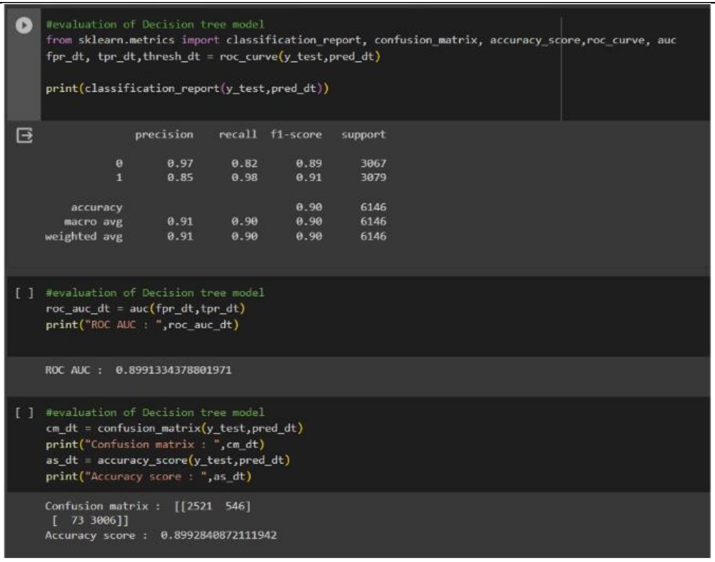
Project Development Phase
Model Performance Test

Date	9 November 2022
Team ID	Team-592447
Project Name	Airline review classificaton using machine learning
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	

```
#evaluation of Decision tree model
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,roc_curve, auc
fpr_dt, tpr_dt,thresh_dt = roc_curve(y_test,pred_dt)

print(classification_report(y_test,pred_dt))
```

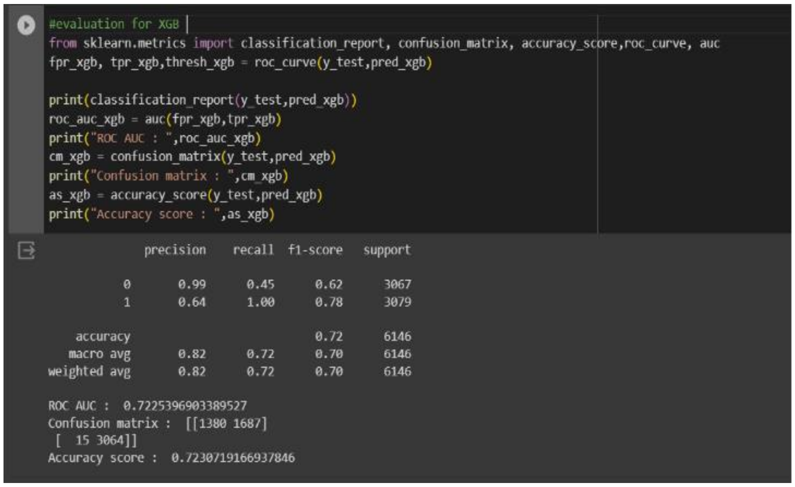
```
              precision    recall  f1-score   support

           0       0.97      0.82      0.89      3067
           1       0.85      0.98      0.91      3079

    accuracy                           0.90      6146
   macro avg       0.91      0.90      0.90      6146
weighted avg       0.91      0.90      0.90      6146
```

```
#evaluation of Decision tree model
roc_auc_dt = auc(fpr_dt,tpr_dt)
print("ROC AUC : ",roc_auc_dt)
```

```
ROC AUC :  0.8991334378801971
```

