Toronto Metropolitan University

Final Project Report

COURSE

Business Intelligence and Analytics

COURSE CODE AND SECTION

ITM 618-031

TOPIC

Marketing For The Banking System

GROUP MEMBERS

Student 1

Amaya Shields

Student 2

Student 3

COURSE PROFESSOR

Dr. Mehdi Kargar

Table of Contents

Introduction

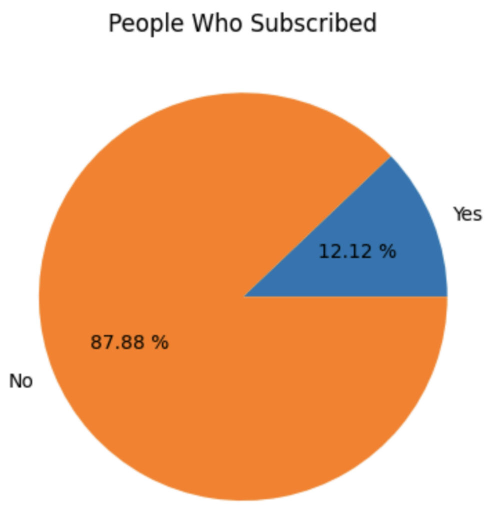
The project’s objective is to arrange the problem in regard to banking institutions' direct campaigns for marketing. The project's end goal is determining whether the term deposit of the client is subscribed or not subscribed. The target class with the yes or no response is found in the Target_Attribute. The response describes the term deposit subscription. Testing both databases given is how the process starts, focusing primarily on the data. In addition, the classification learning model is created and the data is processed.

Data Exploration

While exploring the data we found a lot of different insights regarding the different attributes in relation to the target attribute. To start analyzing the data we found that there were a lot of “unknown” values that had to be cleaned, to do this we decided to go with ‘fillna’ pandas command to fill in the unknown values with the median value from that row. There were 1157 cells in the test data and 2964 cells in the train data that had to be filled with the median value.

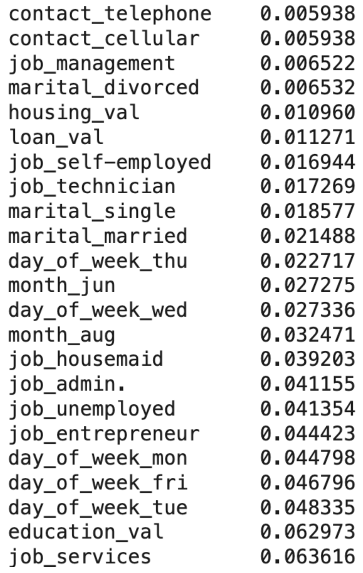
When exploring the target attribute for the traindata set, there were more no’s, represented by the number 0, than yes’s, represented by the number 1. In figure 1 we can see in the Pie chart for the target attribute “subscribed” there were 87.9% who did not subscribe in the test data and 12.1% of people who did subscribe. Majority of people are not subscribing to a term deposit. However there is some higher correlation between people who did subscribe and some of the attributes provided.

Figure 1:

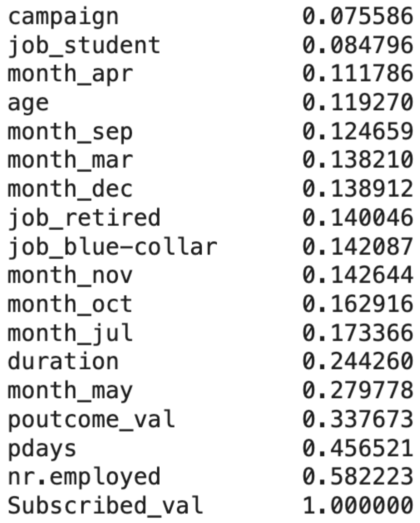


Below in table 1, are the correlations between the other attributes and the target attribute ‘Subscribed’ in increasing order. The top 3 correlations were explored in detail below, however some other attributes were helpful in finding different models to predict the target attribute. The column where we had the most unknown values was the “education” column for both the train data set and the test data set. This is interesting considering that it is around the middle of the correlation for our target attribute, with a correlation to the target attribute of 6.29%. With the least correlation being 5.94% which was the “contact” attribute, and the highest correlation being “nr.employed”. Maybe if we had more information regarding the education of the customers, this would have had a higher correlation to the subscribed attribute.

Table 3: Correlation between Target Attributes and Other Attributes



```
contact_telephone     0.005938
contact_cellular      0.005938
job_management        0.006522
marital_divorced      0.006532
housing_val           0.010960
loan_val              0.011271
job_self-employed     0.016944
job_technician        0.017269
marital_single        0.018577
marital_married       0.021488
day_of_week_thu       0.022717
month_jun             0.027275
day_of_week_wed       0.027336
month_aug             0.032471
job_housemaid         0.039203
job_admin.            0.041155
job_unemployed        0.041354
job_entrepreneur      0.044423
day_of_week_mon       0.044798
day_of_week_fri       0.046796
day_of_week_tue       0.048335
education_val         0.062973
job_services          0.063616
```



```
campaign            0.075586
job_student         0.084796
month_apr           0.111786
age                 0.119270
month_sep           0.124659
month_mar           0.138210
month_dec           0.138912
job_retired         0.140046
job_blue-collar     0.142087
month_nov           0.142644
month_oct           0.162916
month_jul           0.173366
duration            0.244260
month_may           0.279778
poutcome_val        0.337673
pdays               0.456521
nr.employed         0.582223
Subscribed_val      1.000000
```

The top 3 attributes with the highest information gain are “nr.employed”, “pdays” and “poutcome”. Interestingly enough these are all correlated with each other when we compare them, when looking at the correlation between one and the other two attributes the correlation is higher than 32% for all of them. The fact that these are correlated makes sense since they all directly relate to a campaign that was done.

