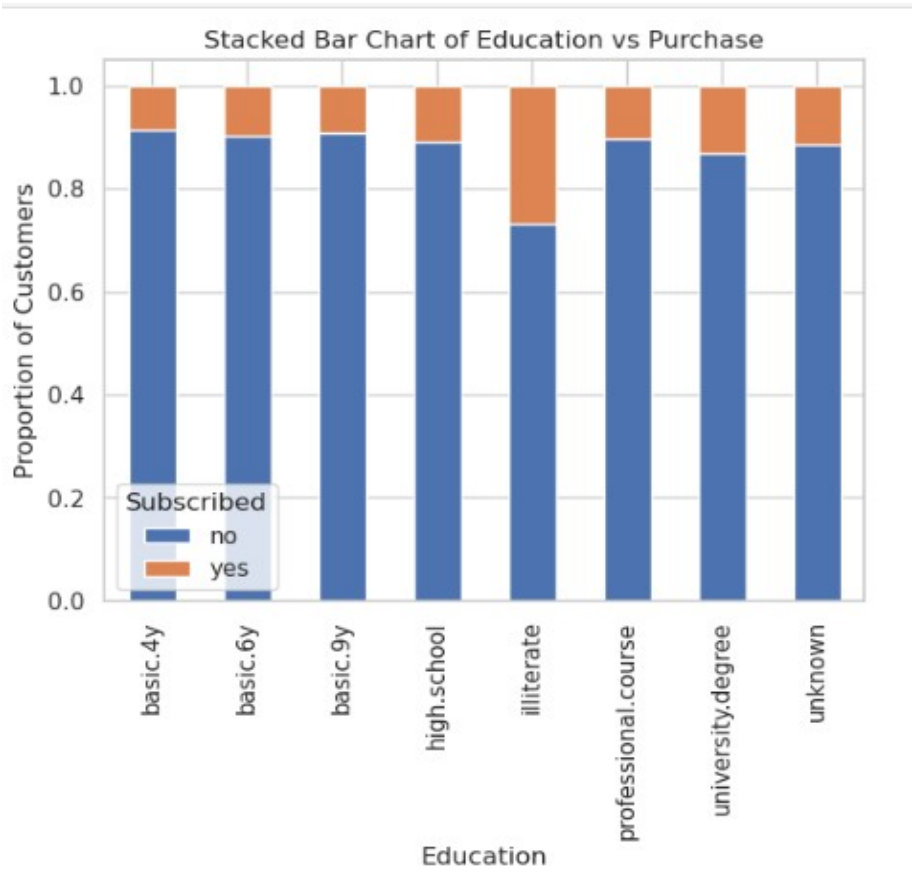
The frequency of purchase of the deposit depends a great deal on the job title. Thus, the job title can be a good predictor of the outcome variable. The statistical data from the job feature is



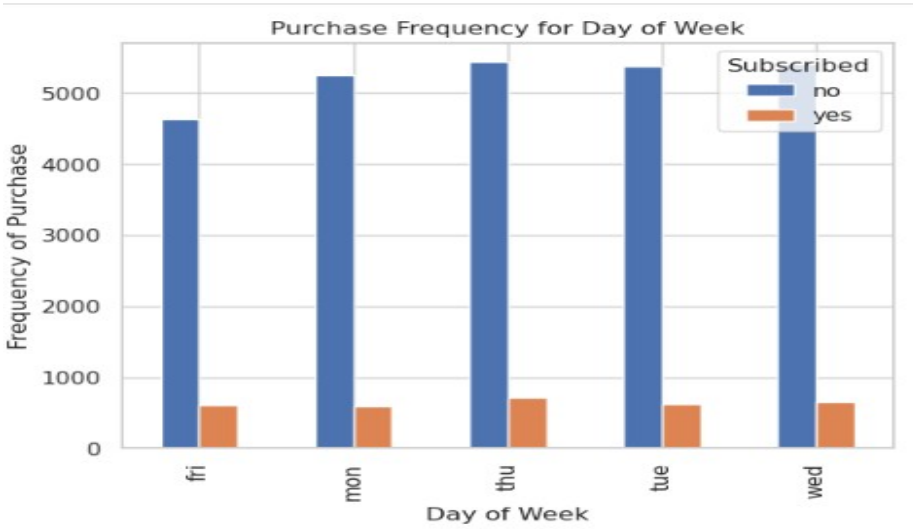
The marital status does not seem a strong predictor for the outcome variable. The statistical data for this feature is shown



