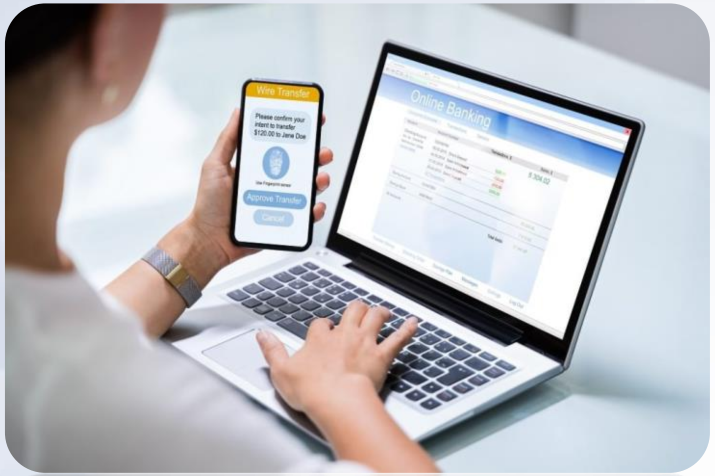


Bank Term Deposit Predictive Model



Introduction

Business Use Case

There has been a revenue decline for a Best Bank and they would like to know what actions to take. After investigation, they found out that the root cause is that their clients are not depositing as frequently as before. Knowing that term deposits allow banks to hold onto a deposit for a specific amount of time, so banks can invest in higher gain financial products to make a profit. In addition, banks also hold better chance to persuade term deposit clients into buying other products such as funds or insurance to further increase their revenues. As a result, the Best bank would like to identify existing clients that have higher chance to subscribe for a term deposit and focus marketing efforts on such clients.

Data Science Problem Statement

Predict if the client will subscribe to a term deposit based on the analysis of the marketing campaigns the bank performed.

Presenter: Simon Christopher Jacobe

EDA and Data Preparation

Data Loading and Cleaning

```
# accessing to the folder where the file is stored
path = 'new_train.csv'

# Load the dataframe
dataframe = pd.read_csv(path)

print('Shape of the data is: ',dataframe.shape)

dataframe.head()
```

```
Shape of the data is:  (32950, 16)
```

Check Data types

```
age            int64
job            int32
marital        int32
education      int32
default        int32
housing        int32
loan           int32
contact        int32
month          int32
day_of_week    int32
duration       int64
campaign       int64
poutcome       int32
y              int32
dtype: object
```

Dropping Missing Values

```
age            0
job            0
marital        0
education      0
default        0
housing        0
loan           0
contact        0
month          0
day_of_week    0
duration       0
campaign       0
pdays          0
previous       0
poutcome       0
y              0
dtype: int64
```

Check of Class Imbalance

```
no     88.734446
yes    11.265554
Name: y, dtype: float64
```

imbalanced dataset

Identifying Numeric Features

```
Numeric Features:
   age  duration  campaign  pdays  previous
0   49       227         4    999         0
1   37       202         2    999         1
2   78      1148         1    999         0
3   36       120         2    999         0
4   59       368         2    999         0
==============================================
```

Identifying Categorical Features

```
Categorical Features:
           job   marital          education  default housing loan    contact
0  blue-collar   married           basic.9y  unknown      no   no   cellular
1  entrepreneur  married  university.degree       no      no   no  telephone
2      retired   married           basic.4y       no      no   no   cellular
3       admin.   married  university.degree       no     yes   no  telephone
4      retired  divorced  university.degree       no      no   no   cellular

  month day_of_week    poutcome    y
0   nov         wed  nonexistent   no
1   nov         wed      failure   no
2   jul         mon  nonexistent  yes
3   may         mon  nonexistent   no
4   jun         tue  nonexistent   no
================================================================================
```

EDA and Data Preparation

Function to Label Encode Categorical variables

Before applying our machine learning algorithm, we need to recollect that any algorithm can only read numerical values.
It is therefore essential to encode categorical features into numerical values.

