**Bank Marketing Effectiveness Prediction**

**Technical document**

**Sananda Biswas Chatterjee**

**AlmaBetter**

**Abstract**

At present, under the background that machine learning technology is becoming more mature and widely used in various fields, and due to the advent of the customer-oriented era and increased competition from banks. The finance sector is one of the ones that has been most affected by recent advances in machine learning. Whether it's predicting stock prices or, in our case, predicting a customer's tendency to sign up for a term deposit. As a result, we have developed a solution for our project that improves the success rate while increasing efficiency by making fewer calls.

By choosing appropriate features, analysing the data, and making predictions using machine learning algorithms that take into account historical trends, our experiment will try to light up the possible causes of the classification of these labels.

**Introduction**

The finance sector is one of the ones that has been most affected by recent advances in machine learning. Whether it's forecasting stock prices or, in our case, forecasting a customer's propensity to sign up for a term deposit. As a result, we have developed a solution for our project that improves the success rate while increasing efficiency by making fewer calls.

Our aim is to develop a predictive model that could aid in determining whether the client will subscribe for a term deposit or not.

**Problem Statement**

The data is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to assess if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe to a term deposit (variable y).

**Feature Information**

* **Age:** Age of customers (Numeric)
* **Job:** type of job that clients do. (Categorical)
* **Marital:** marital status (If a client is married, single, or divorced)
* **Education:** client's qualifications (Categorical)
* **Default:** has credit in default or not. (Categorial)
* **Housing:** Client has housing loan or not (Categorical: 'no', 'yes', 'unknown')
* **Loan:** has personal loan or not (Categorical: 'no', 'yes', 'unknown')
* **Contact:** contact communication type (categorical: 'cellular', 'telephone')
* **Month:** last contact month of year (categorical: 'Jan', 'Feb', ...,'Dec')
* **Day of week:** last contact day of the week. (Categorical)

### Campaign: number of contacts performed during this campaign and for this client (numeric)

### 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: outcome of the previous marketing campaign (categorical: 'failure', 'nonexistent', 'success')

### y: has the client subscribed a term deposit? (Binary: ‘yes’, ‘no’)

**Steps Involved**

**Importing important libraries and dataset**

Importing libraries is the first thing we'll do. Libraries to help us research the problem and conduct analysis to draw conclusions based on a body of data.

We are writing our script for this project using Google Collab. We used Bank Marketing Effectiveness prediction data that is freely available online under the Creative Commons License in order to obtain the information.

The dataset was provided by the Machine Learning Repository and

contained information on 45,211 clients across 17 different features, both

categorial (marital status, job type, education, etc.) and numeric (age, number

of days since previous contact, etc.). The target variable is a binary “Yes” (client subscribed) or “No” (client did not subscribe). The first step is to load the dataset into a data frame for easy manipulation and exploration using the

pandas’ package.

**Cleaning the dataset**

The next stage requires cleaning up the data. The data we import frequently has a number of problems, such as incorrect or missing values, etc. The data quality must be improved through cleaning in order to be used for more thorough analysis. In our dataset, are no duplicate and null values, which is very good because our accuracy may be affected by null values.

**Exploratory Data Analysis (EDA)**

After loading the dataset, we used EDA to determine how the data related to one another. EDA is a technique for data analysis that makes use of visual techniques. It is utilized to find trends, patterns, or to confirm assumptions by using statistical summaries and graphical representations.

Before you conduct data analysis or put your data through an algorithm, it is critical to have a thorough understanding of it. The patterns in your data must be understood in order to identify between the variables that are crucial and those that have little impact on the result. It's also possible that some variables will correlate with others. Detecting data errors is another important step.

Exploratory Data Analysis is capable of handling all of this. It takes out irregularities and unnecessary values from data and assist your ability to gain insights and make sense of the data.

**Feature Engineering**

Feature engineering is a machine learning method that uses data to generate new variables that aren't present in the training set. It can generate fresh features for both supervised and unsupervised learning, with the aim of simplifying and advancing data transformations while also improving model accuracy. When working with machine learning models, feature engineering is required. Unfavourable features will directly affect your model, regardless of the architecture or the data.

In machine learning, we always try to choose the optimal model to get good results. However, sometimes after choosing the wrong model, still, we can get better predictions, and this is because of better features. The flexibility in features will enable you to select the less complex models. Because less complex models are faster to run, easier to understand and maintain, which is always desirable

If we input the well-engineered features to our model, then even after selecting the wrong parameters (Not much optimal), we can still get good results. After feature engineering, it is not necessary to do hard for picking the right model with the most optimized parameters. If we have good features, we can better represent the complete data and use it to best characterize the given problem. If we have strong features, we can use them to more accurately represent the entire set of data and best describe the specific problem we're dealing with.