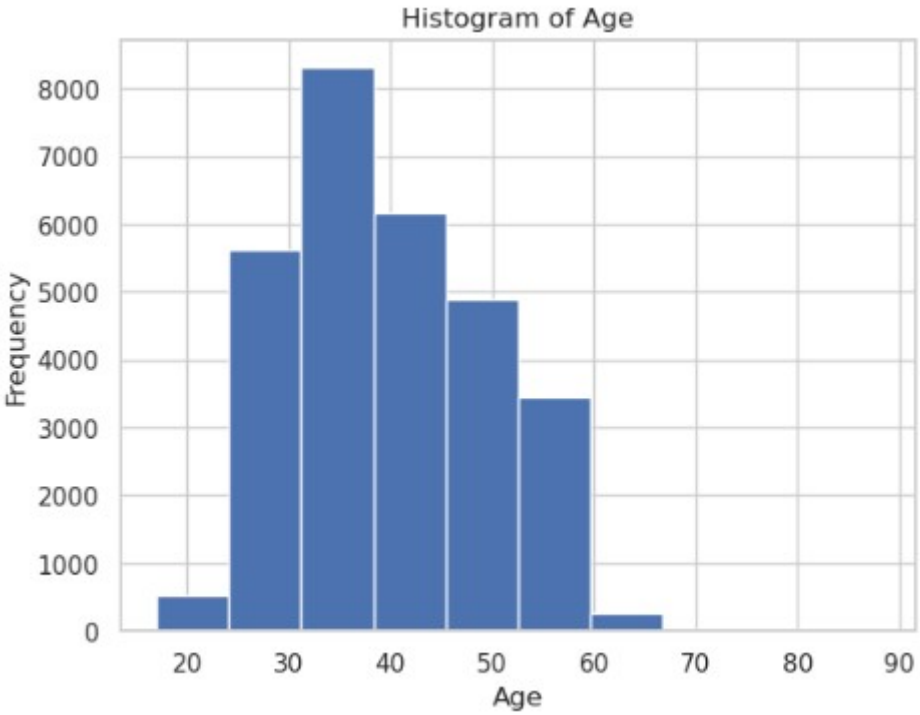
Education seems a good predictor of the outcome variable. The statistical data for the education is also shown here



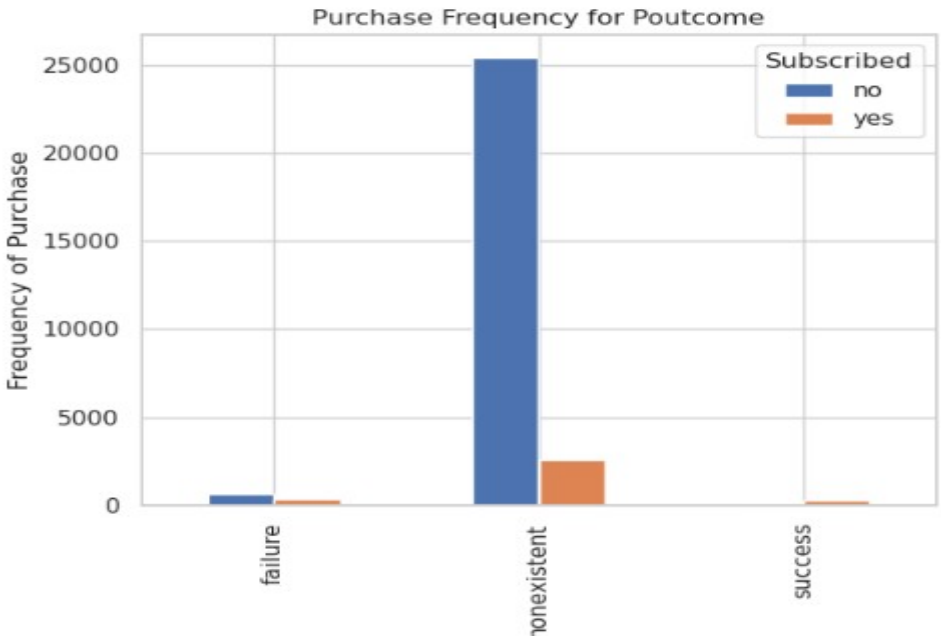
The Day of week may not be a good predictor of the outcome. Its statistical data is shown here