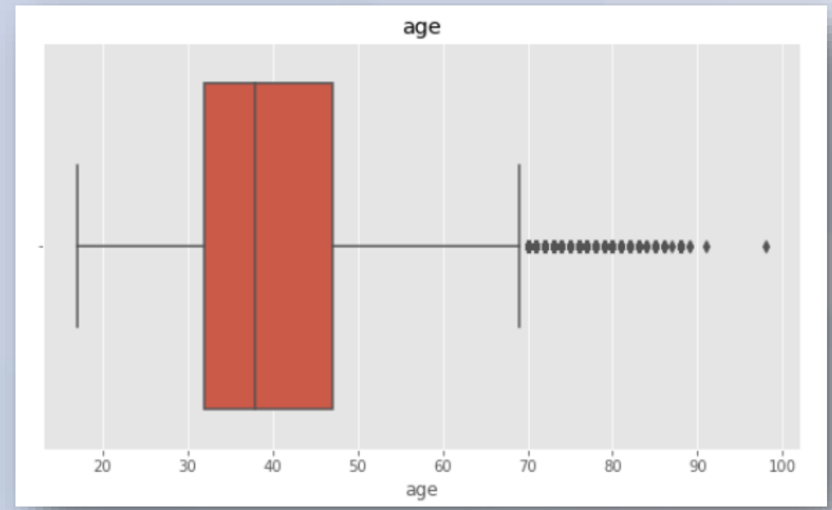
For the given dataset, we are going to label encode the categorical columns.

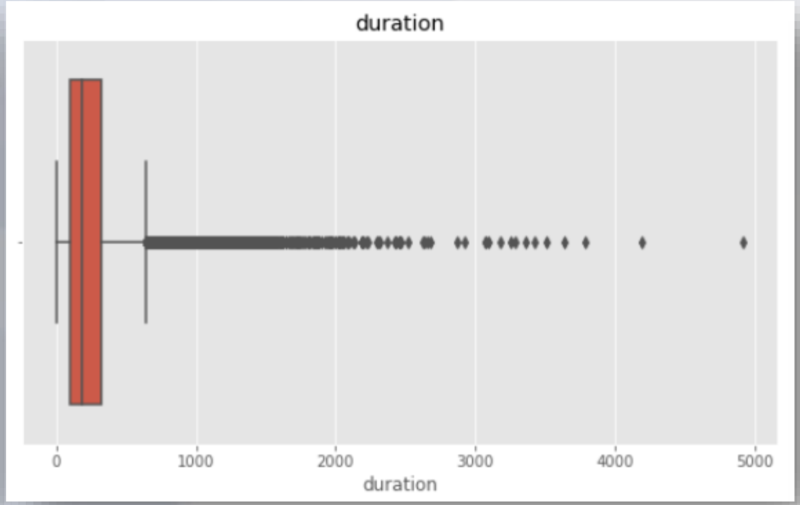
In [39]:
```
dataframe.head()
```

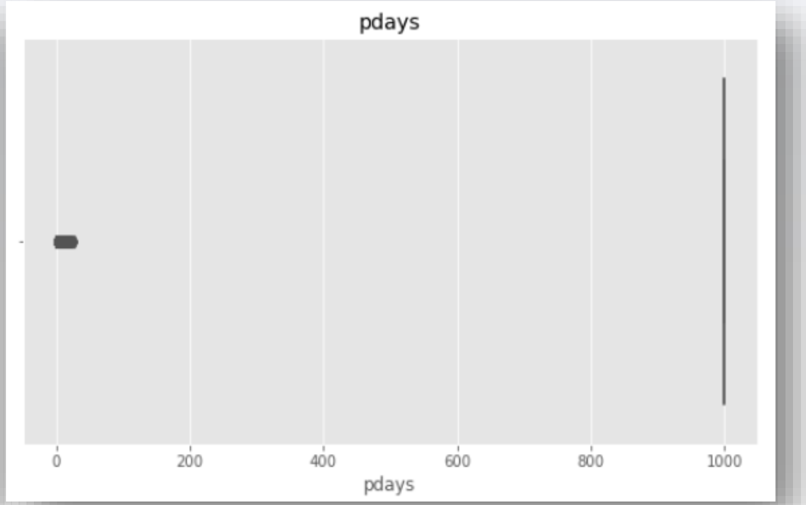
Out[39]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	poutcome	y
0	49	1	1	2	0	0	0	0	7	4	227	4	1	0
1	37	2	1	6	0	0	0	1	7	4	202	2	0	0
2	55	5	1	0	0	0	0	0	3	1	550	1	1	1
3	36	0	1	6	0	1	0	1	6	1	120	2	1	0
4	55	5	0	6	0	0	0	0	4	3	368	2	1	0