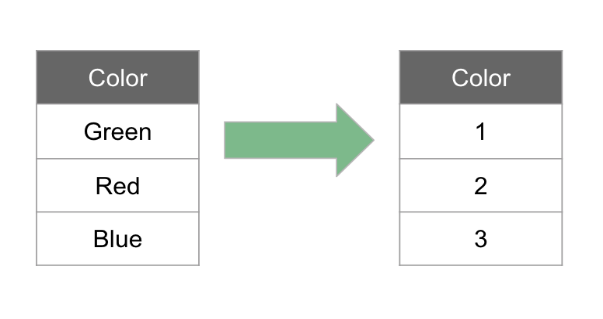
**Encoding Categorical Variables**

Encoding categorical data is a process of converting categorical data into integer format so that the data with converted categorical values can be provided to the models to give and improve the predictions.

In this encoding scheme, an ordinal encoder is used to first translate the categorical feature into a numerical format. The numbers are then converted to binary numbers. After that, the binary value is divided into various columns. A large number of categories makes binary encoding very effective.

**Label Encoding**

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine- readable form.



**One Hot Encoding**

It is also the process of turning categorical data into numerical data, but in this case, rather than labelling each category, we instead create new columns and assign binary values to each one.



**Model Building**

The dataset is divided into training data and test data with the intention of using the training data to find the parameters of the particular model being used (fitting the model on the training data) and them applying this to the test data to determine the model’s performance and to draw conclusions about its predictive capability.

We must resolve a binary classification problem as part of our project. To train and test our data, we use a supervised learning binary classification model to predict whether the customer will accept a term deposit or not.

A given set of data is defined into classes through the process of classification. Predicting the class of the provided data points is the first step in the process. The terms target, label, and classes are frequently used to describe the classes.

**Hyperparameter Tuning**

Hyperparameters are groups of data that are used to regulate how an algorithm learns. Their definitions alter the model parameters as a result of the new hyperparameters, which is viewed as a learning process.

This group of parameters has an impact on a model's functionality, stability, and interpretation. Each algorithm needs its own unique hyperparameter grid, which can be altered depending on the business problem. Hyperparameters change how a model is taught to start the training algorithm and then use the parameters to produce outputs.

To train and test the performances, we used five different kinds of models.

1) Logistic Regression

2) Decision Tree Classifier

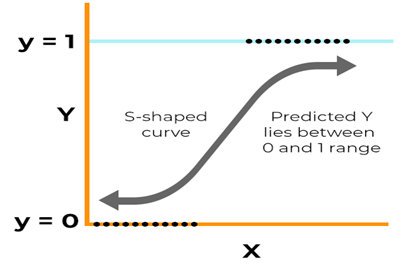
3) Random Forest Classifier

4) K-Nearest Neighbors (KNN)

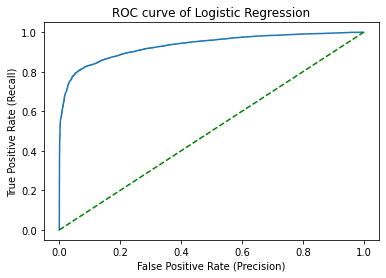
5) XGBoost Classifier

**1)Logistic Regression**

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).



**Metrics and Performance of Logistic Regression**

****

Training accuracy Score: 0.8675458194830437

Testing accuracy Score: 0.8639311209198409

precision recall f1-score support

0 0.89 0.84 0.87 9669

1 0.84 0.89 0.86 8682

accuracy 0.86 18351

macro avg 0.86 0.87 0.86 18351

weighted avg 0.87 0.86 0.86 18351

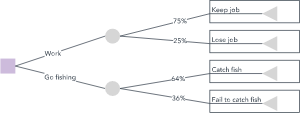
**2)Decision Tree Classifier**

Decision Tree is a Supervised Machine Learning Algorithm that uses a set of rules to make decision

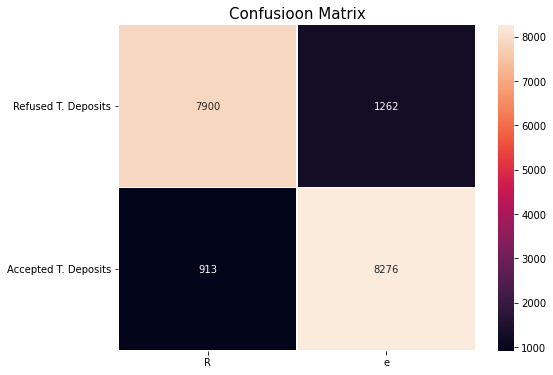
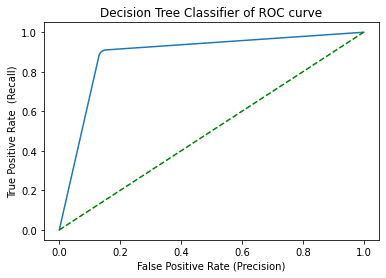
ns, similarly to how humans make decisions.

