EDA and Building machine learning model for Santander Customer Satisfaction Data

Siri Kademani

Agenda

Introduction

Understanding Dataset

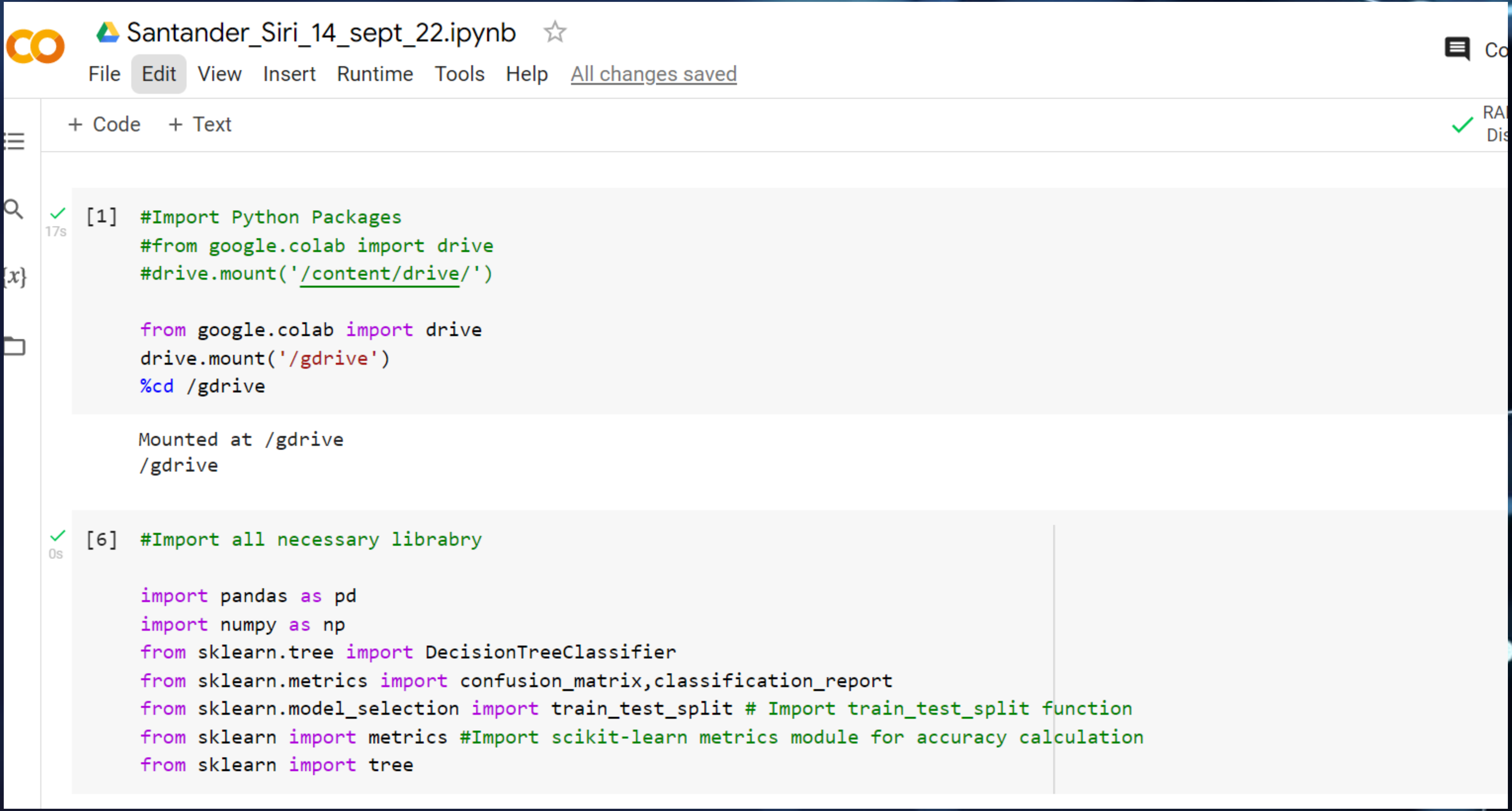
Prediction using Decision Tree Classifier

Exploratory Data Analysis

Introduction

- Exploratory Data Analysis (EDA), trying to understand the provided data and to find potential relationships between variables.

Connecting to Google drive and importing the libraries

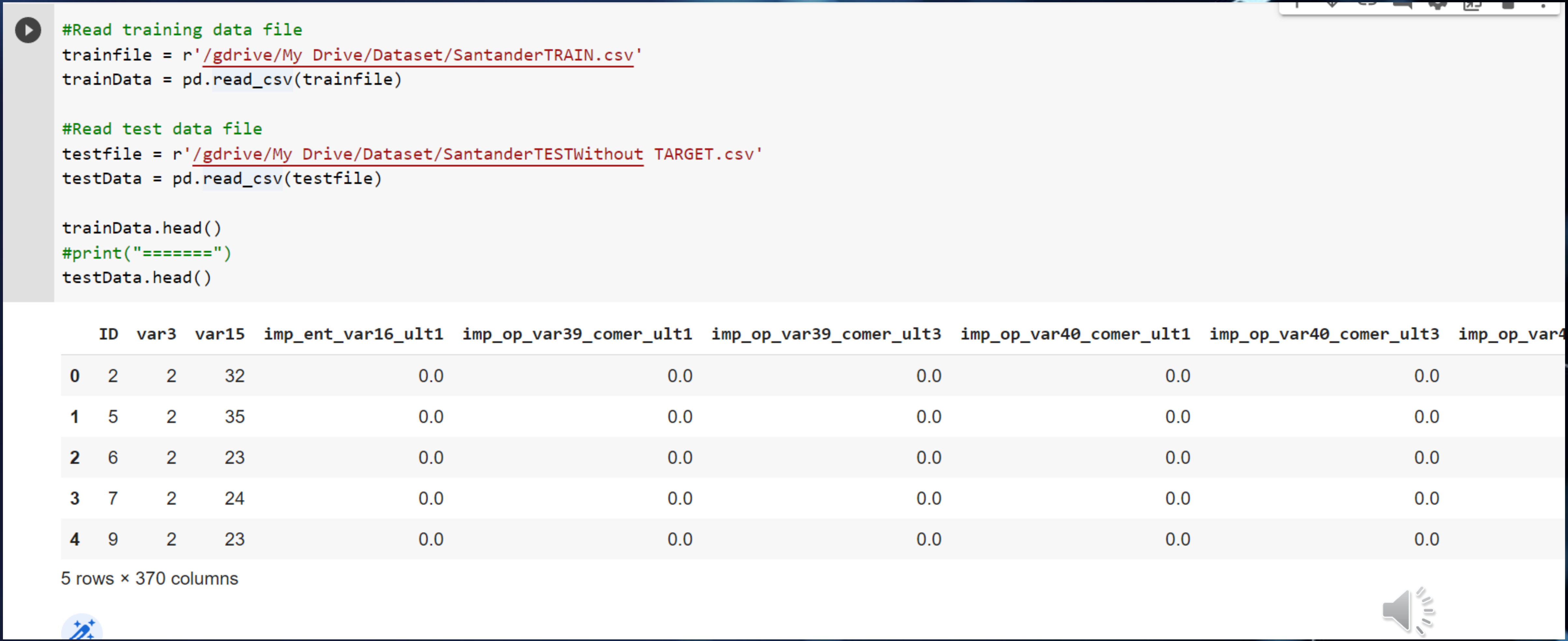


```
#Import Python Packages
#from google.colab import drive
#drive.mount('/content/drive/')

from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive
```

```
Mounted at /gdrive
/gdrive
```

```
#Import all necessary librabry

import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
from sklearn import tree
```