EDA







Observation:

- As we can see from the histogram, the features age, duration and campaign are heavily skewed and this is due to the presence of outliers as seen in the boxplot for these features. We will deal with these outliers in the steps below.

- Looking at the plot for pdays, we can infer that majority of the customers were being contacted for the first time because as per the feature description for pdays the value 999 indicates that the customer had not been contacted previously.

- Since the features pdays and previous consist majorly only of a single value, their variance is quite less and hence we can drop them since technically will be of no help in prediction.

EDA and Data Preparation

Dropping the columns pdays & previous

```
dataframe.drop(['pdays','previous'],1,inplace=True)
```

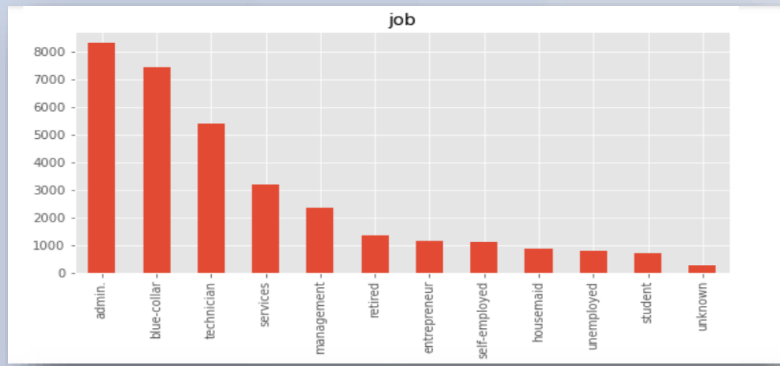
```
Shape of the data is:  (32950, 14)
```

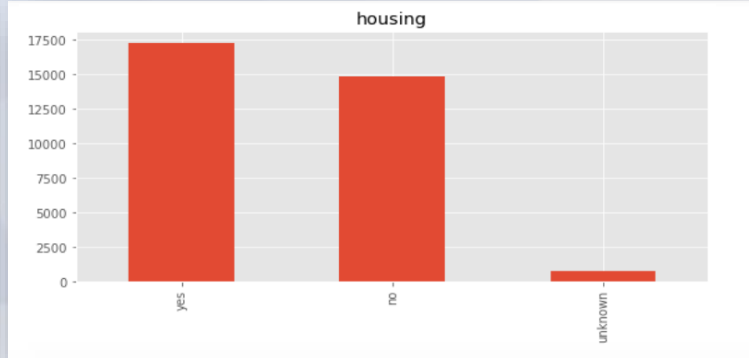
Out[3]:

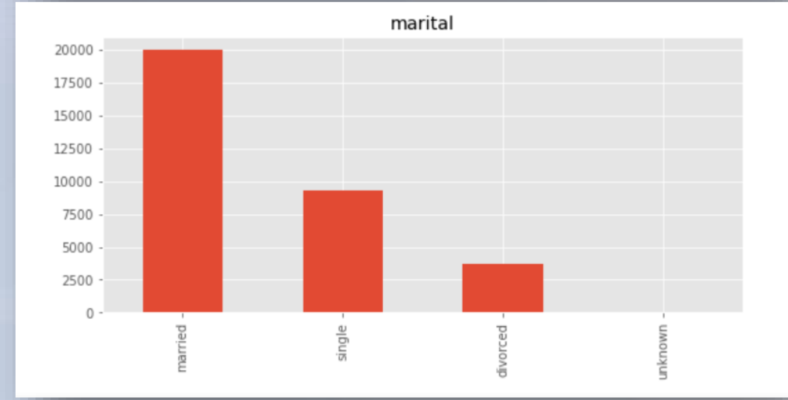
	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	poutcome	y
0	49	1	1	2	0	0	0	0	7	4	227	4	1	0
1	37	2	1	6	0	0	0	1	7	4	202	2	0	0
2	55	5	1	0	0	0	0	0	3	1	550	1	1	1
3	36	0	1	6	0	1	0	1	6	1	120	2	1	0
4	55	5	0	6	0	0	0	0	4	3	368	2	1	0