A decision tree typically starts with a single node, which branches into possible outcomes. Each of those outcomes leads to additional nodes, which branch off into other possibilities. This gives it a treelike shape.

There are three different types of nodes: chance nodes, decision nodes, and end nodes. A chance node, represented by a circle, shows the probabilities of certain results. A decision node, represented by a square, shows a decision to be made, and an end node shows the final outcome of a decision path,

****

**Metrics and Performance of Decision Tree Classifier**

** **

Training accuracy Score: 0.9917352369534812

Testing accuracy score: 0.8814778486186039

precision recall f1-score support

0 0.86 0.90 0.88 8813

1 0.90 0.87 0.88 9538

accuracy 0.88 18351

macro avg 0.88 0.88 0.88 18351

weighted avg 0.88 0.88 0.88 18351

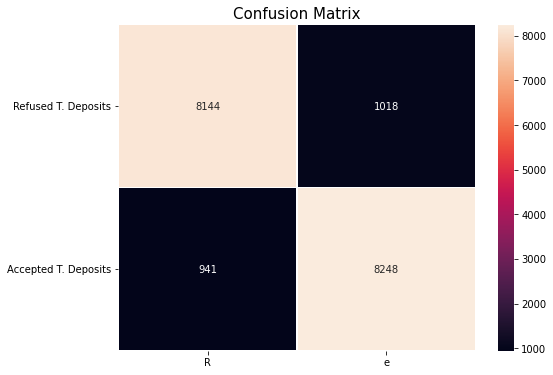
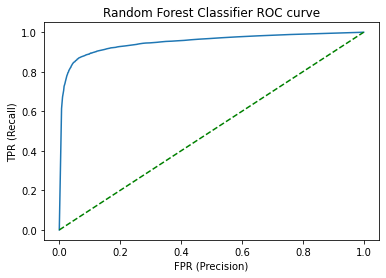
**Random Forest Classifier**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

**The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

**Metrics and parameters of Random Forest (before hyperparameter tuning)**

** **

Training accuracy Score: 0.9916807440103174

Testing accuracy Score: 0.8932483243419977

precision recall f1-score support

0 0.89 0.90 0.89 9085

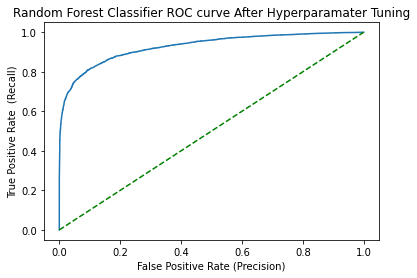
1 0.90 0.89 0.89 9266

accuracy 0.89 18351

macro avg 0.89 0.89 0.89 18351

weighted avg 0.89 0.89 0.89 18351

**Metrics and parameters of Random Forest (after hyperparameter tuning)**

** **

Training accuracy Score: 0.8596806713530598

Testing accuracy Score: 0.857228488910686

precision recall f1-score support

0 0.90 0.83 0.86 9940

1 0.82 0.89 0.85 8411

accuracy 0.86 18351

macro avg 0.86 0.86 0.86 18351

weighted avg 0.86 0.86 0.86 18351

**K-Nearest Neighbors (KNN)**

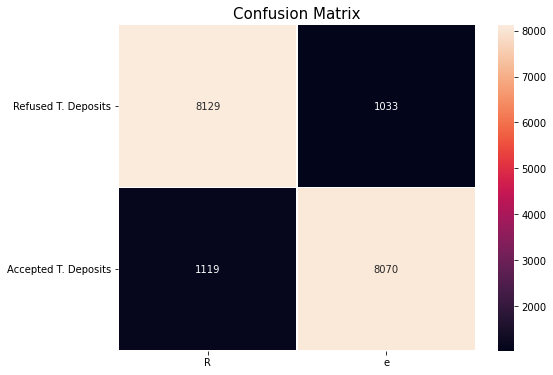
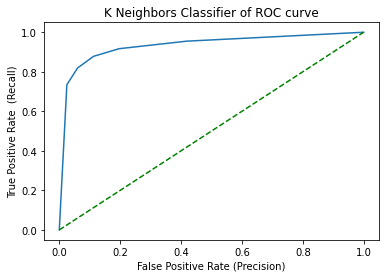
K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data.

