ANTIM 2021ume0204@iitjammu.ac.in **IIT JAMMU**

LINK FOR THE COOLAB FILE : https://colab.research.google.com/drive/1g5v8JE_pGZlTp0-zruGiPZPOoqTn93eX?usp=sharing

Machine Learning Intern Assessment Assignment

Customer Churn Prediction

TASKS:

- Analysed the given customer dataset.
- Made the proper visualization of the dataset using pie chart, bar graphs, histograms, curves etc
- DATA Preprocessing
- Developed a machine learning model to predict customer churn based on historical customer data.
- Model Evaluation And Prediction

Data: In this assignment we were provided with a dataset containing historical customer information, including customer attributes

- Customer ID ,
- Name , Age ,
- Gender ,
- Location,
- Subscription_Length_Months ,
- Monthly_Bill ,
- Total_Usage_GB ,
- Churn.

Description of Work:

1: **DATA LOADING:** First we converted the xlsx format data into csv format. Then converted the dataset into pandas dataframe. We got the understanding of the data using various python libraries pandas , numpy , missingno , seaborn , plotly , sklearn , xgboost , catboost.

All the dependencies are mentioned and imported under the Dependencies Section.

2: **DATA VISUALIZATION**: The basic information about the dataset was retrieved by printing shape of the data, attributes , values of the attributes and many statistical measures were analysed

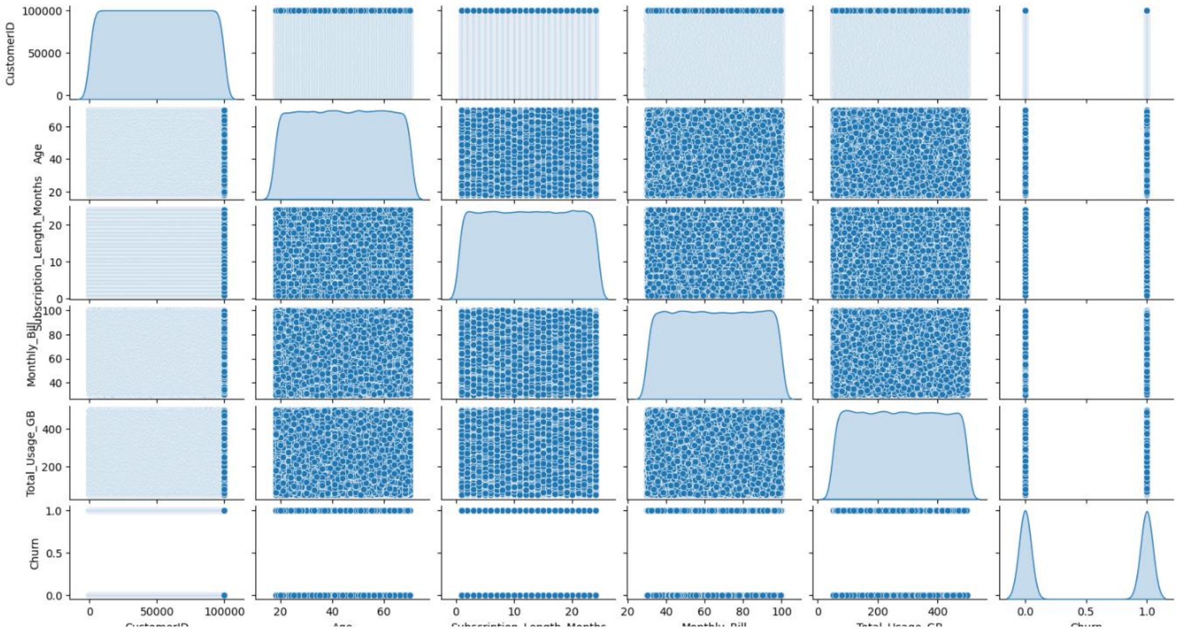
All the basic visualization of the data is performed in the first cell of the VISUALIZING THE DATA section.

For more Illustration and Analysis many plots are also provided in the subsequent cells of this section.

It includes Pi chart, Bar graph, Histograms, Curves plotted between many pair of attributes in the Customer Data set.

A Correlation plot between various Attributes was also analysed and I found out that no Two different attributes were following any standard relation i.e. No correlation was found between the Attributes.

Here is the Figure which I analysed.



As we can observe there is not any nice curve between the pair of attributes except the diagonal elements of the plot.

Which is also affecting the accuracy of the model.

3: **DATA PREPROCESSING**:

Under this Section I performed the following Tasks:

- Wrote a function to convert the column with dtype=’object’ to int using the LabelEncoder() function.
- Wrote a function to Split the Input data into Training and Test Data with Test size of 30% and applying the Standard Scaler Transform .
- Wrote a function to displaying the Distribution plot of the features.