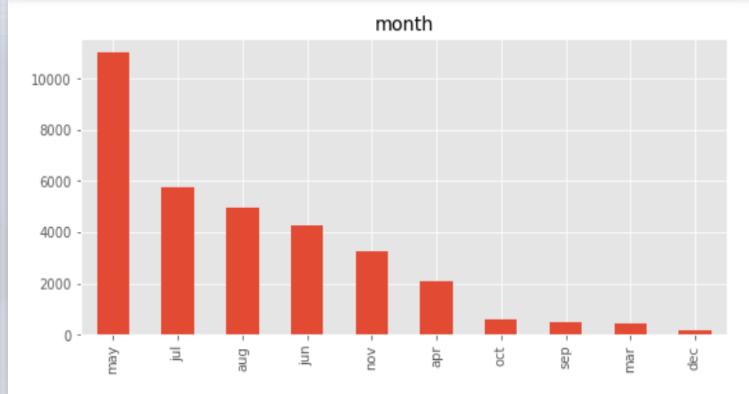
Before applying our machine learning algorithm, we need to recollect that any algorithm can only read numerical values.
It is therefore essential to encode categorical features into numerical values.

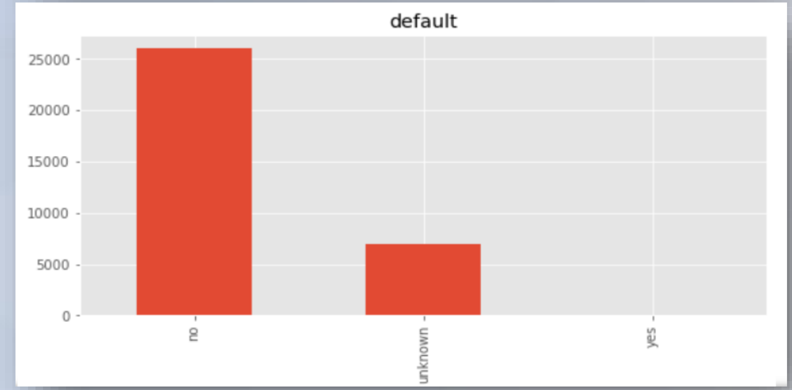
For the given dataset, we are going to label encode the categorical columns.

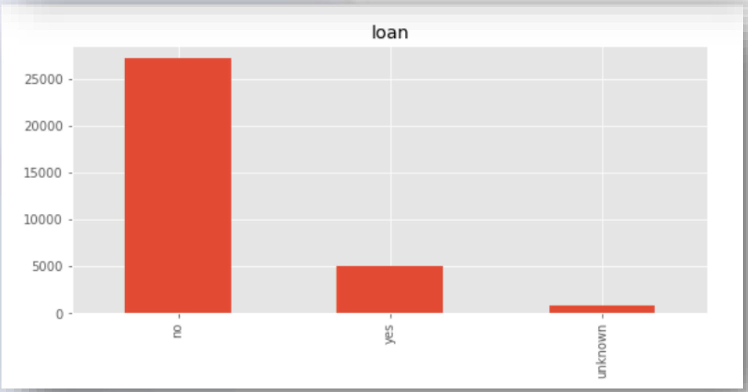












Data Visualizations

Observations :

From the visuals, we can make the following observations:

• The top three professions that our customers belong to are - administration, blue-collar jobs and technicians.

• A huge number of the customers are married.

• Majority of the customers do not have a credit in default

• Many of our past customers have applied for a housing loan but very few have applied for personal loans

• Many customers have been contacted in the month of **May**.