Importing Train and Test Dataset

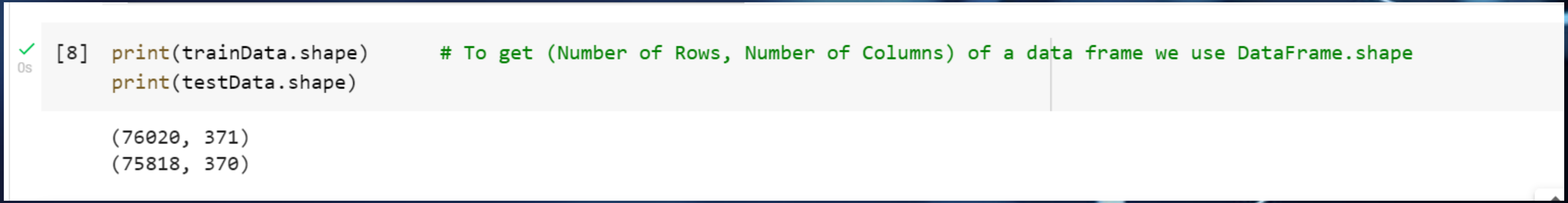


```
#Read training data file
trainfile = r'/gdrive/My Drive/Dataset/SantanderTRAIN.csv'
trainData = pd.read_csv(trainfile)

#Read test data file
testfile = r'/gdrive/My Drive/Dataset/SantanderTESTWithout TARGET.csv'
testData = pd.read_csv(testfile)

trainData.head()
#print("=======")
testData.head()
```

	ID	var3	var15	imp_ent_var16_ult1	imp_op_var39_comer_ult1	imp_op_var39_comer_ult3	imp_op_var40_comer_ult1	imp_op_var40_comer_ult3	imp_op_var4
0	2	2	32	0.0	0.0	0.0	0.0	0.0	
1	5	2	35	0.0	0.0	0.0	0.0	0.0	
2	6	2	23	0.0	0.0	0.0	0.0	0.0	
3	7	2	24	0.0	0.0	0.0	0.0	0.0	
4	9	2	23	0.0	0.0	0.0	0.0	0.0	

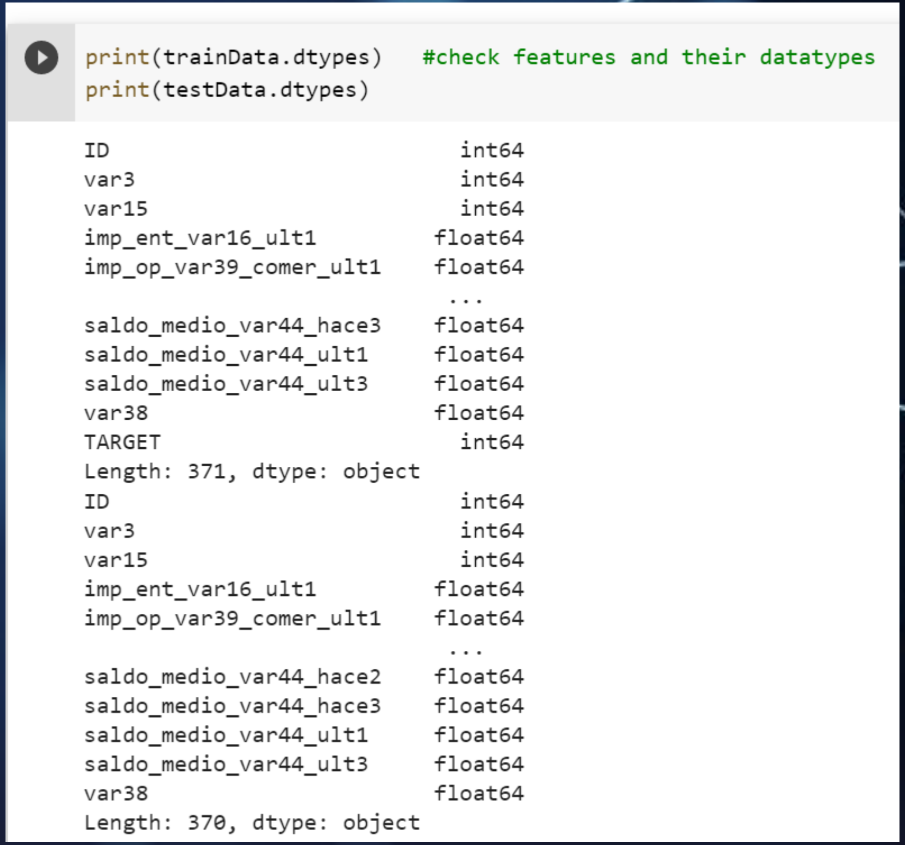
5 rows × 370 columns

Getting count of number of rows and columns from Train and test Data set



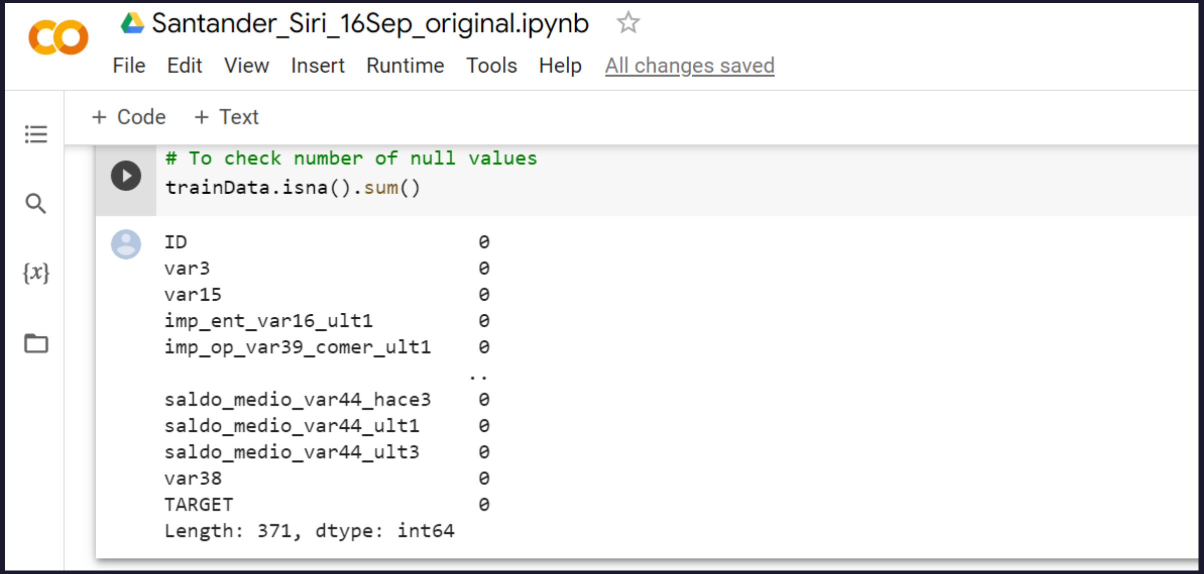
```
[8]  print(trainData.shape)      # To get (Number of Rows, Number of Columns) of a data frame we use DataFrame.shape
     print(testData.shape)

(76020, 371)
(75818, 370)
```