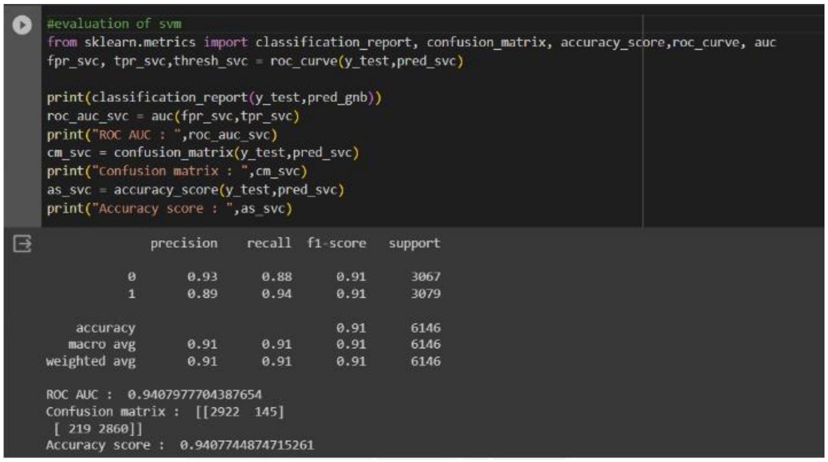
```
#evaluation of Decision tree model
cm_dt = confusion_matrix(y_test,pred_dt)
print("Confusion matrix : ",cm_dt)
as_dt = accuracy_score(y_test,pred_dt)
print("Accuracy score : ",as_dt)
```

```
Confusion matrix :  [[2521  546]
 [  73 3006]]
Accuracy score :  0.8992840872111942
```



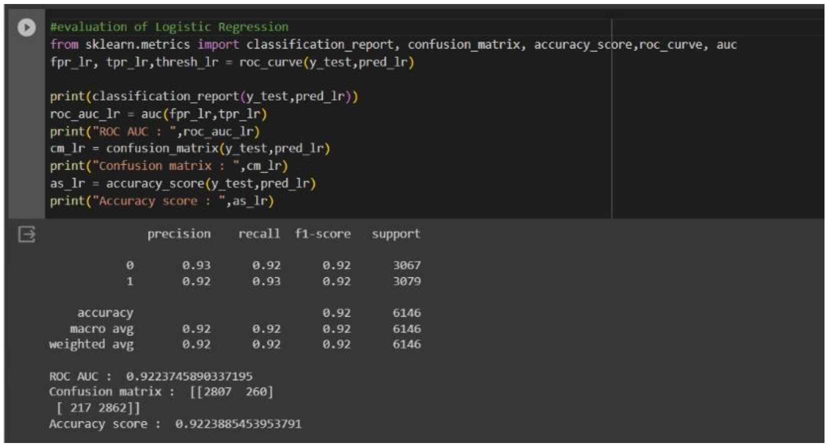
```
#evaluation for XGB
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,roc_curve, auc
fpr_xgb, tpr_xgb,thresh_xgb = roc_curve(y_test,pred_xgb)

print(classification_report(y_test,pred_xgb))
roc_auc_xgb = auc(fpr_xgb,tpr_xgb)
print("ROC AUC : ",roc_auc_xgb)
cm_xgb = confusion_matrix(y_test,pred_xgb)
print("Confusion matrix : ",cm_xgb)
as_xgb = accuracy_score(y_test,pred_xgb)
print("Accuracy score : ",as_xgb)
```

```
              precision    recall  f1-score   support

           0       0.99      0.45      0.62      3067
           1       0.64      1.00      0.78      3079

    accuracy                           0.72      6146
   macro avg       0.82      0.72      0.70      6146
weighted avg       0.82      0.72      0.70      6146

ROC AUC :  0.7225396903389527
Confusion matrix :  [[1380 1687]
 [  15 3064]]
Accuracy score :  0.7230719166937846
```