4: **MODEL TRAINING AND EVALUATION:**

I tried to fit various classifiers on this Dataset and trained the model,namely

KNN classifier,AdaBoost,Random Forest, Logistic Regression,SVM,SVC;

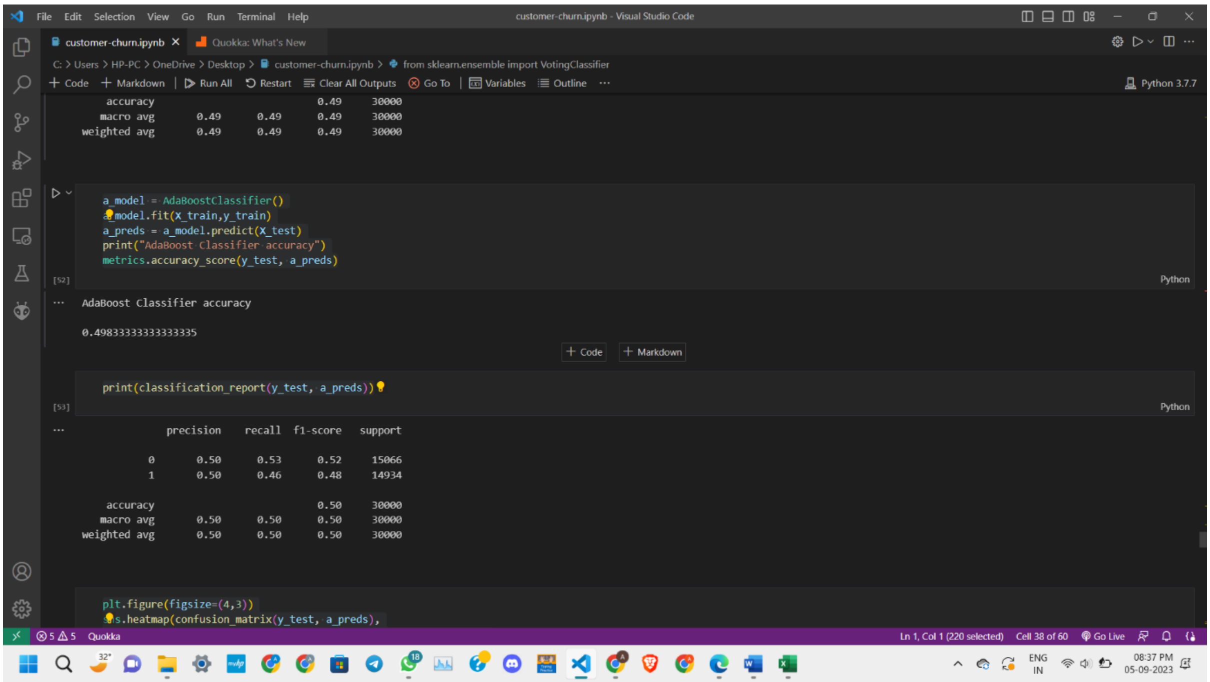
ANTIM 2021ume0204@iitjammu.ac.in **IIT JAMMU**

LINK FOR THE COOLAB FILE : https://colab.research.google.com/drive/1g5v8JE_pGZlTp0-zruGiPZPOoqTn93eX?usp=sharing

Out of all these classifiers None of them was giving the accuracy of more than 50 % ;

But the Support Vector Classifier was giving the best accuracy of 51% , so I used that classifier in my Model.

Here are attachments of the obtained accuracies for the different Classifiers:



```
     accuracy                           0.49     30000
    macro avg       0.49      0.49      0.49     30000
 weighted avg       0.49      0.49      0.49     30000
```

```
a_model = AdaBoostClassifier()
a_model.fit(X_train,y_train)
a_preds = a_model.predict(X_test)
print("AdaBoost Classifier accuracy")
metrics.accuracy_score(y_test, a_preds)
```

[52]