Checking the data types of rows in each dataset



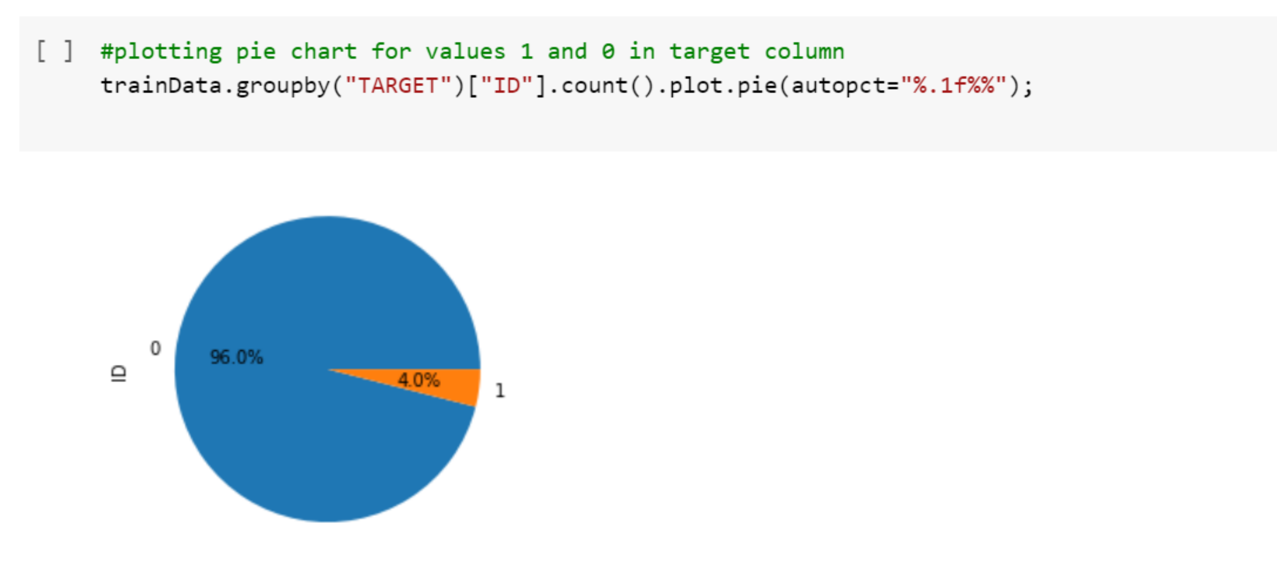
```
print(trainData.dtypes)   #check features and their datatypes
print(testData.dtypes)

ID                             int64
var3                           int64
var15                          int64
imp_ent_var16_ult1           float64
imp_op_var39_comer_ult1      float64
                              ...
saldo_medio_var44_hace3      float64
saldo_medio_var44_ult1       float64
saldo_medio_var44_ult3       float64
var38                        float64
TARGET                         int64
Length: 371, dtype: object
ID                             int64
var3                           int64
var15                          int64
imp_ent_var16_ult1           float64
imp_op_var39_comer_ult1      float64
                              ...
saldo_medio_var44_hace2      float64
saldo_medio_var44_hace3      float64
saldo_medio_var44_ult1       float64
saldo_medio_var44_ult3       float64
var38                        float64
Length: 370, dtype: object
```

Checking for any null values in each column of train data


Santander_Siri_16Sep_original.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
# To check number of null values
trainData.isna().sum()
```

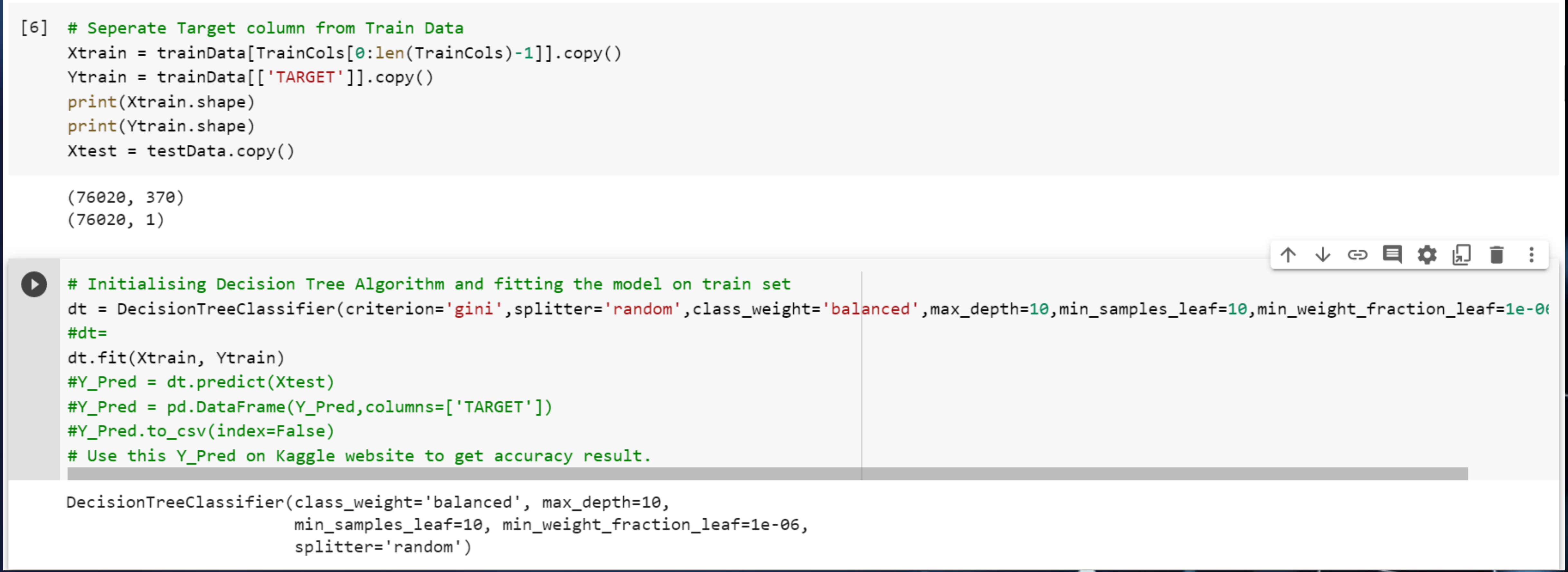
```
ID                          0
var3                        0
var15                       0
imp_ent_var16_ult1          0
imp_op_var39_comer_ult1     0
                           ..
saldo_medio_var44_hace3     0
saldo_medio_var44_ult1      0
saldo_medio_var44_ult3      0
var38                       0
TARGET                      0
Length: 371, dtype: int64
```