The histogram of the age is shown here



Most of the customers of the bank in this dataset are in the age range of 30–40.
Poutcome seems to be a good predictor of the outcome variable.



3 Learning Methods

There are two methods used in the learning of our dataset, but first thing after data exploration is the data preprocessing. In the data preprocessing, first of all we eliminate the features with unknown values and keep the features with values only yes or no, after that we up-sample the no-subscription using the SMOTE algorithm(Synthetic Minority Oversampling Technique). At a high level, SMOTE:

- Works by creating synthetic samples from the minor class (no-subscription) instead of creating copies.
- Randomly choosing one of the k-nearest-neighbors and using it to create a similar, but randomly tweaked, new observations.

Its python implementation gives us the result.

```
length of oversampled data is  36524
Number of no subscription in oversampled data 18262
Number of subscription 18262
Proportion of no subscription data in oversampled data is  0.5
Proportion of subscription data in oversampled data is  0.5
```

So, by doing this we have a perfect balanced data set, Keep in mind we over-sampled only on the training data, because by oversampling only on the training data, none of the information in the test data is being used to create synthetic observations, therefore, no information will bleed from test data into the model training.

After balancing the data, we used Recursive features elimination technique, Recursive Feature Elimination (RFE) is based on the idea to repeatedly construct a model and choose either the best or worst performing feature, setting the feature aside and then repeating the process with the rest of the features. This process is applied until all features in the dataset are exhausted. The goal of RFE is to select features by recursively considering smaller and smaller sets of features. The RFE help to select the following features

```
cols=['job_blue-collar', 'job_housemaid', 'marital_unknown', 'education_illiterate',
      'contact_cellular', 'contact_telephone', 'month_apr', 'month_aug', 'month_dec', 'month_jul', 'month_jun', 'month_mar',
      'month_may', 'month_nov', 'month_oct', "poutcome_failure", "poutcome_success"]
```

Now we implement the two models, first one is logistic regression and second one is decision tree. Now I discuss these methods one by one

a) Logistic Regression

Logistic Regression is a machine learning algorithm used for binary classification, which means it's designed to predict one of two possible outcomes. Despite its name, logistic regression is used for classification rather than regression. It's particularly useful when the dependent variable is categorical and represents two classes (e.g., 0 or 1, Yes or No, True or False)

The algorithm predicts the probability that a given instance belongs to one of these two classes, utilizing the logistic (sigmoid) function to constrain predictions between 0 and 1. The decision boundary, often set at 0.5, dictates the classification outcome, with instances above the threshold assigned to the positive class and those below to the negative class. Logistic Regression is valued for its interpretability, as it provides insights into the influence and direction of each feature through its coefficients. Widely used in binary classification scenarios, Logistic Regression's training process involves optimizing these coefficients to minimize the disparity between predicted probabilities and actual class labels. Its versatility extends to multi-class problems through strategies like one-vs-all or one-vs-one. Overall, Logistic Regression stands as a foundational algorithm in machine learning, offering simplicity and efficiency in predictive modeling.