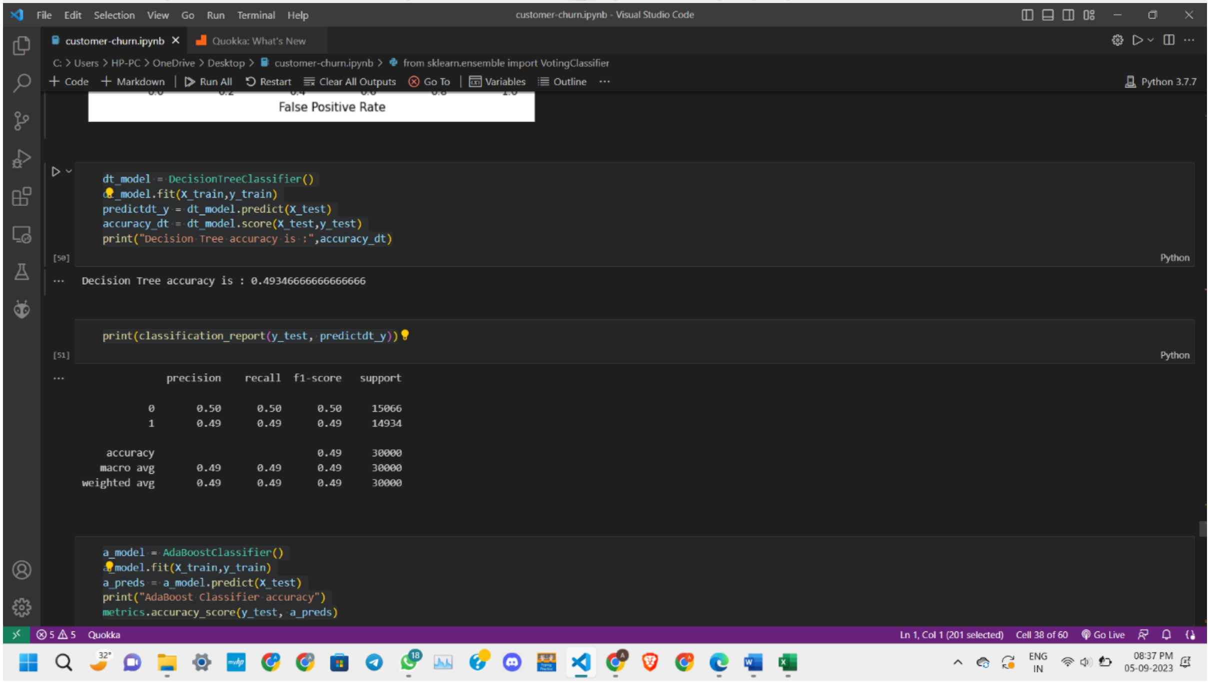
```
AdaBoost Classifier accuracy

0.49833333333333335
```

```
print(classification_report(y_test, a_preds))
```

[53]

```
              precision    recall  f1-score   support

           0       0.50      0.53      0.52     15066
           1       0.50      0.46      0.48     14934

     accuracy                           0.50     30000
    macro avg       0.50      0.50      0.50     30000
 weighted avg       0.50      0.50      0.50     30000
```

```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, a_preds),
```



```
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train,y_train)
predictdt_y = dt_model.predict(X_test)
accuracy_dt = dt_model.score(X_test,y_test)
print("Decision Tree accuracy is :",accuracy_dt)
```

[50]

```
Decision Tree accuracy is : 0.49346666666666666
```

```
print(classification_report(y_test, predictdt_y))
```

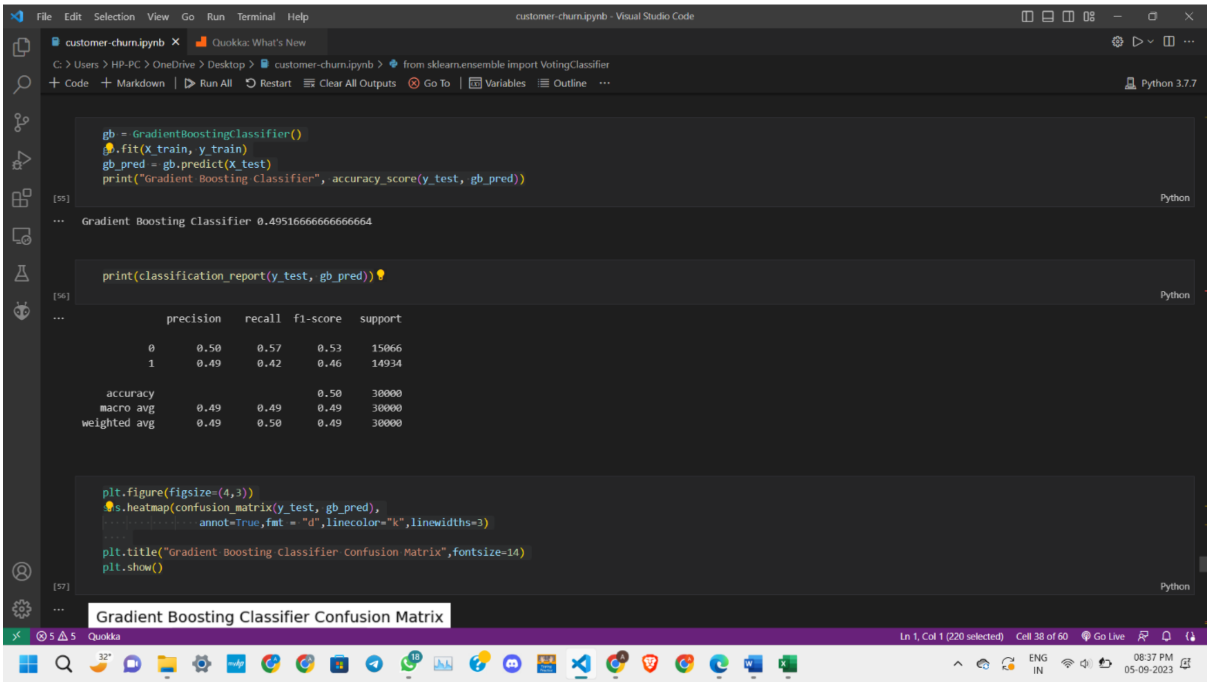
[51]

```
              precision    recall  f1-score   support

           0       0.50      0.50      0.50     15066
           1       0.49      0.49      0.49     14934

     accuracy                           0.49     30000
    macro avg       0.49      0.49      0.49     30000
 weighted avg       0.49      0.49      0.49     30000
```

```
a_model = AdaBoostClassifier()
a_model.fit(X_train,y_train)
a_preds = a_model.predict(X_test)
print("AdaBoost Classifier accuracy")
metrics.accuracy_score(y_test, a_preds)
```