Exploring the “nr.employed” attribute compared to the target attribute, which had 58.2% correlation, there we found that the more employees that we had the less people subscribed. If we look at table 1 and figure 2 we can see the exact numbers and a comparison between the subscribed people and the number of employees. When the company had 4991.6 employees there were the most yes’s recorded compared to the other numbers of employees. With a 52% chance of getting a yes compared to when there were 5099.1 employees and there was a 0% chance of getting a yes. The reason for not getting as many yeses may be due to since there were more people, less people did work, as the old saying goes “one bad apple spoils the whole bunch”. Maybe employees were not motivated enough during this time to try and get a sale or some other outside factor

Figure 2 : Stacked Bar of ‘Nr.Employed’ Compared to Target Attribute Options

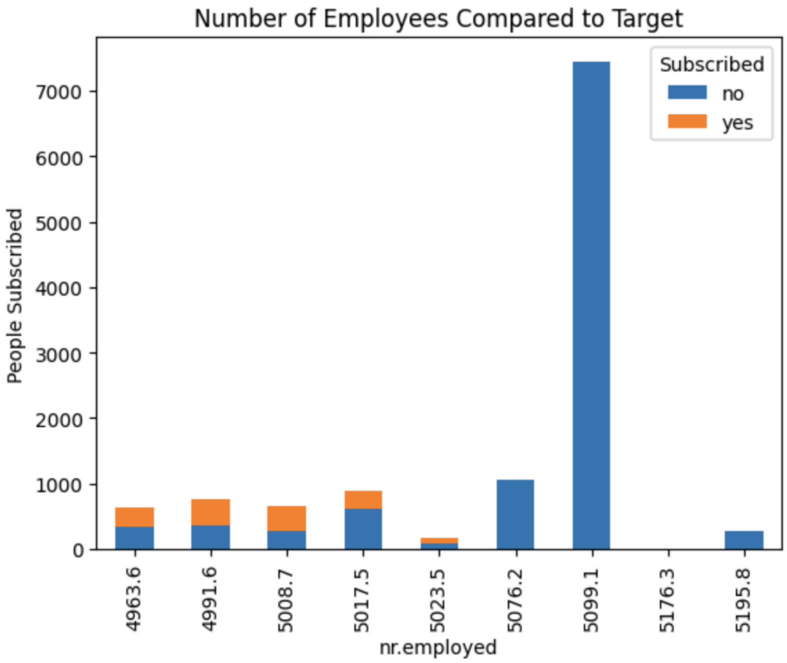
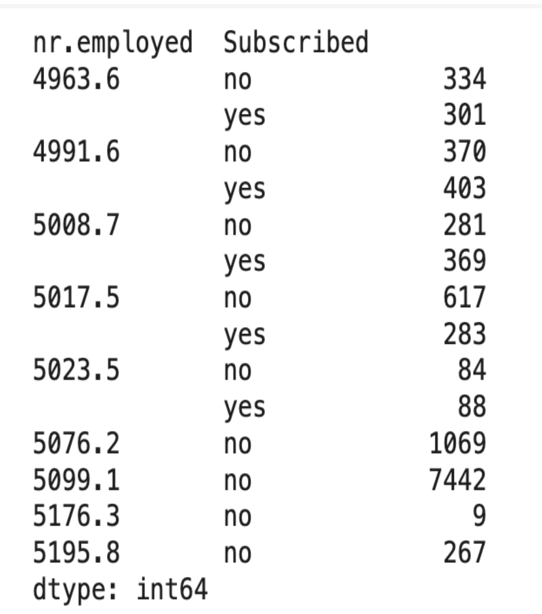


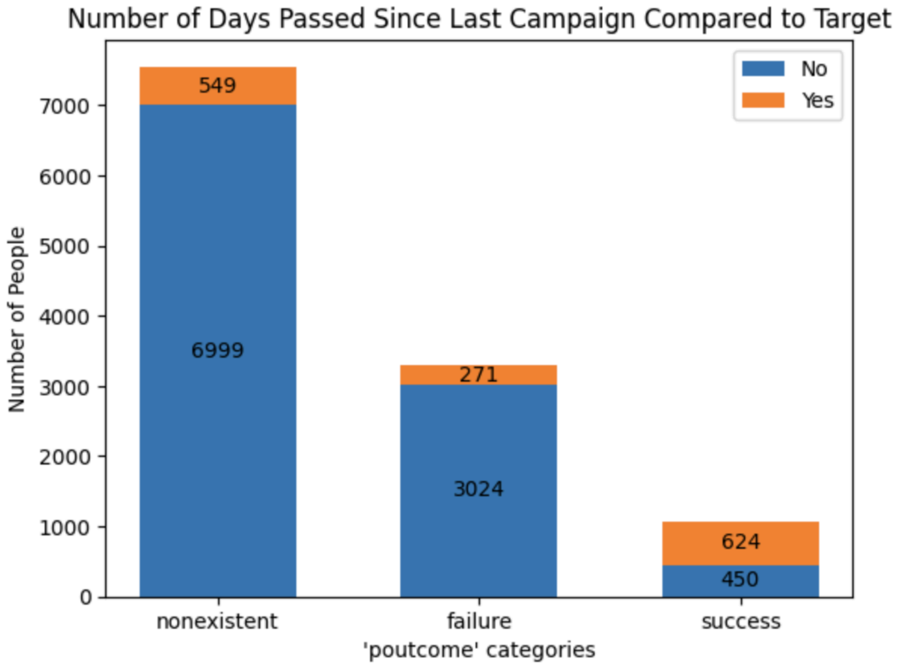
Table 1: ‘Nr.Employed’ Attribute Compared to Target Attribute Options



```
nr.employed  Subscribed
4963.6       no            334
             yes           301
4991.6       no            370
             yes           403
5008.7       no            281
             yes           369
5017.5       no            617
             yes           283
5023.5       no             84
             yes            88
5076.2       no           1069
5099.1       no           7442
5176.3       no              9
5195.8       no            267
dtype: int64
```