b) Decision Tree

A Decision Tree is a versatile and intuitive machine learning algorithm used for both classification and regression tasks. It operates by recursively partitioning the data into subsets based on the values of input features, creating a tree-like structure of decisions. At each node of the tree, a feature is chosen to split the data, optimizing the separation of instances into distinct classes or predicting numerical values. The decision-making process is visualized as a tree, where each internal node represents a decision based on a specific feature, and each leaf node corresponds to the predicted outcome. Decision Trees are valued for their transparency and interpretability, allowing users to easily comprehend the decision logic. However, they can be prone to overfitting, capturing noise in the training data. Techniques like pruning and setting depth limits are employed to mitigate this. Ensemble methods like Random Forests further enhance

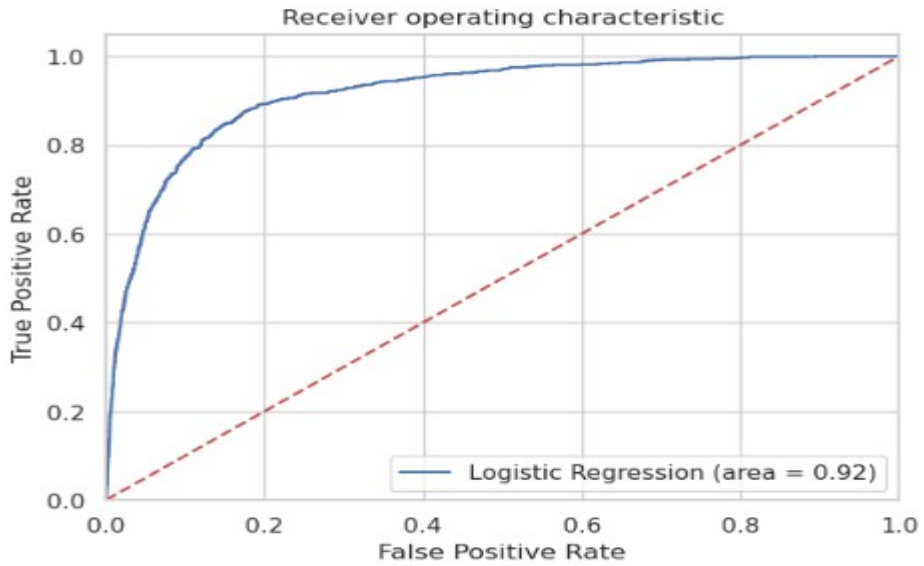
Decision Trees' performance by aggregating the predictions of multiple trees. Decision Trees find applications in various domains due to their simplicity, ease of understanding, and ability to handle both categorical and numerical features.

4 Evaluation

First of all we split our dataset into two parts, first one which is of 70% used for the training and the remaining 30% we use for the testing purposes. When we apply the logistic regression technique on our test dataset, and after that I will test it on the test set it will give me an accuracy of 92%. The confusion matrix in this case is

```
[[7676  137]
 [ 594  375]]
```

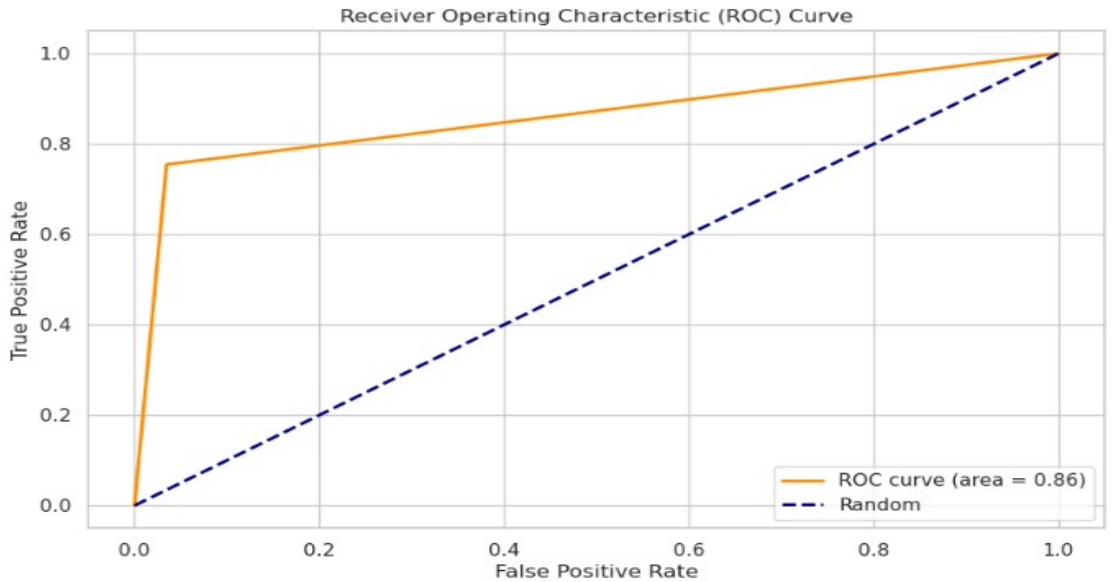
The result is telling us that we have 7676+375 correct predictions and 137+594 incorrect predictions. The F1 scores for the 'yes' case is 0.51 and for the 'no' case is 0.95. and the roc curve in this case is