ANTIM 2021ume0204@iitjammu.ac.in **IIT JAMMU**

LINK FOR THE COOLAB FILE : https://colab.research.google.com/drive/1g5v8JE_pGZlTp0-zruGiPZPOoqTn93eX?usp=sharing



File Edit Selection View Go Run Terminal Help
customer-churn.ipynb - Visual Studio Code
customer-churn.ipynb
Quokka: What's New
C: > Users > HP-PC > OneDrive > Desktop > customer-churn.ipynb > from sklearn.ensemble import VotingClassifier
+ Code + Markdown Run All Restart Clear All Outputs Go To Variables Outline
Python 3.7.7

```
gb = GradientBoostingClassifier()
gb.fit(X_train, y_train)
gb_pred = gb.predict(X_test)
print("Gradient Boosting Classifier", accuracy_score(y_test, gb_pred))
```

[55]

```
Gradient Boosting Classifier 0.49516666666666664
```

```
print(classification_report(y_test, gb_pred))
```

[56]

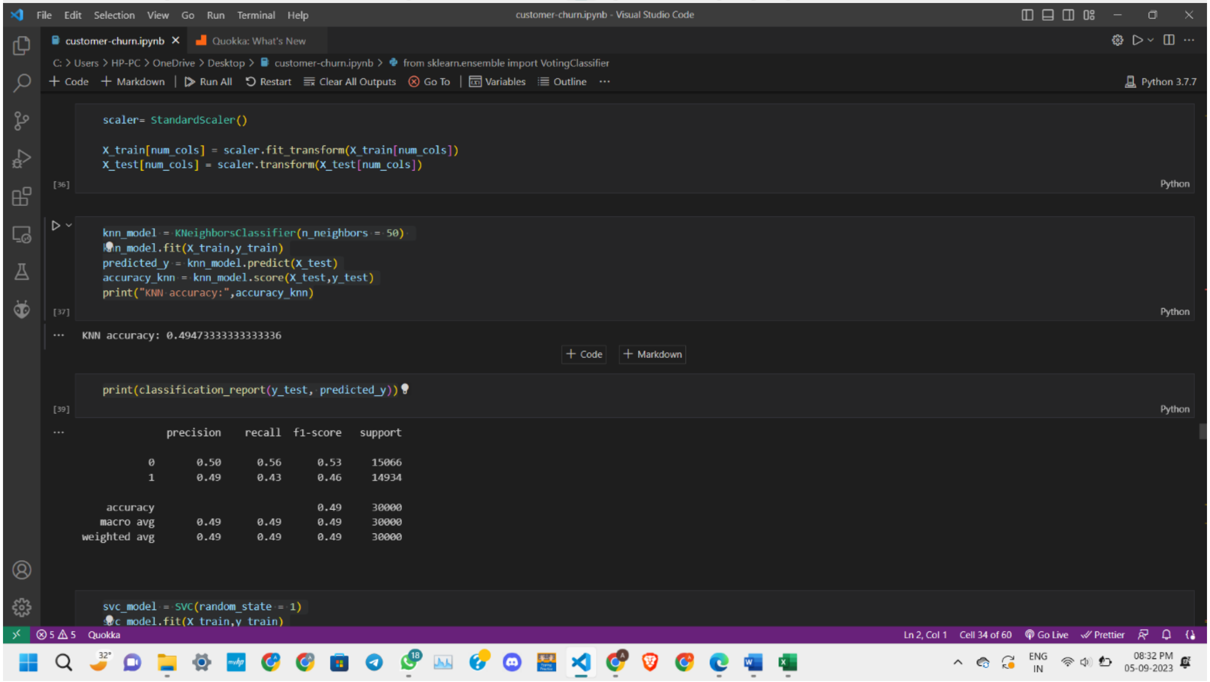
```
              precision    recall  f1-score   support

           0       0.50      0.57      0.53     15066
           1       0.49      0.42      0.46     14934

    accuracy                           0.50     30000
   macro avg       0.49      0.49      0.49     30000
weighted avg       0.49      0.50      0.49     30000
```

```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, gb_pred),
                annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title("Gradient Boosting Classifier Confusion Matrix",fontsize=14)
plt.show()
```

[57]

Gradient Boosting Classifier Confusion Matrix

Ln 1, Col 1 (220 selected) Cell 38 of 60 Go Live
ⓧ 5 ⚠ 5 Quokka
08:37 PM
05-09-2023



File Edit Selection View Go Run Terminal Help
customer-churn.ipynb - Visual Studio Code
customer-churn.ipynb
Quokka: What's New
C: > Users > HP-PC > OneDrive > Desktop > customer-churn.ipynb > from sklearn.ensemble import VotingClassifier
+ Code + Markdown Run All Restart Clear All Outputs Go To Variables Outline
Python 3.7.7

```
scaler= StandardScaler()

X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
X_test[num_cols] = scaler.transform(X_test[num_cols])
```