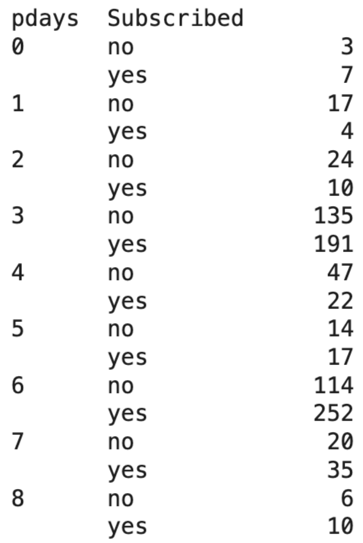
When comparing ‘poutcome’ to the target attribute, which had 33.7% correlation, we can see in figure 3 that there were a lot of nonexistent values for the previous marketing campaign as well as a large number of those who decided to subscribe to a term deposit. However when comparing the ‘poutcome’ to the success value we see that there is a higher chance of getting someone to subscribe when the outcome of the campaign was a success. Therefore the company’s campaigns are working, however we may need to look into how we can get them to be more appealing so that they are more successful.

Figure 3: Stacked Bar of 'Pdays' Compared to Target Attribute Options



Exploring the 'pdays' attribute we found that there was a 73% correlation between this and 'poutcome', and a 45.6% correlation between 'pdays' and the target attribute. As seen in table 2, depending on when the agent called the customer back since the last campaign there were more people who subscribed. There was a higher rate of subscribers when there were calls on the 3rd and 6th day from the last campaign showing that this is probably the best time to call customers back if we are trying to get them to subscribe.

Table 2: Snippet of 'Pdays' Attribute Compared to Target Attribute Options



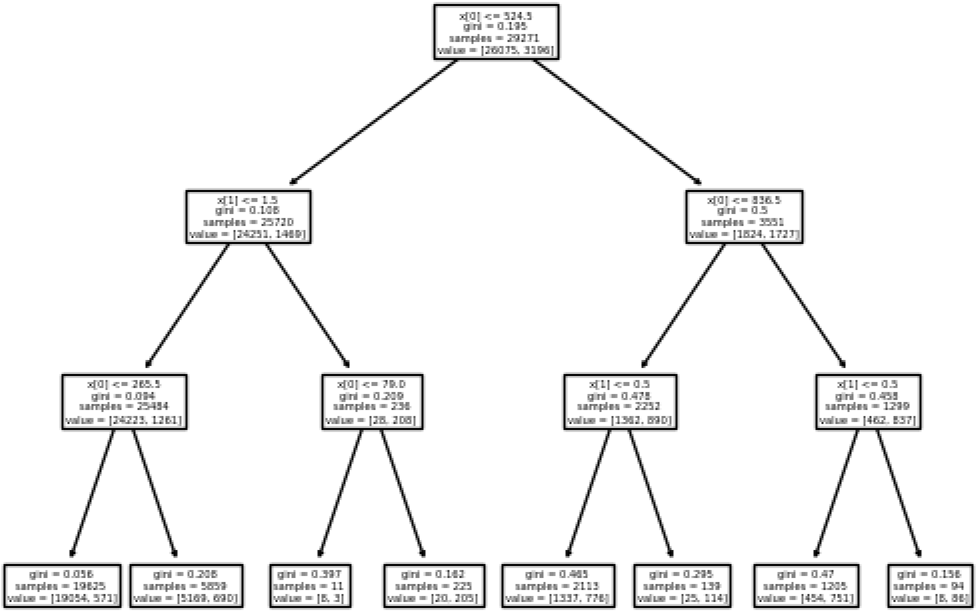
```
pdays  Subscribed
0      no              3
       yes             7
1      no             17
       yes             4
2      no             24
       yes            10
3      no            135
       yes           191
4      no             47
       yes            22
5      no             14
       yes            17
6      no            114
       yes           252
7      no             20
       yes            35
8      no              6
       yes            10
```

Learning Methods:

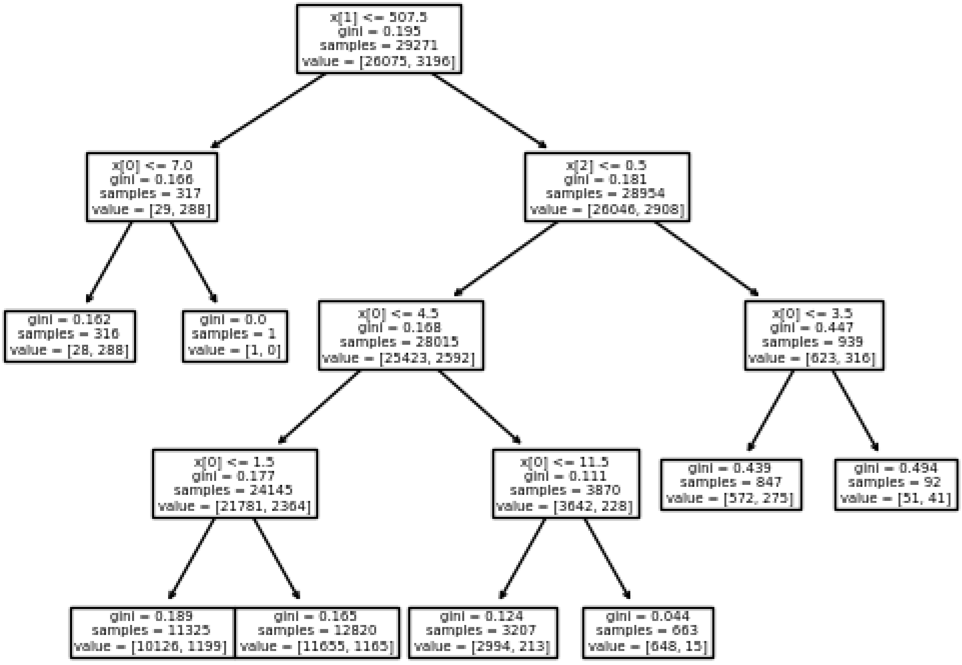
Model 1: Decision Tree

The first model that was done was the decision tree. We used this method because we are dealing with supervised data and a binary target attribute. Classification would work best with this kind of data so we felt like a decision tree made the most sense. There are a few different models that were made to make sure the model was as accurate as possible.