• The plot for the target variable shows heavy imbalance in the target variable.

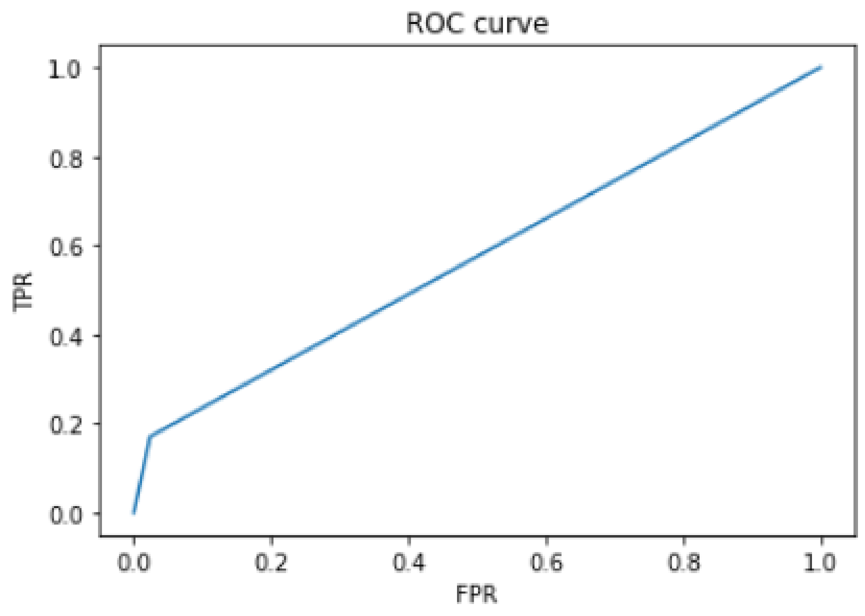
Logistic Regression

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

           0       0.90      0.98      0.93      5798
           1       0.50      0.17      0.25       792

    accuracy                           0.88      6590
   macro avg       0.70      0.57      0.59      6590
weighted avg       0.85      0.88      0.85      6590

ROC_AUC_SCORE is 0.5734990905955031
```



Precision and Recall:

Recall: Is the total number of "Yes" in the label column of the dataset. So how many "Yes" labels does our model detect.

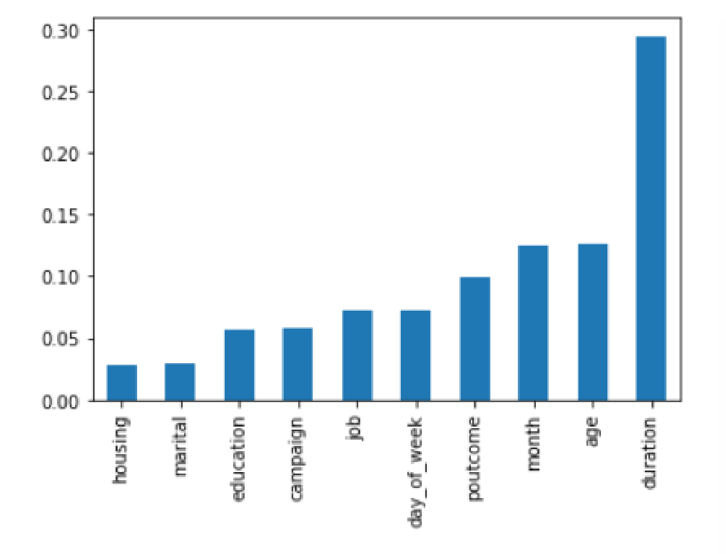
Precision: Means how sure is the prediction of our model that the actual label is a "Yes".

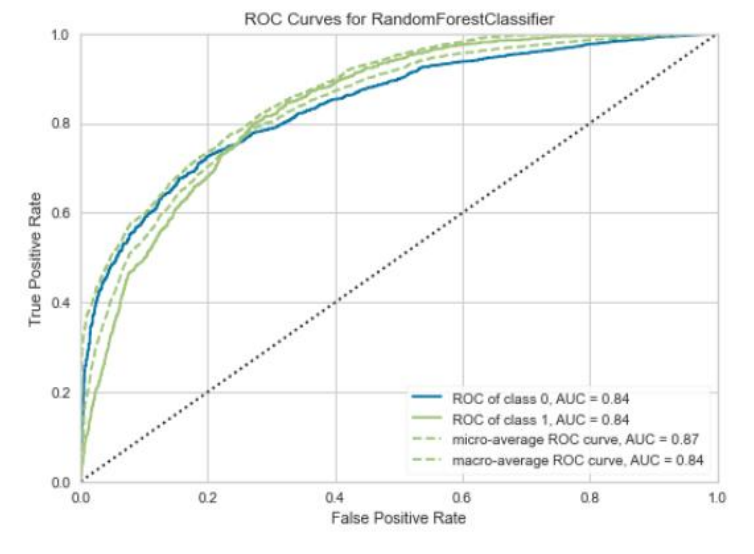
Recall Precision Tradeoff:

As the precision gets higher the recall gets lower and vice versa.

Random Forest