[36]

```
knn_model = KNeighborsClassifier(n_neighbors = 50)
knn_model.fit(X_train,y_train)
predicted_y = knn_model.predict(X_test)
accuracy_knn = knn_model.score(X_test,y_test)
print("KNN accuracy:",accuracy_knn)
```

[37]

```
KNN accuracy: 0.49473333333333336
```

```
print(classification_report(y_test, predicted_y))
```

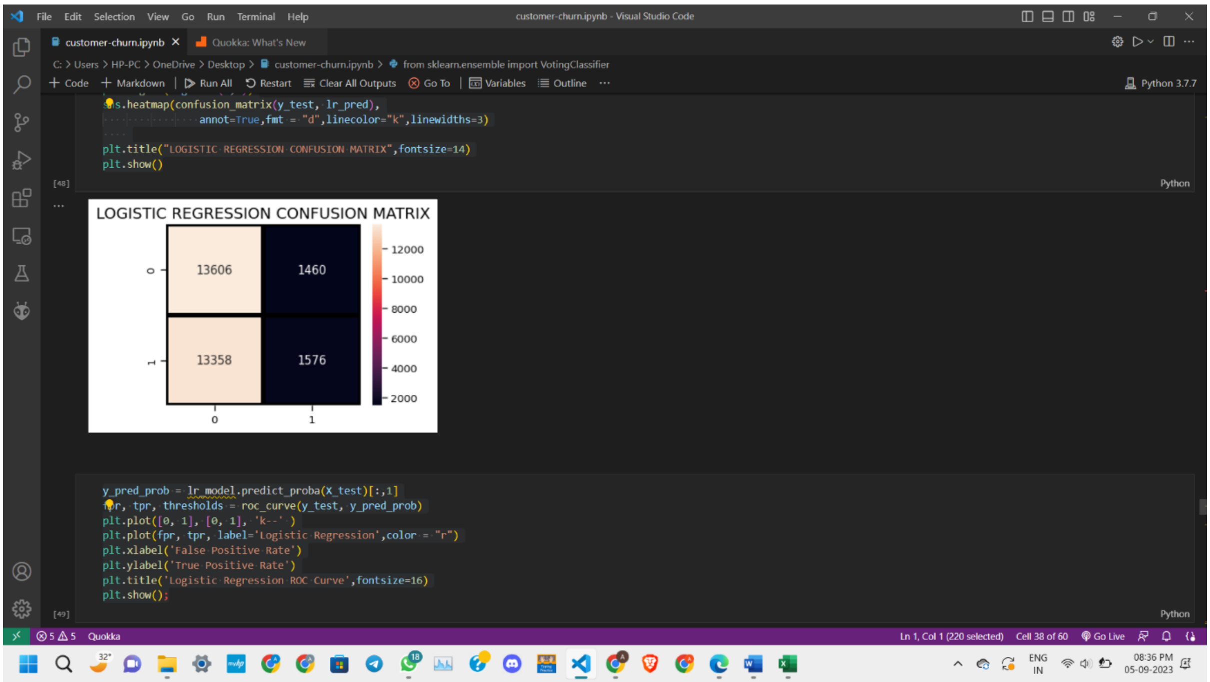
[39]

```
              precision    recall  f1-score   support

           0       0.50      0.56      0.53     15066
           1       0.49      0.43      0.46     14934

    accuracy                           0.49     30000
   macro avg       0.49      0.49      0.49     30000
weighted avg       0.49      0.49      0.49     30000
```

```
svc_model = SVC(random_state = 1)
svc_model.fit(X_train,y_train)
```