```
[ ] #plotting pie chart for values 1 and 0 in target column
    trainData.groupby("TARGET")["ID"].count().plot.pie(autopct="%.1f%%");
```

Pie chart representing the % of satisfied and dissatisfied Santander customers.

Decision tree algorithm and fitting the model on train Data



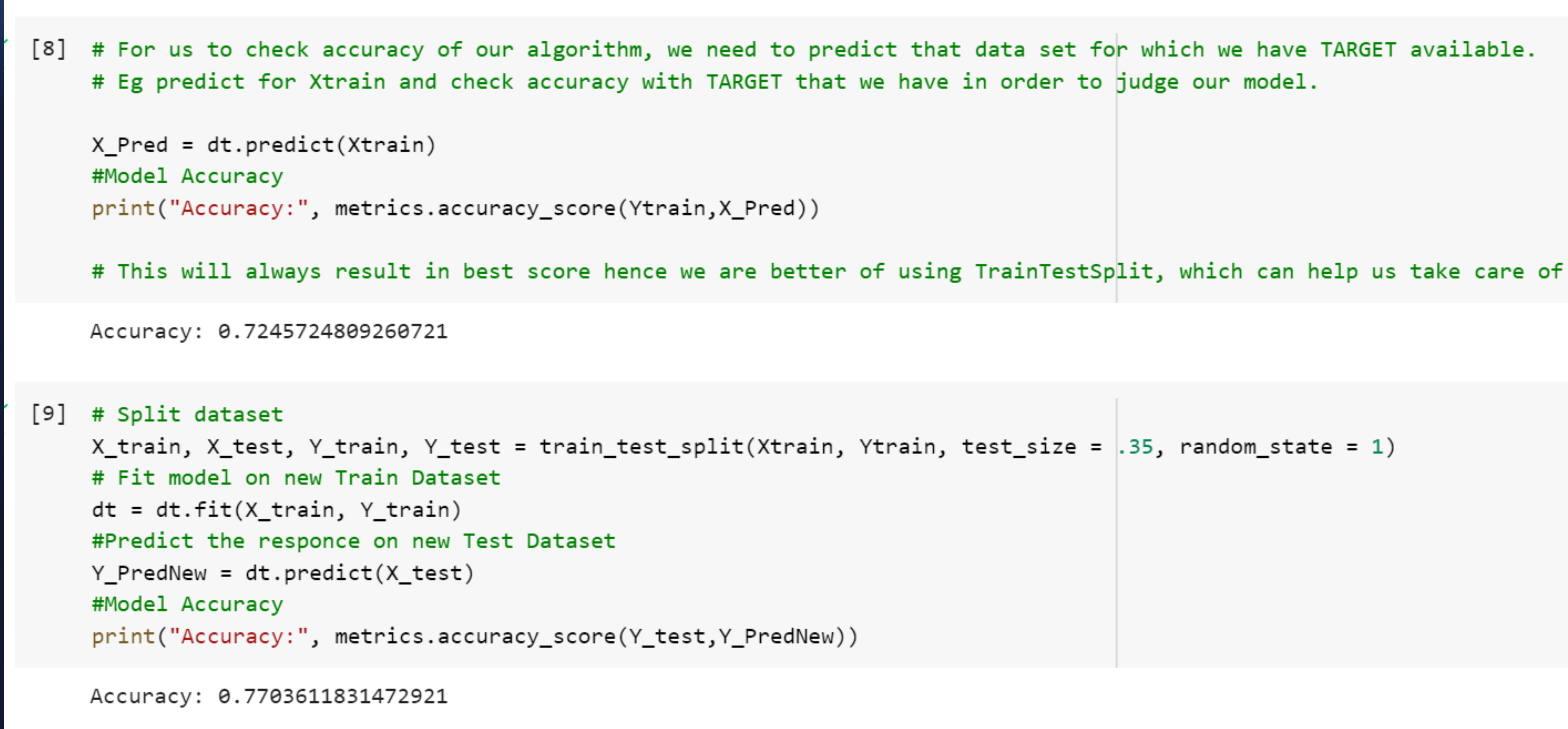
```
[6]  # Seperate Target column from Train Data
     Xtrain = trainData[TrainCols[0:len(TrainCols)-1]].copy()
     Ytrain = trainData[['TARGET']].copy()
     print(Xtrain.shape)
     print(Ytrain.shape)
     Xtest = testData.copy()
```

```
(76020, 370)
(76020, 1)
```

```
# Initialising Decision Tree Algorithm and fitting the model on train set
dt = DecisionTreeClassifier(criterion='gini',splitter='random',class_weight='balanced',max_depth=10,min_samples_leaf=10,min_weight_fraction_leaf=1e-0
#dt=
dt.fit(Xtrain, Ytrain)
#Y_Pred = dt.predict(Xtest)
#Y_Pred = pd.DataFrame(Y_Pred,columns=['TARGET'])
#Y_Pred.to_csv(index=False)
# Use this Y_Pred on Kaggle website to get accuracy result.
```

```
DecisionTreeClassifier(class_weight='balanced', max_depth=10,
                       min_samples_leaf=10, min_weight_fraction_leaf=1e-06,
                       splitter='random')
```

Splitting the train dataset into train and test data and predicting the accuracy score.



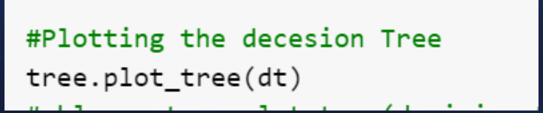
```
[8]  # For us to check accuracy of our algorithm, we need to predict that data set for which we have TARGET available.
     # Eg predict for Xtrain and check accuracy with TARGET that we have in order to judge our model.

     X_Pred = dt.predict(Xtrain)
     #Model Accuracy
     print("Accuracy:", metrics.accuracy_score(Ytrain,X_Pred))

     # This will always result in best score hence we are better of using TrainTestSplit, which can help us take care of
```

```
Accuracy: 0.7245724809260721
```

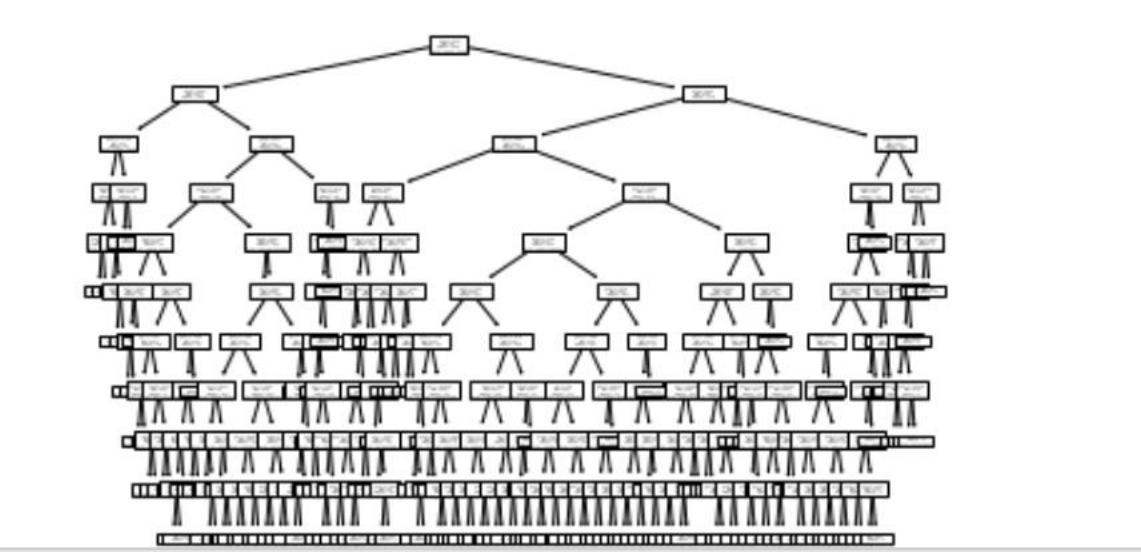
```
[9]  # Split dataset
     X_train, X_test, Y_train, Y_test = train_test_split(Xtrain, Ytrain, test_size = .35, random_state = 1)
     # Fit model on new Train Dataset
     dt = dt.fit(X_train, Y_train)
     #Predict the responce on new Test Dataset
     Y_PredNew = dt.predict(X_test)
     #Model Accuracy
     print("Accuracy:", metrics.accuracy_score(Y_test,Y_PredNew))
```