When we apply the Decision tree technique on our test dataset, and after that I will test it on the test set it will give me an accuracy of 94%. The confusion matrix in this case is

```
[[7542  271]
 [ 238  731]]
```

The result is telling us that we have 7542+731 correct predictions and 238+271 incorrect predictions. The F1 scores for the 'yes' case is 0.74 and for the 'no' case is 0.97. and the roc curve in this case is



5 Discussion

According to the result, it seems that the decision tree model has a higher accuracy of 94% compared to the logistic regression model which has an accuracy of 92%. However, it is important to note that accuracy alone may not be the best metric to evaluate the performance of a model. The choice of the best model depends on the nature of the data and the problem you are trying to solve.

Logistic regression is better suited for problems where the relationship between the predictors and the response can be modeled by a linear equation, when interpretability and transparency are important, when dealing with continuous predictors, when the sample size is small and when it's needed to predict class probabilities directly 2. On the other hand, decision trees are non-linear classifiers and do not require data to be linearly separable. They are better suited for problems where the data is not linearly separable, when dealing with categorical predictors, when the sample size is large, and when the interpretability of the model is not a priority.

Confusion matrix is an important tool for evaluating the performance of a classification model. It is a matrix that summarizes the performance of a machine learning model on a set of test data. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance. The matrix displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) produced by the model on the test data 1.

The confusion matrix can be used to calculate various performance metrics such as accuracy, precision, recall, and F1-score. The F1 score is a metric that combines precision and recall to provide a single measure of a model's performance. It is the harmonic mean of precision and recall, and ranges from 0 to 1, with higher values indicating better performance. The formula for calculating the F1 score is:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

where precision is the number of true positives divided by the sum of true positives and false positives, and recall is the number of true positives divided by the sum of true positives and false negatives.

In the context of a confusion matrix, the F1 score is calculated as the harmonic mean of precision and recall, where precision is the number of true positives divided by the sum of true positives and false positives, and recall is the number of true positives divided by the sum of true positives and false negatives 14. The F1 score is a useful metric for evaluating the performance of a classification model, especially when the classes are imbalanced 14. It provides a balance between precision and recall, and is often used in situations where both precision and recall are important

6 Conclusion

In this project, we build two machine learning models, Logistic Regression and Decision Tree from the provided dataset and then test it, here is the result of our discussion.

The accuracy for the logistic regression is 92% while for the Decision tree it is 94%, the f1 score for the yes case of the logistic regression model is 0.51 and for the no case is 0.95, and for the decision tree model it is for the yes case is 0.74 and for the no case is 0.97. so based on the result that in our data set decision tree performing good so it is best model for this data set.

References (in case any references are used)

I have taken this material from the following websites.
https://www.datacamp.com/tutorial/decision-tree-classification-python
https://aws.amazon.com/free/machine-learning
https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd
https://dzone.com/articles/logistic-regression-vs-decision-tree