When an algorithm examines a single point on a grid to determine whether it belongs to group A or group B, it also examines the states of the nearby points. The range is chosen at random, but the goal is to take a representative sample of the data. The data point in question is probably going to be in group A rather than B if the majority of the points are in group A, and vice versa.

Due to the fact that it does not create a model of the data set in advance, the k-nearest-neighbors algorithm is an illustration of a "lazy learner." It only performs calculations when asked to poll the data point's neighbours.

**Metrics and Performance of K-Nearest Neighbors (KNN)**

** **

Training accuracy score: 0.914191778831308

Testing accuracy score: 0.8827311863113727

precision recall f1-score support

0 0.89 0.88 0.88 9248

1 0.88 0.89 0.88 9103

accuracy 0.88 18351

macro avg 0.88 0.88 0.88 18351

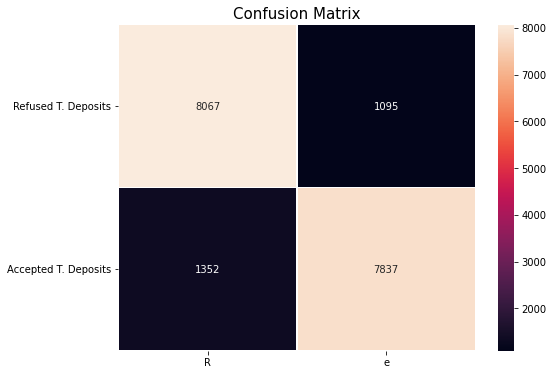
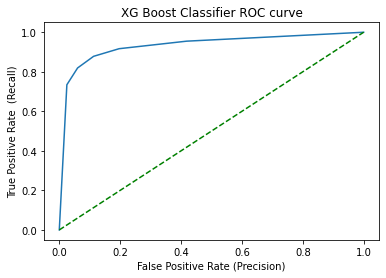
weighted avg 0.88 0.88 0.88 18351

**XGBoost Classifier**

**One of the most well-liked gradient boosting variations is XGBoost. It is an ensemble machine learning algorithm built on decision trees that makes use of the gradient boosting framework. A Machine Learning model's performance and speed can be improved by using XGBoost. Artificial neural networks frequently outperform all other algorithms or frameworks in prediction problems involving unstructured data (images, text, etc.). However, decision tree-based algorithms are currently regarded as best-in-class for handling small to medium amounts of structured or tabular data.**

**To determine the best split, XGBoost uses a pre-sorted algorithm and a histogram-based algorithm. The histogram-based algorithm divides each data point for a feature into discrete bins, then uses these discrete bins to calculate the histogram's split value. Additionally, in XGBoost, the trees' number of terminal nodes can vary, and the left weights of the trees whose calculations use less evidence are shrunk more severely.**

**Metrics and Performance of XGBoost classifier**

** **

Training accuracy score: 0.8687264999182606

Testing accuracy score: 0.8666557680780339

precision recall f1-score support

0 0.88 0.86 0.87 9419

1 0.85 0.88 0.86 8932

accuracy 0.87 18351

macro avg 0.87 0.87 0.87 18351

weighted avg 0.87 0.87 0.87 18351

**Comparing ROC AUC curves and it’s performance**

**roc\_auc\_score for different classifiers**

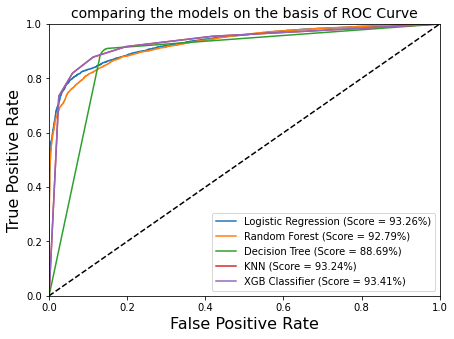
Logistic Regression score: 0.9326524085190647

Random Forest Classifier Score: 0.9279227457713373

Decision Tree Score: 0.8869899611612445

XGB Classifier score: 0.9341349784958046

KNN Score: 0.93245774674972

****

**Conclusion**

* Blue-collar, management and technician showed maximum interest in subscription.
* Divorce people have no interest in term deposit.
* The majority of the customers are between the ages of 30 and 40.
* The model can assist in identifying customers based on whether they have made deposits or not.
* Most people have home loans, but only a small percentage of them chose term deposits.
* The outcome of the campaign is significantly influenced by the customer's account balance. We can then interact with those customers who have a balanced account balance.
* The model can assist in identifying customers based on whether they have made deposits or not.
* Instead of wasting time on the wrong customer, the model helps to target the right one.
* After implementing all the ML models, We get maximum accuracy and ROC-AUC score in XGBoost. So, we can conclude that it is the best model for us.