```
Accuracy: 0.7703611831472921
```

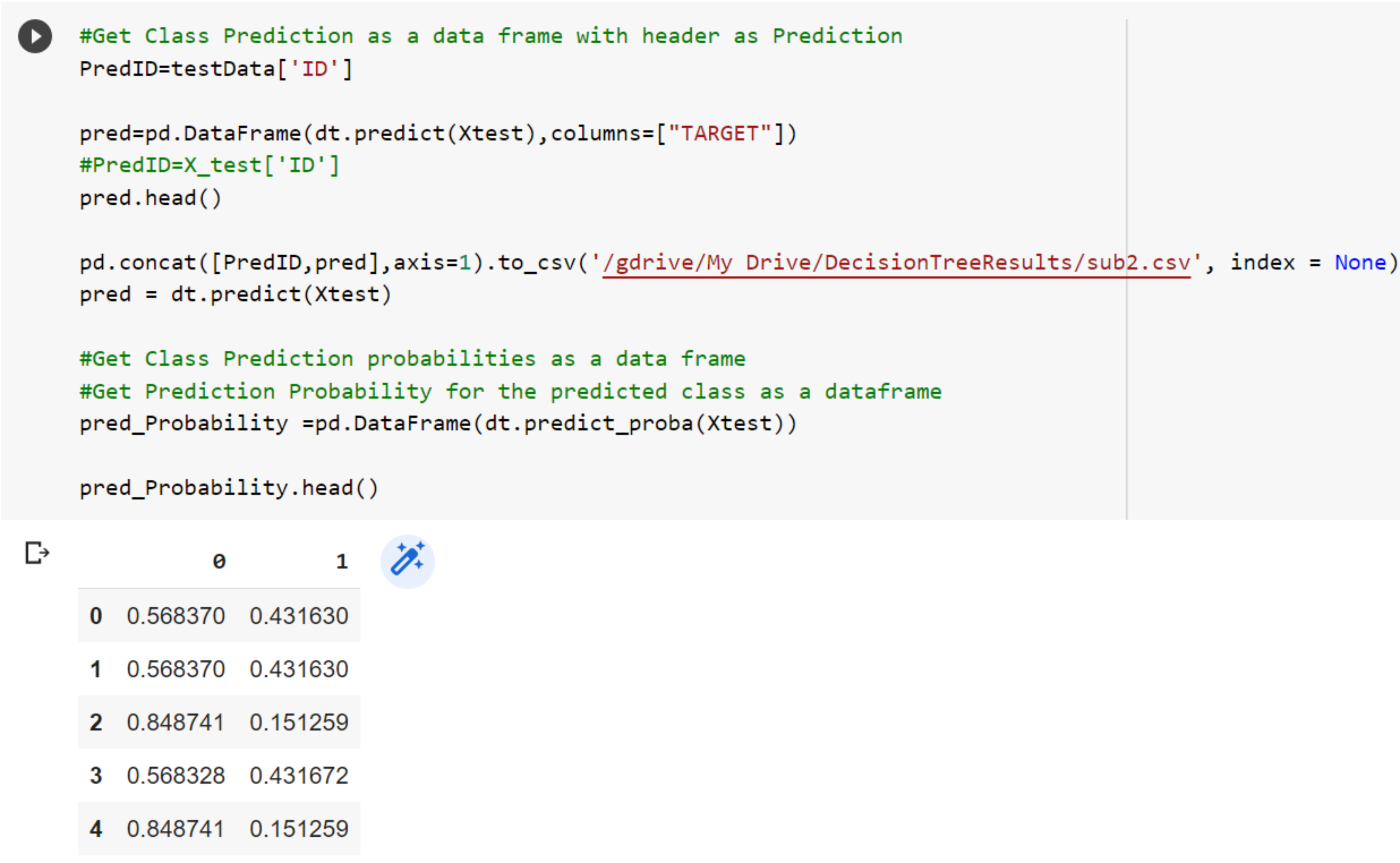
Plotting the decision tree



```
#Plotting the decesion Tree
tree.plot_tree(dt)
```



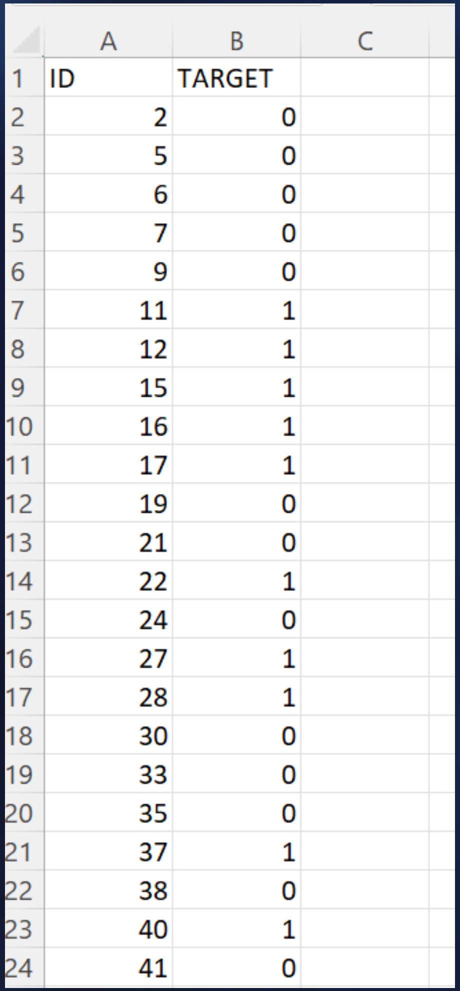
Getting the class prediction using test data and fetching the results into sub2.csv file



```
#Get Class Prediction as a data frame with header as Prediction
PredID=testData['ID']

pred=pd.DataFrame(dt.predict(Xtest),columns=["TARGET"])
#PredID=X_test['ID']
pred.head()

pd.concat([PredID,pred],axis=1).to_csv('/gdrive/My Drive/DecisionTreeResults/sub2.csv', index = None)
pred = dt.predict(Xtest)

#Get Class Prediction probabilities as a data frame
#Get Prediction Probability for the predicted class as a dataframe
pred_Probability =pd.DataFrame(dt.predict_proba(Xtest))