```
Features  to be selected for Logistic Regression model are:
['marital', 'education', 'housing', 'loan', 'contact', 'day_of_week', 'campaign', 'poutcome']
================================================================================================
```





	precision	recall	f1-score	support
0	0.96	0.77	0.86	8723
1	0.30	0.73	0.43	1162
accuracy			0.77	9885
macro avg	0.63	0.75	0.64	9885
weighted avg	0.88	0.77	0.80	9885

Observations:

RFE (Recursive Feature Elimination) is a wrapper method that uses the model to identify the best features. For the task, we have inputted 8 feature. You can change this value and input the number of features you want to retain for your model

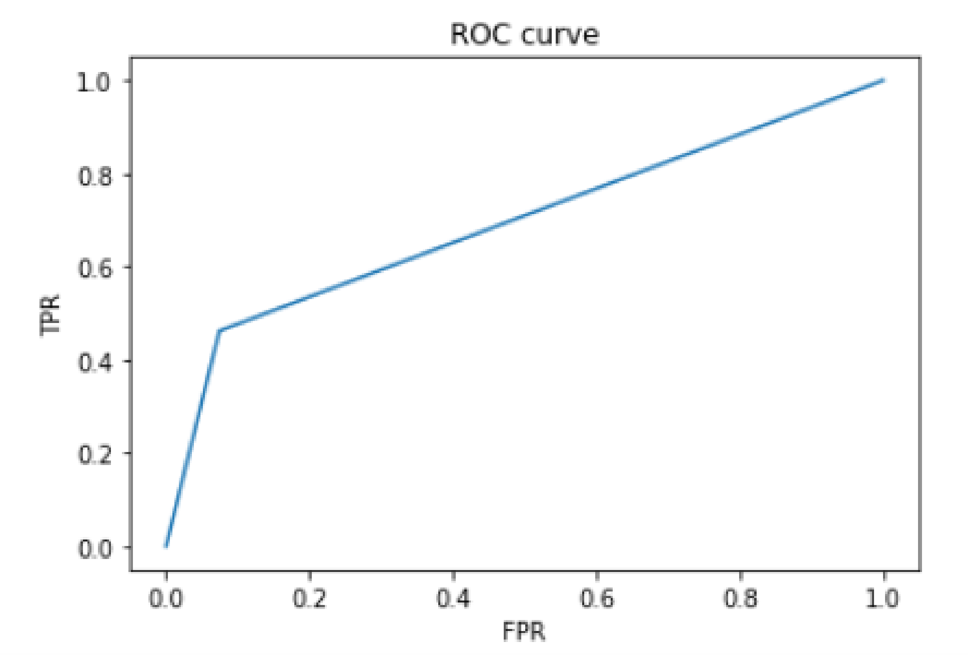
We can test the features obtained from both the feature selection techniques by inserting these features to the model and depending on which set of features perform better, we can retain them for the model.

Decision Tree

Classification Report:

	precision	recall	f1-score	support
0	0.93	0.93	0.93	5798
1	0.46	0.46	0.46	792
accuracy			0.87	6590
macro avg	0.69	0.69	0.69	6590
weighted avg	0.87	0.87	0.87	6590

ROC_AUC_SCORE is 0.6938926170988952



93% of the predictions for each of the classes are actually of the predicted class, and 7% are actually of the opposite class.

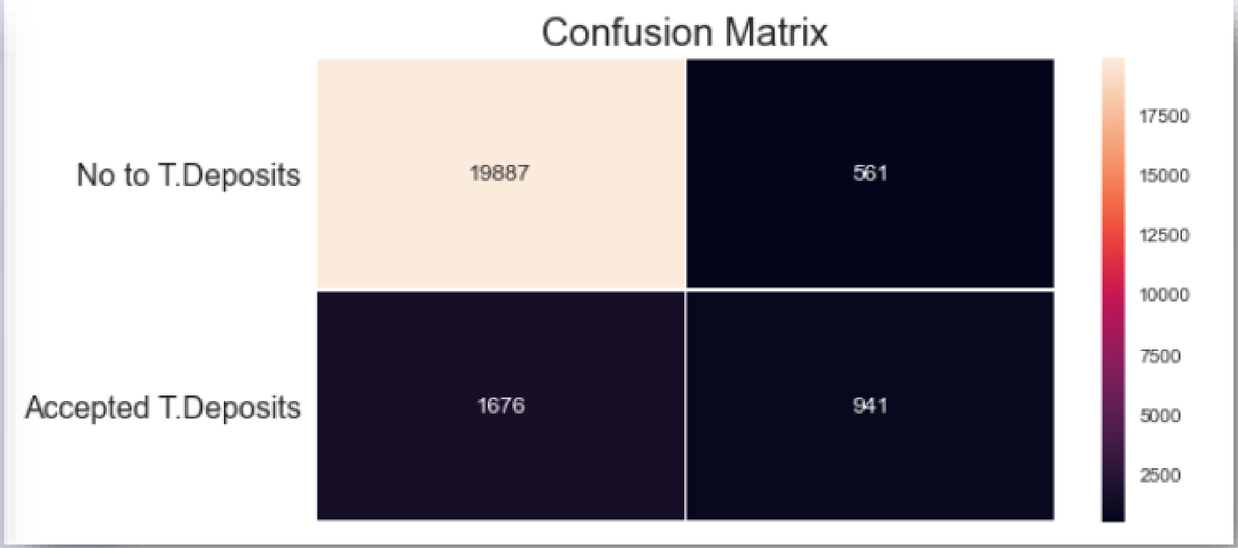
Recall is the proportion of the true positives that are identified as such. This means that the model is correctly identifying 93% of the class 0s, but only 7% of the class 1s.