The first model that was made was with two attributes ‘duration’ and ‘poutcome_val’. The depth of the tree and the number of leaf nodes was restricted to 3 and 10 respectively.



The second model had three attributes which were: 'campaign', 'pdays', 'poutcome_val'. This tree's depth and maximum leaf nodes were also restricted to 4 and 8 respectively.

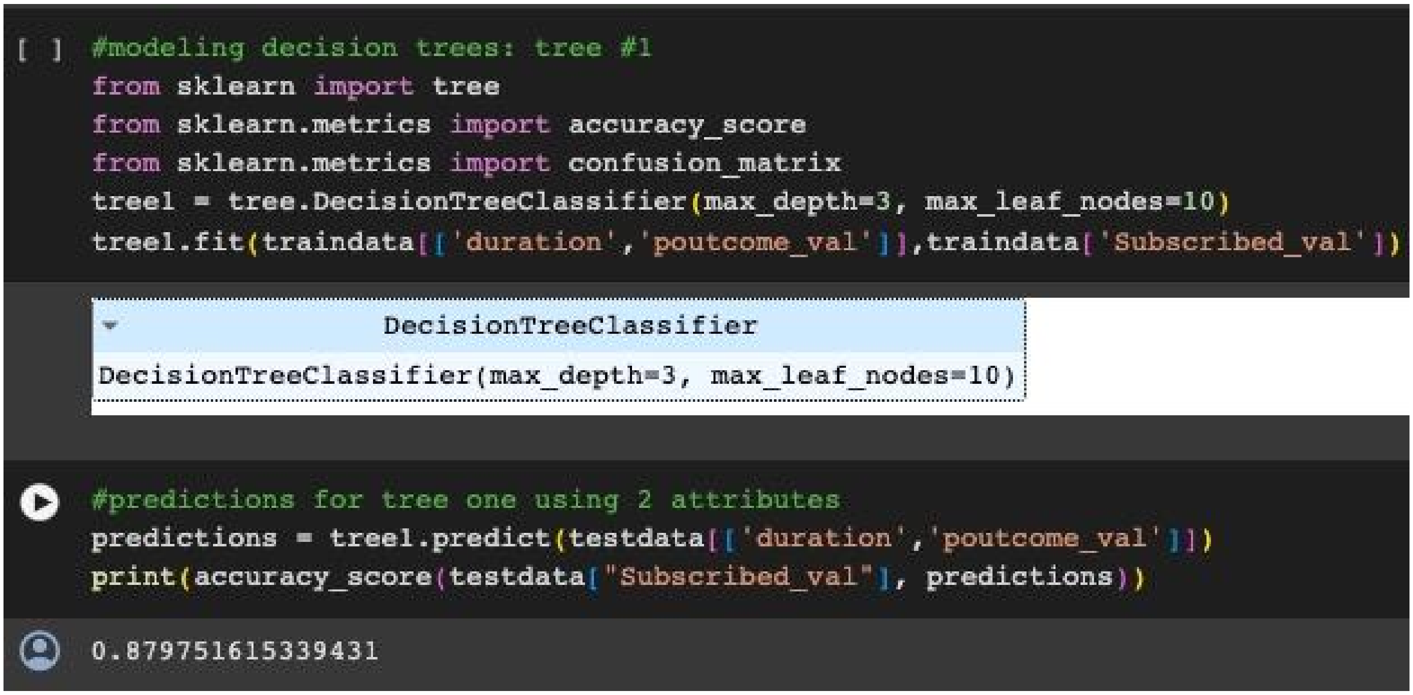


Model 2: Logistic Regression

We use logistic regression as a learning method because we can easily interpret the coefficients of the model and understand the impact of each feature on the probability of belonging to a certain class. We built 3 different models and trained each model on the training set, and evaluated each performance on the testing set to assess the generalization accuracy. For the first model we used four attributes to build the model, 'age','campaign','duration','pdays'. For the second model we used three attributes to build the model 'campaign', 'poutcome_val', 'pdays' and for the third model we used two attributes to build the model 'age','duration'. Using a variety of attribute combinations was done to examine various facets of the data and identify a range of patterns that could enhance prediction accuracy. For each model, the method predicts the class for the input dataset as we want to test the performance on the test dataset. To assess the performance on the test dataset, the regression method predicts the class for each model given the input dataset. The real class is then extracted from the test dataset to create the confusion matrix, and lastly, we extract the true positives, following our prediction of the input dataset's class.

Evaluation

Classification Tree model 1:



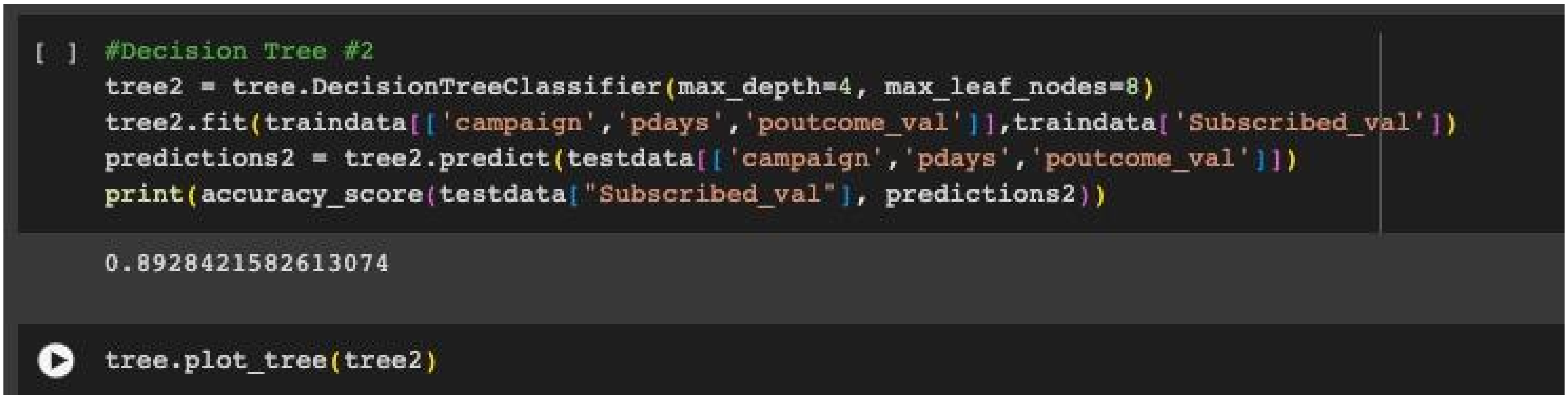
```
[ ]  #modeling decision trees: tree #1
     from sklearn import tree
     from sklearn.metrics import accuracy_score
     from sklearn.metrics import confusion_matrix
     treel = tree.DecisionTreeClassifier(max_depth=3, max_leaf_nodes=10)
     treel.fit(traindata[['duration','poutcome_val']],traindata['Subscribed_val'])
```

```
                 DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3, max_leaf_nodes=10)
```

```
#predictions for tree one using 2 attributes
predictions = treel.predict(testdata[['duration','poutcome_val']])
print(accuracy_score(testdata["Subscribed_val"], predictions))
```

```
0.879751615339431
```