pred_Probability.head()
```

```
          0         1
0  0.568370  0.431630
1  0.568370  0.431630
2  0.848741  0.151259
3  0.568328  0.431672
4  0.848741  0.151259
```

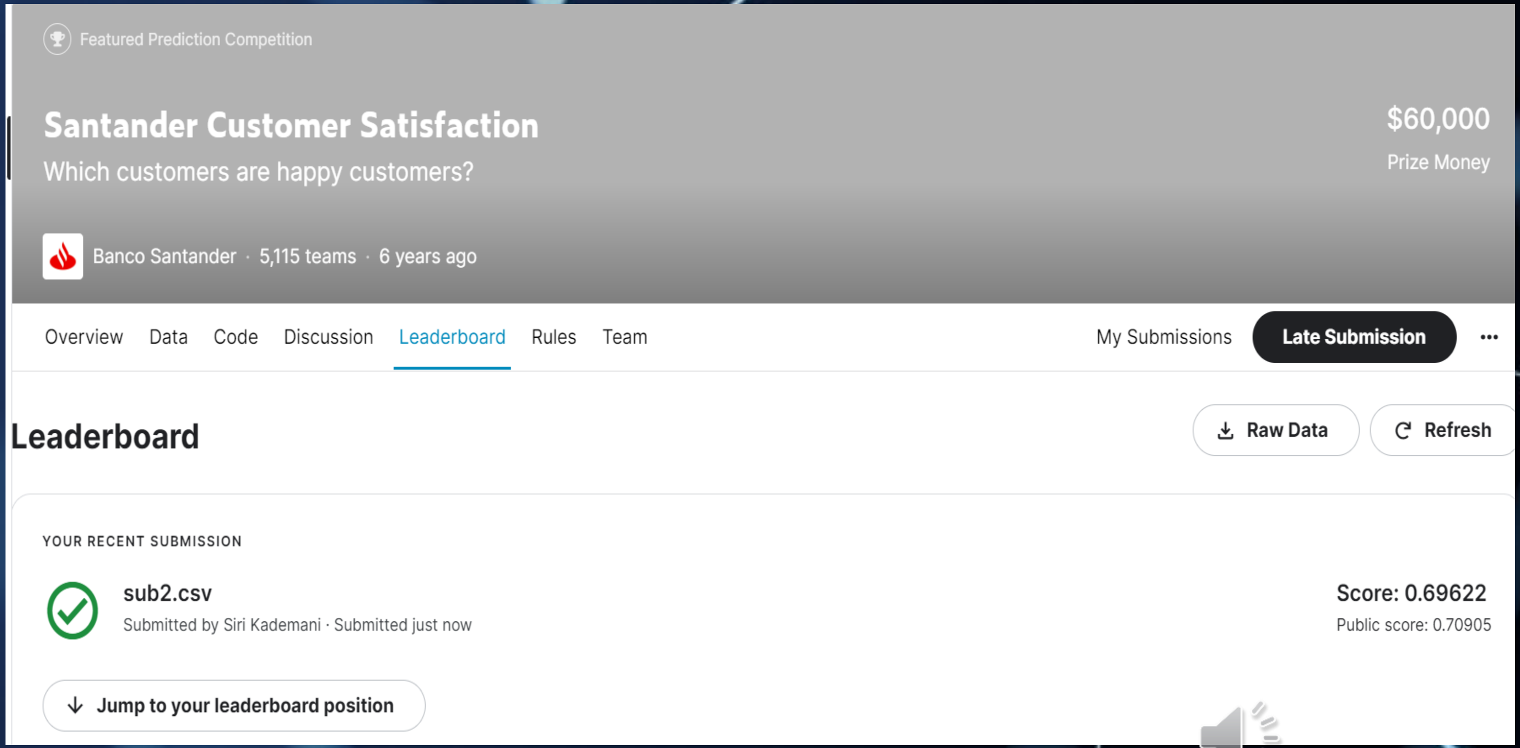
Result file- sub2.csv



	A	B	C
1	ID	TARGET	
2	2	0	
3	5	0	
4	6	0	
5	7	0	
6	9	0	
7	11	1	
8	12	1	
9	15	1	
10	16	1	
11	17	1	
12	19	0	
13	21	0	
14	22	1	
15	24	0	
16	27	1	
17	28	1	
18	30	0	
19	33	0	
20	35	0	
21	37	1	
22	38	0	
23	40	1	
24	41	0	

Snapshot of Kaggle submission score

Score is – 0.6962



Featured Prediction Competition

Santander Customer Satisfaction

Which customers are happy customers?

$60,000
Prize Money

Banco Santander · 5,115 teams · 6 years ago