ANTIM 2021ume0204@iitjammu.ac.in **IIT JAMMU**

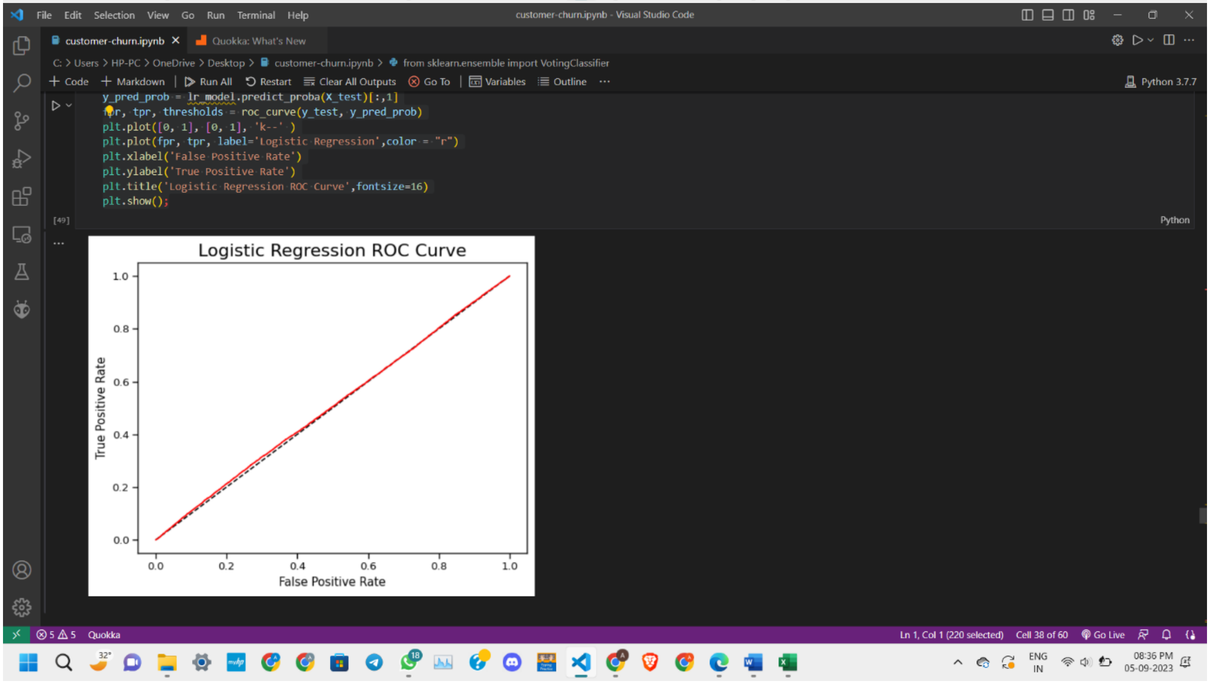
LINK FOR THE COOLAB FILE : https://colab.research.google.com/drive/1g5v8JE_pGZlTp0-zruGiPZPOoqTn93eX?usp=sharing



```
s.heatmap(confusion_matrix(y_test, lr_pred),
            annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title("LOGISTIC REGRESSION CONFUSION MATRIX",fontsize=14)
plt.show()
```

customer-churn.ipynb - Visual Studio Code
File Edit Selection View Go Run Terminal Help
customer-churn.ipynb
Quokka: What's New
C: > Users > HP-PC > OneDrive > Desktop > customer-churn.ipynb > from sklearn.ensemble import VotingClassifier
+ Code + Markdown Run All Restart Clear All Outputs Go To Variables Outline
Python 3.7.7
[48]
LOGISTIC REGRESSION CONFUSION MATRIX
0
1
13606
1460
13358
1576
12000
10000
8000
6000
4000
2000
[49]
Ln 1, Col 1 (220 selected) Cell 38 of 60 Go Live
08:36 PM
05-09-2023

```
y_pred_prob = lr_model.predict_proba(X_test)[:,1]
r, tpr, thresholds = roc_curve(y_test, y_pred_prob)
plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr, tpr, label='Logistic Regression',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Logistic Regression ROC Curve',fontsize=16)
plt.show();
```

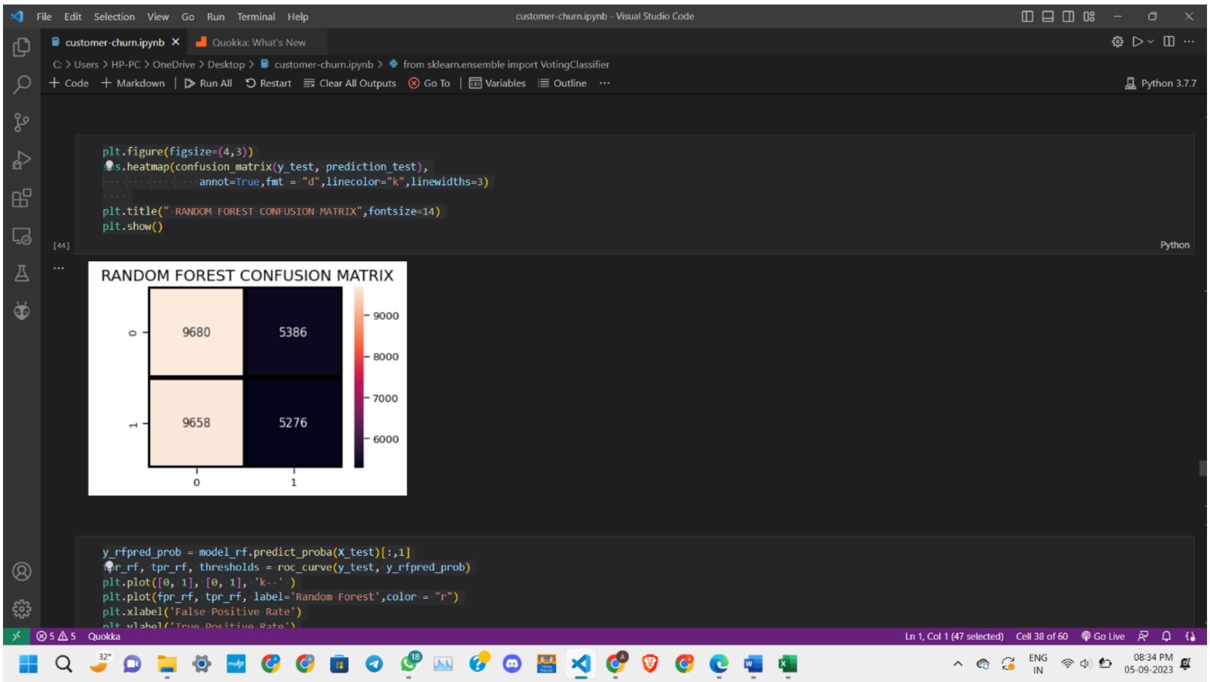


```
y_pred_prob = lr_model.predict_proba(X_test)[:,1]
r, tpr, thresholds = roc_curve(y_test, y_pred_prob)
plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr, tpr, label='Logistic Regression',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Logistic Regression ROC Curve',fontsize=16)
plt.show();
```