Classification Tree model 2:

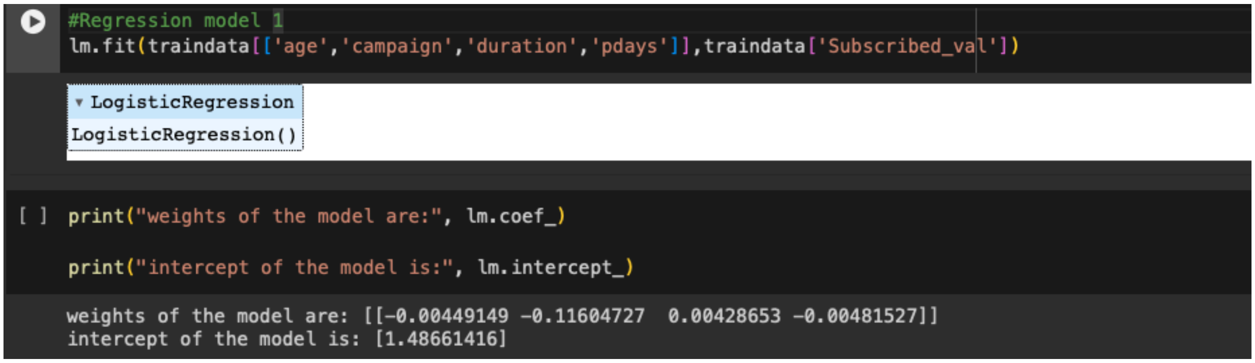


```
[ ]  #Decision Tree #2
     tree2 = tree.DecisionTreeClassifier(max_depth=4, max_leaf_nodes=8)
     tree2.fit(traindata[['campaign','pdays','poutcome_val']],traindata['Subscribed_val'])
     predictions2 = tree2.predict(testdata[['campaign','pdays','poutcome_val']])
     print(accuracy_score(testdata["Subscribed_val"], predictions2))
```

```
0.8928421582613074
```

```
tree.plot_tree(tree2)
```

Regression Model 1:

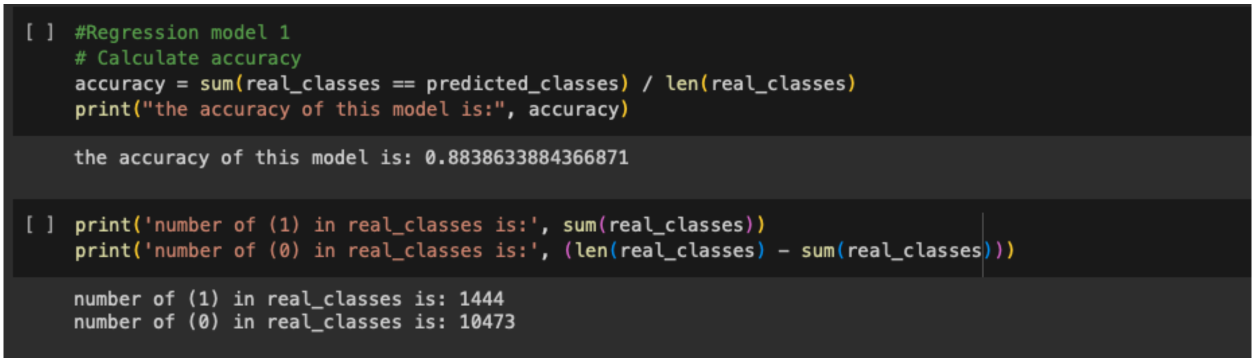


```
#Regression model 1
lm.fit(traindata[['age','campaign','duration','pdays']],traindata['Subscribed_val'])
```

```
▾ LogisticRegression
LogisticRegression()
```

```
print("weights of the model are:", lm.coef_)

print("intercept of the model is:", lm.intercept_)
```

```
weights of the model are: [[-0.00449149 -0.11604727  0.00428653 -0.00481527]]
intercept of the model is: [1.48661416]
```