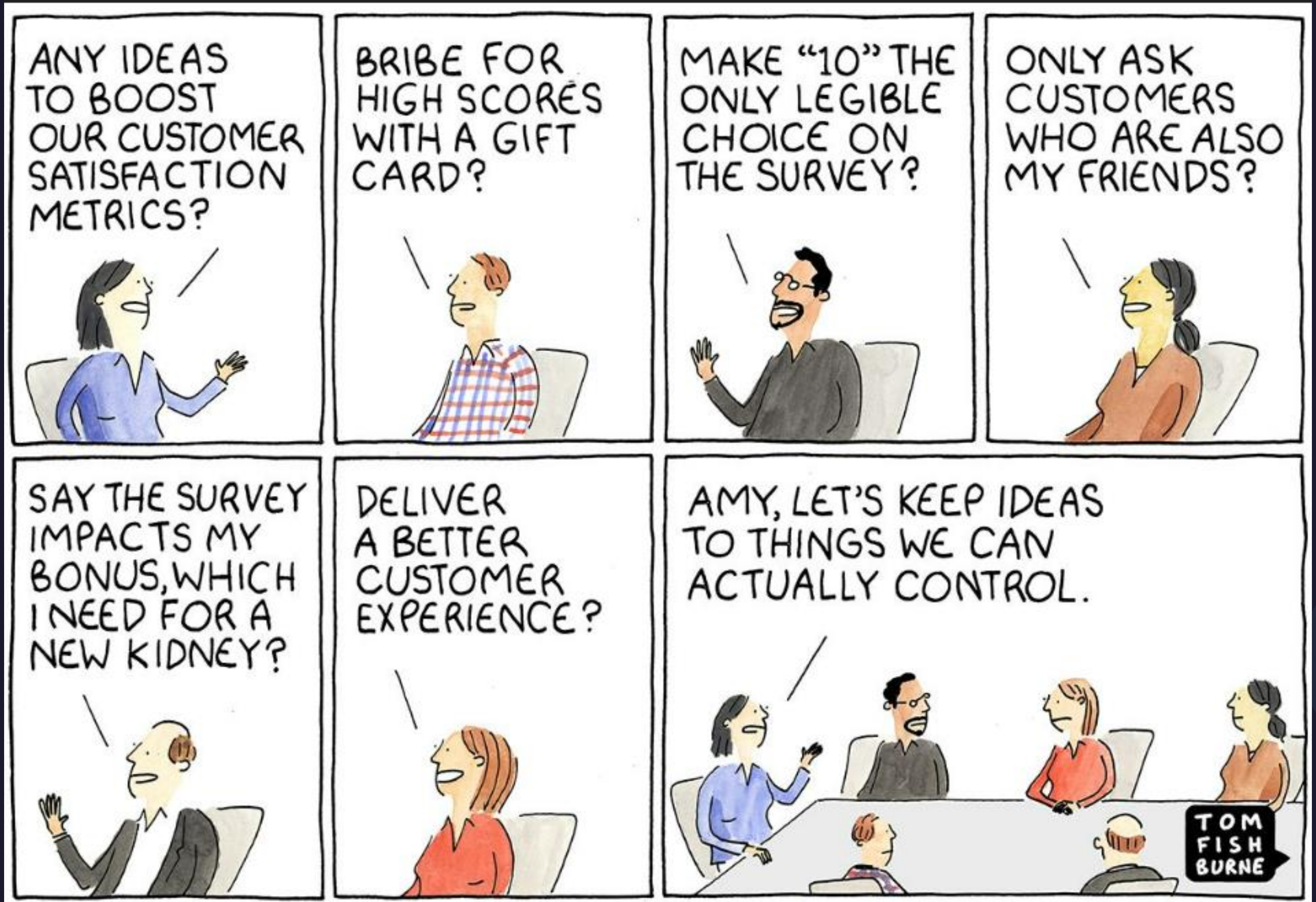
Leaderboard

YOUR RECENT SUBMISSION

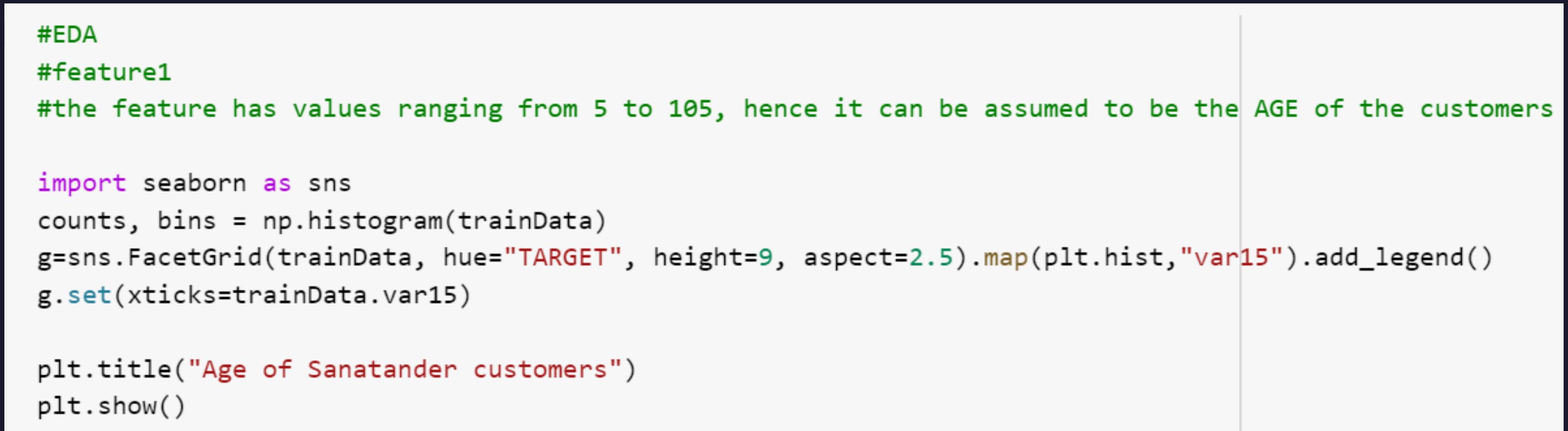
sub2.csv
Submitted by Siri Kademani · Submitted just now

Score: 0.69622
Public score: 0.70905

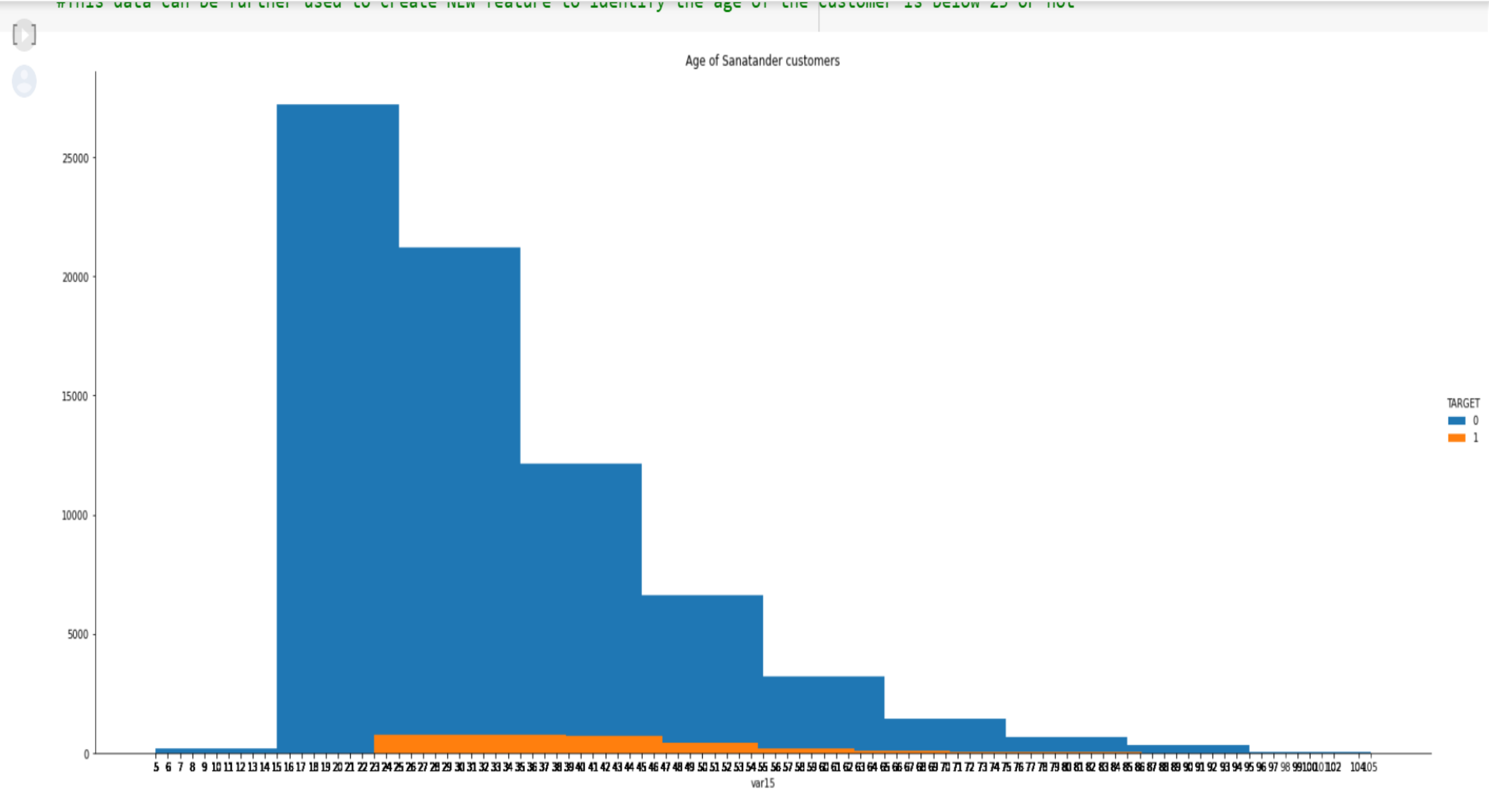
EDA (Exploratory Data Analysis)


ANY IDEAS TO BOOST OUR CUSTOMER SATISFACTION METRICS?
BRIBE FOR HIGH SCORES WITH A GIFT CARD?
MAKE "10" THE ONLY LEGIBLE CHOICE ON THE SURVEY?
ONLY ASK CUSTOMERS WHO ARE ALSO MY FRIENDS?
SAY THE SURVEY IMPACTS MY BONUS, WHICH I NEED FOR A NEW KIDNEY?
DELIVER A BETTER CUSTOMER EXPERIENCE?
AMY, LET'S KEEP IDEAS TO THINGS WE CAN ACTUALLY CONTROL.
TOM FISHBURNE

EDA feature 1