F1-Score is average of the Precision and Recall; it's an attempt to provide a unified figure of the model's performance. It's calculated via the formula 2 x ((precision x recall) / (precision + recall)).

The ROC curve tells us how well our classifier is classifying between term deposit subscriptions (True Positives) and non-term deposit subscriptions. The X-axis is represented by False positive rates (Specificity) and the Y-axis is represented by the True Positive Rate (Sensitivity.) As the line moves the threshold of the classification changes giving us different values. The closer is the line to our top left corner the better is our model separating both classes.

Insights of a Confusion Matrix

Actual \ Prediction	0	1
0	TN	FP
1	FN	TP



True Negatives (Top-Left Square):
This is the number of **correctly** classifications of the "No" class or potential clients that are **not willing** to subscribe a term deposit.

False Positives (Top-Right Square):
This is the number of **incorrectly** classifications of the "Yes" class or potential clients that are **willing** to subscribe a term deposit.

False Negatives (Bottom-Left Square):
This is the number of **incorrectly** classifications of the "No" class or potential clients that are **not willing** to subscribe a term deposit.

True Positives (Bottom-Right Square):
This is the number of **correctly** classifications of the "Yes" class or potential clients that are **willing** to subscribe a term deposit.

In Conclusion

End

Source: [Bank Marketing UCI | Kaggle]