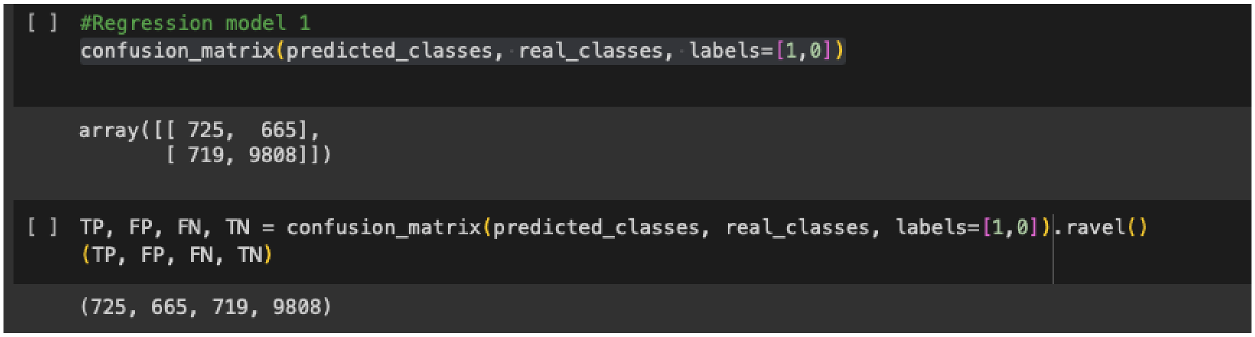


```
#Regression model 1
# Calculate accuracy
accuracy = sum(real_classes == predicted_classes) / len(real_classes)
print("the accuracy of this model is:", accuracy)
```

```
the accuracy of this model is: 0.8838633884366871
```

```
print('number of (1) in real_classes is:', sum(real_classes))
print('number of (0) in real_classes is:', (len(real_classes) - sum(real_classes)))
```

```
number of (1) in real_classes is: 1444
number of (0) in real_classes is: 10473
```



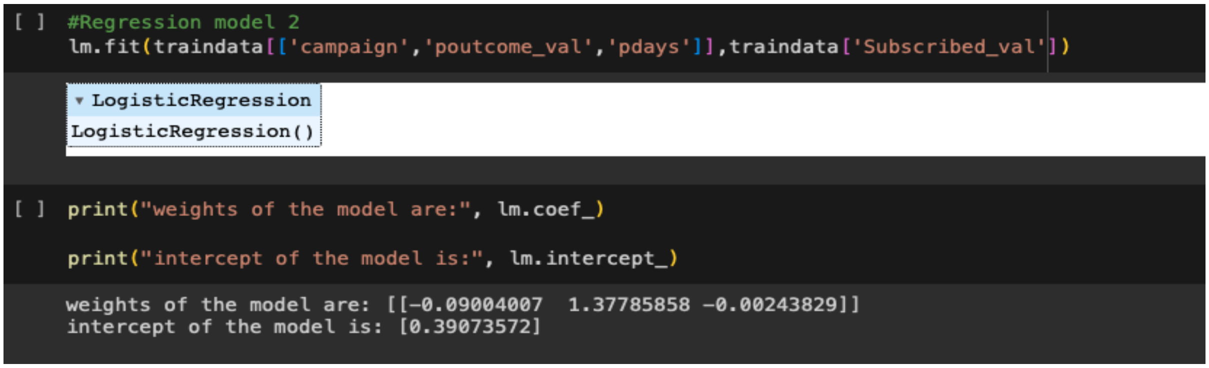
```
#Regression model 1
confusion_matrix(predicted_classes, real_classes, labels=[1,0])
```

```
array([[ 725,  665],
       [ 719, 9808]])
```

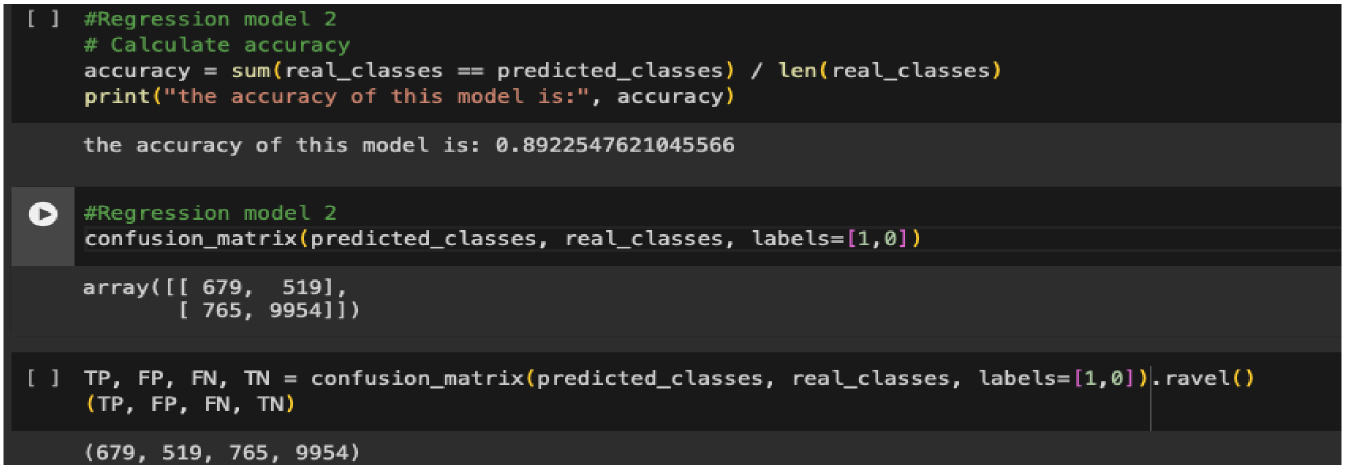
```
TP, FP, FN, TN = confusion_matrix(predicted_classes, real_classes, labels=[1,0]).ravel()
(TP, FP, FN, TN)
```

```
(725, 665, 719, 9808)
```

Regression 2:

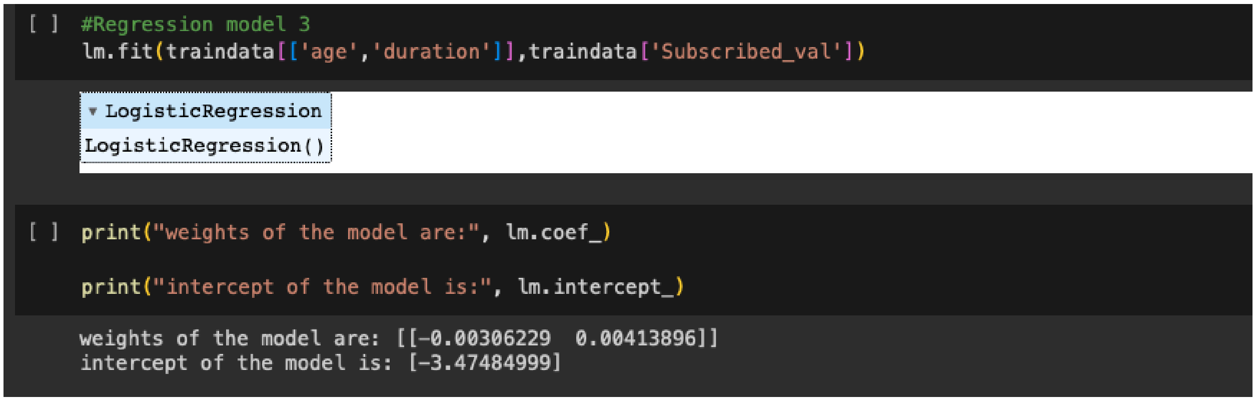


```
[ ] #Regression model 2
    lm.fit(traindata[['campaign','poutcome_val','pdays']],traindata['Subscribed_val'])

    ▾ LogisticRegression
    LogisticRegression()

[ ] print("weights of the model are:", lm.coef_)

    print("intercept of the model is:", lm.intercept_)

    weights of the model are: [[-0.09004007  1.37785858 -0.00243829]]
    intercept of the model is: [0.39073572]
```

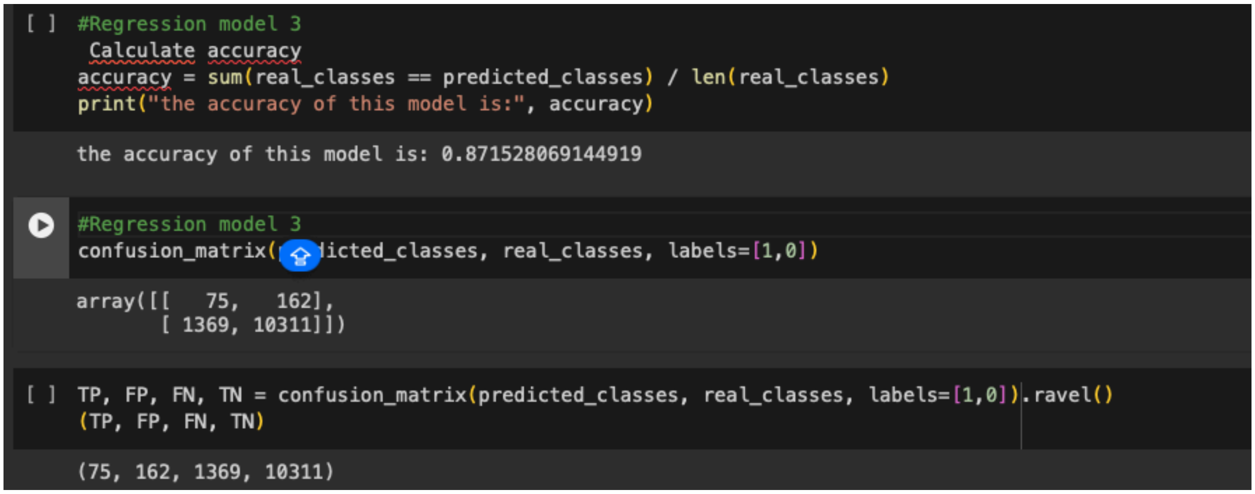


```
[ ] #Regression model 2
    # Calculate accuracy
    accuracy = sum(real_classes == predicted_classes) / len(real_classes)
    print("the accuracy of this model is:", accuracy)

    the accuracy of this model is: 0.8922547621045566

    #Regression model 2
    confusion_matrix(predicted_classes, real_classes, labels=[1,0])

    array([[ 679,  519],
           [ 765, 9954]])

[ ] TP, FP, FN, TN = confusion_matrix(predicted_classes, real_classes, labels=[1,0]).ravel()
    (TP, FP, FN, TN)

    (679, 519, 765, 9954)
```

Regression model 3:



```
[ ] #Regression model 3
    lm.fit(traindata[['age','duration']],traindata['Subscribed_val'])

    ▾ LogisticRegression
    LogisticRegression()

[ ] print("weights of the model are:", lm.coef_)

    print("intercept of the model is:", lm.intercept_)

    weights of the model are: [[-0.00306229  0.00413896]]
    intercept of the model is: [-3.47484999]
```



```
[ ] #Regression model 3
     Calculate accuracy
    accuracy = sum(real_classes == predicted_classes) / len(real_classes)
    print("the accuracy of this model is:", accuracy)

    the accuracy of this model is: 0.871528069144919
```

```
#Regression model 3
confusion_matrix(predicted_classes, real_classes, labels=[1,0])

array([[   75,   162],
       [ 1369, 10311]])
```

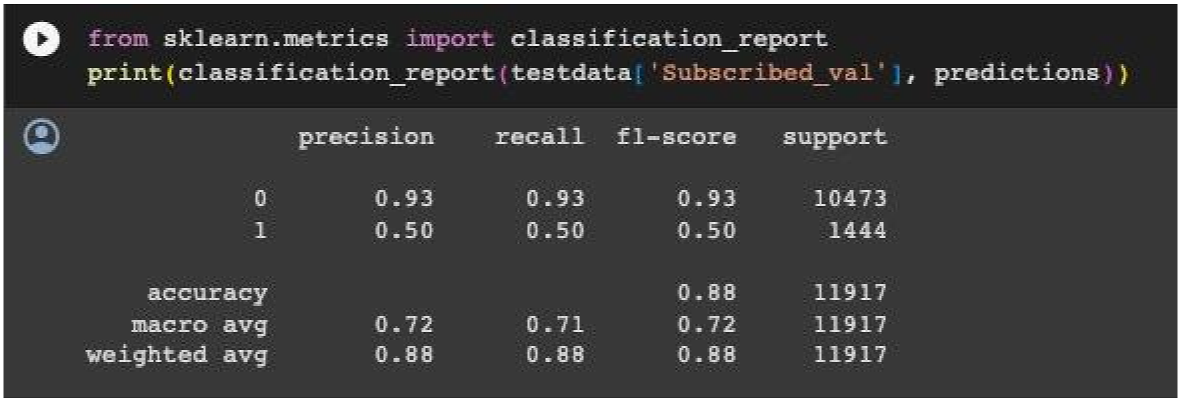
```
[ ] TP, FP, FN, TN = confusion_matrix(predicted_classes, real_classes, labels=[1,0]).ravel()
    (TP, FP, FN, TN)

    (75, 162, 1369, 10311)
```