```
#EDA
#feature1
#the feature has values ranging from 5 to 105, hence it can be assumed to be the AGE of the customers

import seaborn as sns
counts, bins = np.histogram(trainData)
g=sns.FacetGrid(trainData, hue="TARGET", height=9, aspect=2.5).map(plt.hist,"var15").add_legend()
g.set(xticks=trainData.var15)

plt.title("Age of Sanatander customers")
plt.show()
```



```
#This data can be further used to create NEW feature to identify the age of the customer is below 23 or not
```

Field name: var15

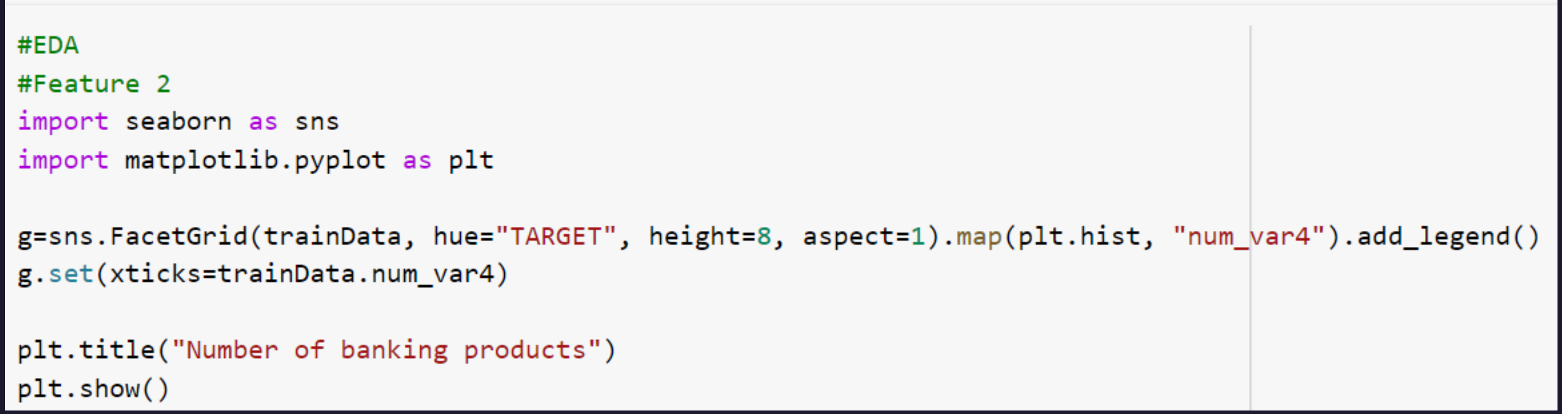
Range of values : 5 to 105

Probable column: Age of customers
('var15' feature according to some literature is believed to be age of the customers)

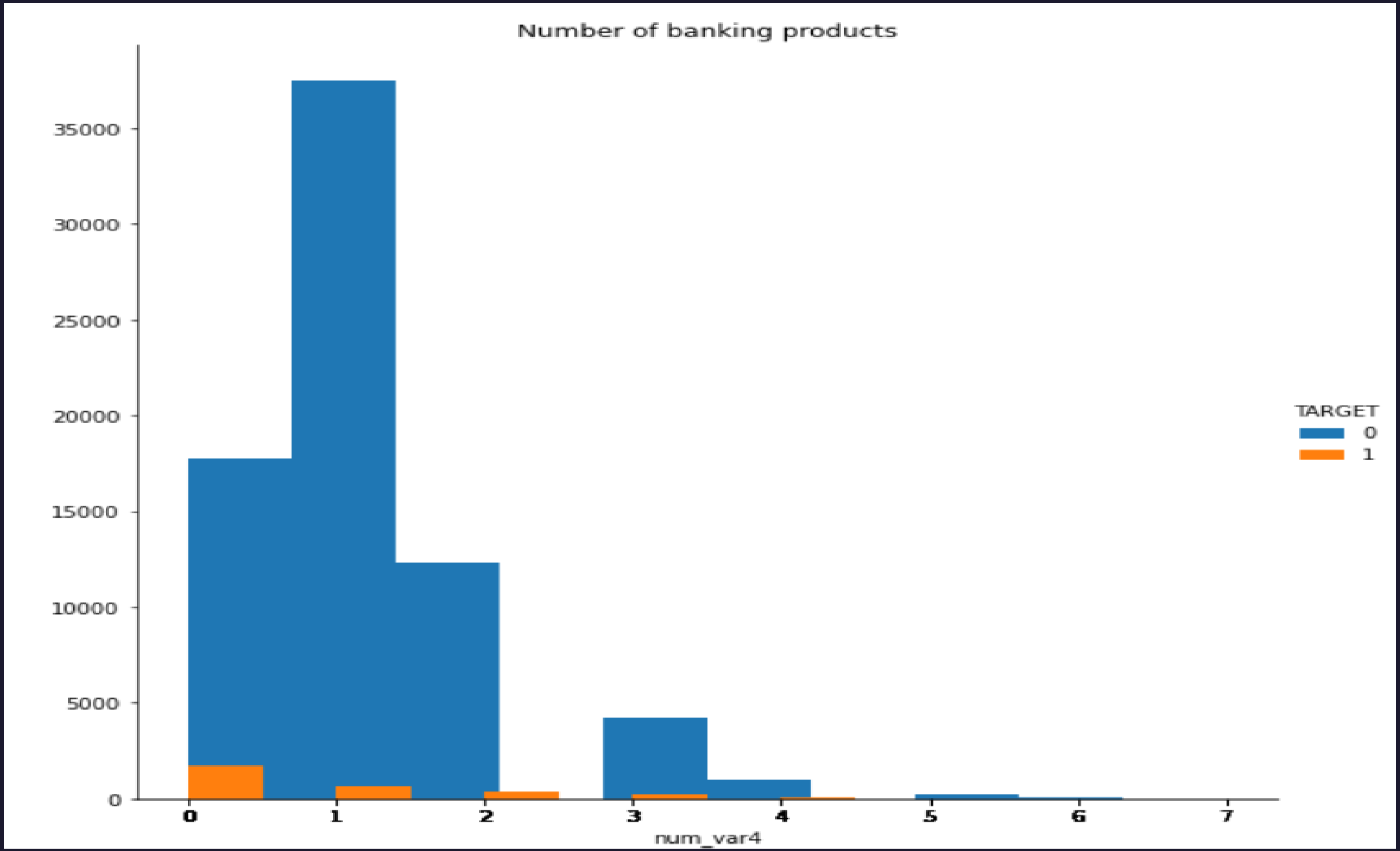
ANALYSIS :

From the shown histogram, we can see that most of the customers below age 23 are dissatisfied.
This data can be further used to create NEW feature to identify the age of the customer is below 23 or not.

EDA feature 2



```
#EDA
#Feature 2
import seaborn as sns
import matplotlib.pyplot as plt

g=sns.FacetGrid(trainData, hue="TARGET", height=8, aspect=1).map(plt.hist, "num_var4").add_legend()
g.set(xticks=trainData.num_var4)

plt.title("Number of banking products")
plt.show()
```

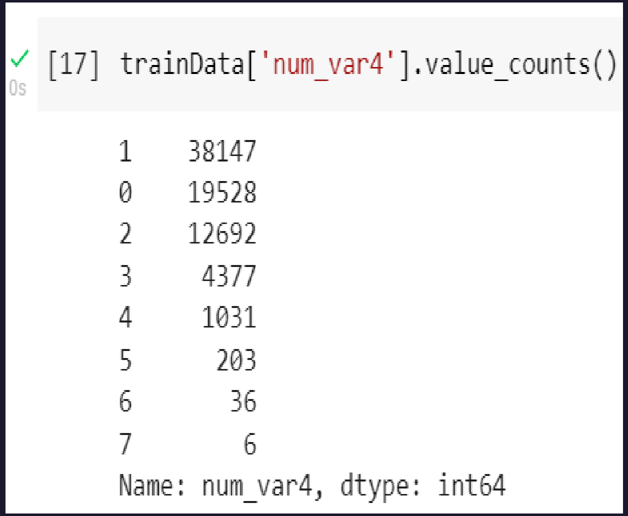


Field name: num_var4

Range of values : 0 to 9

Probable column: Number of banking products ('num_var4' feature according to some literature is believed to be number of banking products bought by each customer)

EDA feature 2



```
✓ [17] trainData['num_var4'].value_counts()
0s

      1    38147
      0    19528
      2    12692
      3     4377
      4     1031
      5      203
      6       36
      7        6
      Name: num_var4, dtype: int64
```

Field name: num_var4

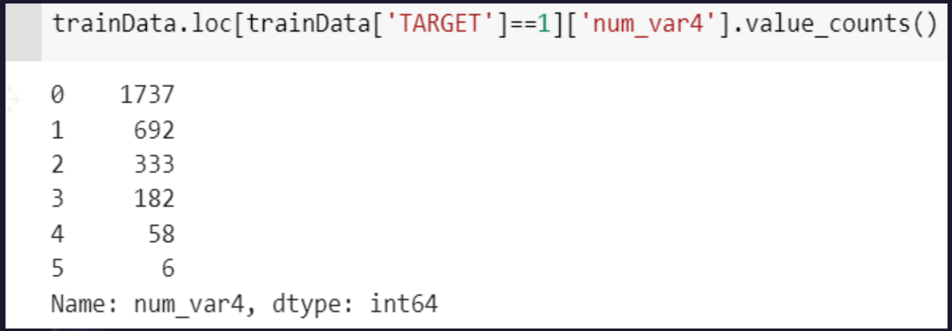
Range of values : 0 to 9

Probable column: Number of banking products

ANALYSIS :

From the given output, we can analyze that the most preferred banking product is 1 and the least preferred banking products are 6 and 7.

EDA feature 3



```
trainData.loc[trainData['TARGET']==1]['num_var4'].value_counts()

0    1737
1     692
2     333
3     182
4      58
5       6
Name: num_var4, dtype: int64
```

Banking products bought by unsatisfied customers

ANALYSIS :

Number of banking products 6 and 7 didn't make it in the series below.
So, customers who have 6 or more than 6 banking products with the bank are all satisfied.
This could be used to make a new feature while feature engineering whether value of 'num_var4' is 6 or more than 6.

Thank You

Siri Kademani

skademan@asu.edu