customer-churn.ipynb - Visual Studio Code
[49]
Logistic Regression ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Ln 1, Col 1 (220 selected) Cell 38 of 60 Go Live
08:36 PM
05-09-2023

ANTIM 2021ume0204@iitjammu.ac.in **IIT JAMMU**

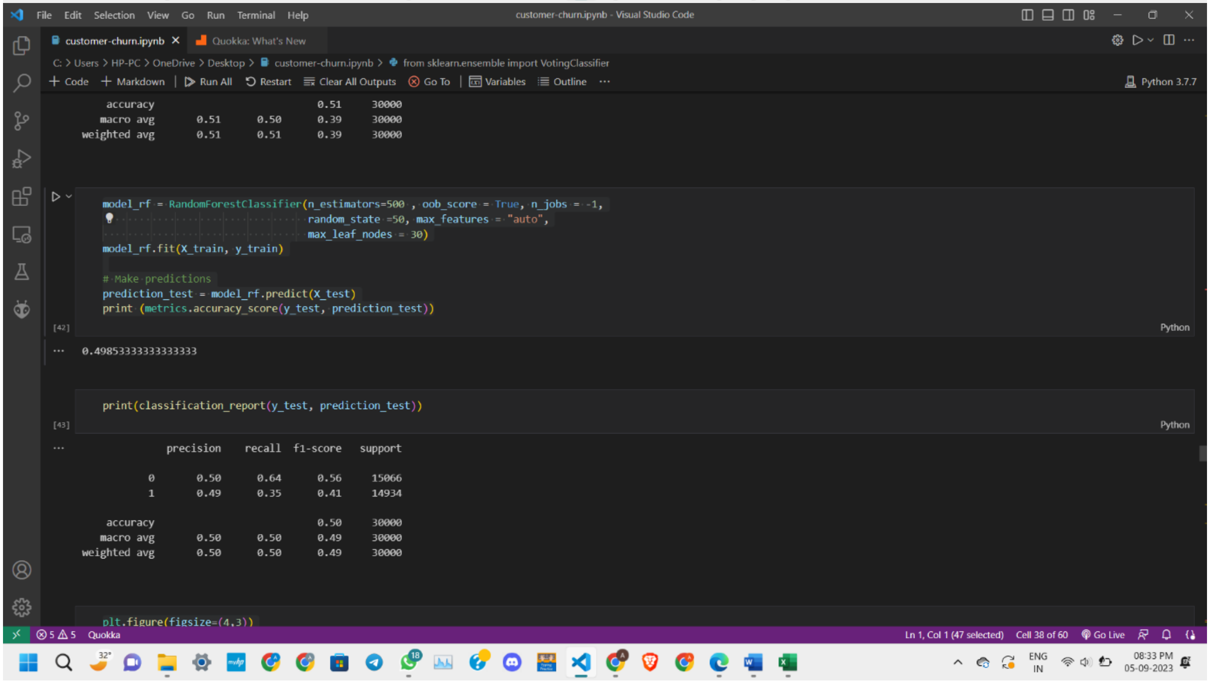
LINK FOR THE COOLAB FILE : https://colab.research.google.com/drive/1g5v8JE_pGZlTp0-zruGiPZPOoqTn93eX?usp=sharing



```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, prediction_test),
                annot=True,fmt = "d",linecolor="k",linewidths=3)
    
plt.title(" RANDOM FOREST CONFUSION MATRIX",fontsize=14)
plt.show()
```

RANDOM FOREST CONFUSION MATRIX

```
y_rfpred_prob = model_rf.predict_proba(X_test)[:,1]
fpr_rf, tpr_rf, thresholds = roc_curve(y_test, y_rfpred_prob)
plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr_rf, tpr_rf, label='Random Forest',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
```



```
    accuracy                           0.51     30000
   macro avg       0.51      0.50      0.39     30000
weighted avg       0.51      0.51      0.39     30000
```

```
model_rf = RandomForestClassifier(n_estimators=500 , oob_score = True, n_jobs = -1,
                                  random_state =50, max_features = "auto",
                                  max_leaf_nodes = 30)
model_rf.fit(X_train, y_train)

# Make predictions
prediction_test = model_rf.predict(X_test)
print (metrics.accuracy_score(y_test, prediction_test))
```

```
0.4985333333333333
```