Tree #	**Attributes**	**Accuracy %**	**Description**
Tree 1	Poutcome_val + Duration	88 %	This combination of attributes provides the second highest accuracy (very accurate)
Tree 2	Campaign + Pdays+ Poutcome_val	89%	This combination of attributes provides the highest accuracy (very accurate)

Regression model#	**Attributes**	**Accuracy %**	**Description**

Model 1	Age + Campaign + Duration + Pdays'.	88%	This combination of attributes provides the second highest accuracy (very accurate)
Model 2	Campaign + Poutcome_val + Pdays	89%	This combination of attributes provides the highest accuracy (very accurate)
Model 3	Age + Duration	87%	This combination of attributes provides the third highest accuracy (very accurate)

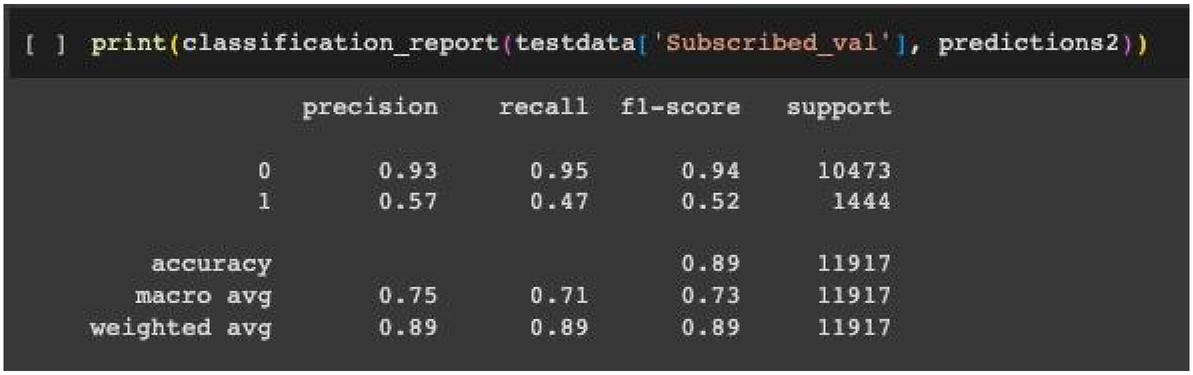
Tree 1 Classification report:



```
from sklearn.metrics import classification_report
print(classification_report(testdata['Subscribed_val'], predictions))
```

```
              precision    recall  f1-score   support

           0       0.93      0.93      0.93     10473
           1       0.50      0.50      0.50      1444

    accuracy                           0.88     11917
   macro avg       0.72      0.71      0.72     11917
weighted avg       0.88      0.88      0.88     11917
```

Tree 2 Classification report:



```
[ ]  print(classification_report(testdata['Subscribed_val'], predictions2))

                 precision    recall  f1-score   support

              0       0.93      0.95      0.94     10473
              1       0.57      0.47      0.52      1444

       accuracy                           0.89     11917
      macro avg       0.75      0.71      0.73     11917
   weighted avg       0.89      0.89      0.89     11917
```

As we can see from the results above, both reports show high accuracy of 88% and 89% based on their targeted attributes.

Discussion:

The main purpose of the designated project is to significantly increase the success rate of the bank's telemarketing campaign. Which has exceeded the expectations and met all the needs in the criteria. This was done through its visualization, data analysis and analytical model building.

After careful evaluation, the findings from whether clients will effectively subscribe to a long term deposit has led us to the results of the accuracy rate being at 12.1 percent whereas the error rate was a wapping 87.9 percent, we were able to come up with these results using a decision tree. Though these were the predictions it did not negatively affect the final results of our objective. In the end there was a true positive rate of 88 percent and 89 percent on their targeted attributes. This has led us to believe that the decision tree was of useful advantage. We found that the decision tree model was easy to implement, clearly shows the data and results in a visual flow, and is easy to understand. It also came to our realization, when the decision tree used to train a model, “nr.employed” had the highest correlation but with that being said, it always had a negative effect on the accuracy which brought it down. As for the Regression Model, we can see the similarities with the Decision Tree with the accuracy outcomes being allied with each other (88 percent and 89 percent). It was helpful to see that it provides us with true positive outcomes that were pulled from the real class which the Regression Model predicted and was easy to follow and understand.

Although, there were not positive rates in regards to the training data. There was also a target customer profile that was installed while regression and classification models were used to implement the predictability of all customer responses to the term deposit campaign.

Conclusion:

In conclusion, the main problem is related to the direct marketing campaigns (phone calls) of a Portuguese Banking Institution which became the classification goal to predict whether the clients of the bank subscribed to a term deposit or not. Once the data was collected and cleaned, we then proceeded to analyze the dataset. We used two methods to help us predict whether the clients subscribed a term deposit or not, Decisions Tree and Logistic Regression. For the Decisions Tree, we thought it made the most sense to use this model to run the test because we were dealing with the binary target attributes and supervised data which classification does work best with this dataset. To be as accurate as possible, we decided to make 3 different models of the Decisions Tree to make sure of that. For the Logistic Regression, we used it so we can interpret the coefficients so we can understand the feature of the probability on a certain class. Just like for the Decision Tree, we made 3 different models for the Regression model as well for the training set and analyzed the model for the 3 different models to get us the most accurate outcome possible. Using these 2 methods, we concluded that both models produced the same output rate when we evaluated it, with the rates being 88 percent and 89 percent. Therefore, we believe that this assessment turned out to be a good evaluation because we could see the similarities between both models and achieved very accurate results.