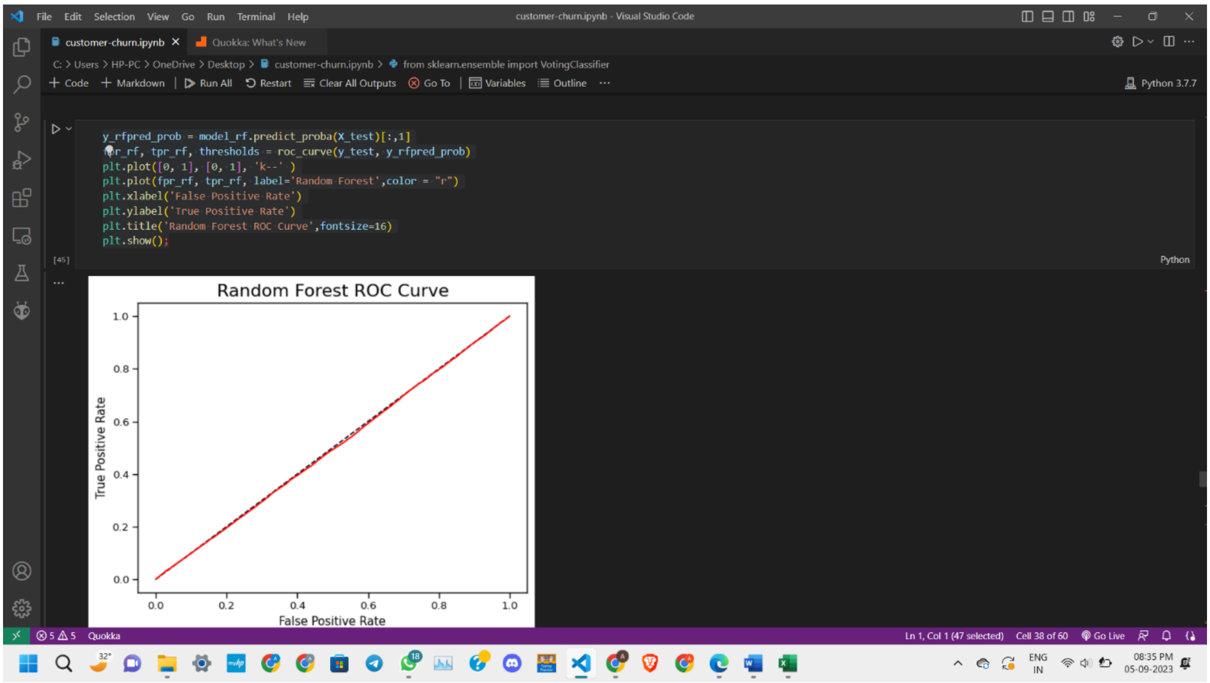
```
print(classification_report(y_test, prediction_test))
```

```
              precision    recall  f1-score   support

           0       0.50      0.64      0.56     15066
           1       0.49      0.35      0.41     14934

    accuracy                           0.50     30000
   macro avg       0.50      0.50      0.49     30000
weighted avg       0.50      0.50      0.49     30000
```

```
plt.figure(figsize=(4,3))
```

ANTIM 2021ume0204@iitjammu.ac.in IIT JAMMU

LINK FOR THE COOLAB FILE : https://colab.research.google.com/drive/1g5v8JE_pGZlTp0-zruGiPZPOoqTn93eX?usp=sharing


File Edit Selection View Go Run Terminal Help
customer-churn.ipynb - Visual Studio Code
customer-churn.ipynb
Quokka: What's New
C: > Users > HP-PC > OneDrive > Desktop > customer-churn.ipynb > from sklearn.ensemble import VotingClassifier
+ Code + Markdown | Run All Restart Clear All Outputs Go To | Variables Outline ...
Python 3.7.7
[45]
Python
Random Forest ROC Curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
5 5 Quokka
Ln 1, Col 1 (47 selected) Cell 38 of 60 Go Live
ENG IN
08:35 PM 05-09-2023

```python
y_rfpred_prob = model_rf.predict_proba(X_test)[:,1]
fpr_rf, tpr_rf, thresholds = roc_curve(y_test, y_rfpred_prob)
plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr_rf, tpr_rf, label='Random Forest',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Random Forest ROC Curve',fontsize=16)
plt.show();
```

As it is clear that all of these classifiers are giving the accuracy less than 50%.

But SVC classifier gave the accuracy of 51% so SVC is used to train the model.

ALL the Code for the Complete Model is given in Detail in the provided Google collab file along with Detailed Data visualization.

ANTIM 2021ume0204@iitjammu.ac.in **IIT JAMMU**

LINK FOR THE COOLAB FILE : https://colab.research.google.com/drive/1g5v8JE_pGZlTp0-zruGiPZPOoqTn93eX?usp=sharing