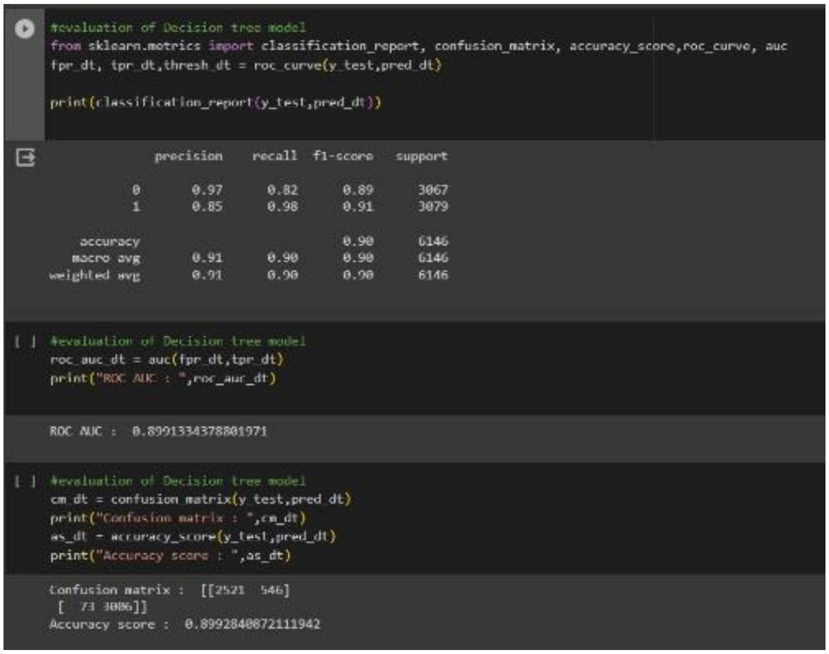
```
#evaluation of svm
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,roc_curve, auc
fpr_svc, tpr_svc,thresh_svc = roc_curve(y_test,pred_svc)

print(classification_report(y_test,pred_gnb))
roc_auc_svc = auc(fpr_svc,tpr_svc)
print("ROC AUC : ",roc_auc_svc)
cm_svc = confusion_matrix(y_test,pred_svc)
print("Confusion matrix : ",cm_svc)
as_svc = accuracy_score(y_test,pred_svc)
print("Accuracy score : ",as_svc)
```

```
              precision    recall  f1-score   support

           0       0.93      0.88      0.91      3067
           1       0.89      0.94      0.91      3079

    accuracy                           0.91      6146
   macro avg       0.91      0.91      0.91      6146
weighted avg       0.91      0.91      0.91      6146

ROC AUC :  0.9407977704387654
Confusion matrix :  [[2922  145]
 [ 219 2860]]
Accuracy score :  0.9407744874715261
```



```
#evaluation of Logistic Regression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,roc_curve, auc
fpr_lr, tpr_lr,thresh_lr = roc_curve(y_test,pred_lr)

print(classification_report(y_test,pred_lr))
roc_auc_lr = auc(fpr_lr,tpr_lr)
print("ROC AUC : ",roc_auc_lr)
cm_lr = confusion_matrix(y_test,pred_lr)
print("Confusion matrix : ",cm_lr)
as_lr = accuracy_score(y_test,pred_lr)
print("Accuracy score : ",as_lr)
```

```
              precision    recall  f1-score   support

           0       0.93      0.92      0.92      3067
           1       0.92      0.93      0.92      3079

    accuracy                           0.92      6146
   macro avg       0.92      0.92      0.92      6146
weighted avg       0.92      0.92      0.92      6146

ROC AUC :  0.9223745890337195
Confusion matrix :  [[2807  260]
 [ 217 2862]]
Accuracy score :  0.9223885453953791
```



```
#evaluation of Decision tree model
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,roc_curve, auc
fpr_dt, tpr_dt,thresh_dt = roc_curve(y_test,pred_dt)

print(classification_report(y_test,pred_dt))
```

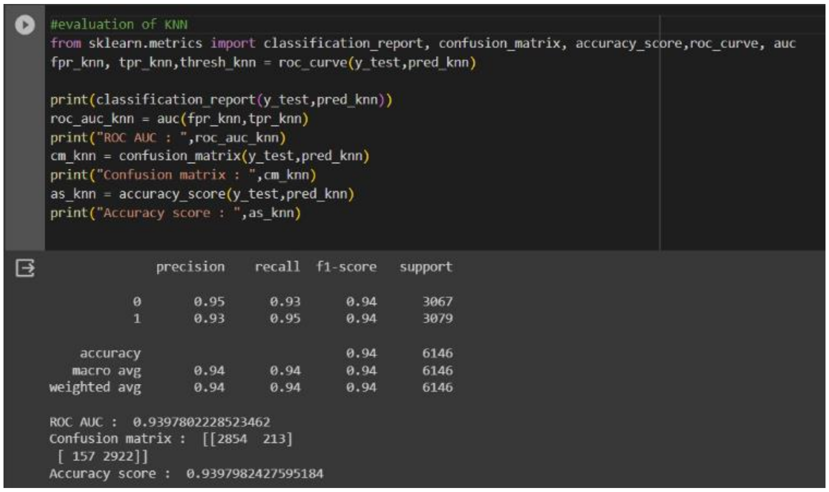
```
              precision    recall  f1-score   support

           0       0.97      0.82      0.89      3067
           1       0.85      0.98      0.91      3079

    accuracy                           0.90      6146
   macro avg       0.91      0.90      0.90      6146
weighted avg       0.91      0.90      0.90      6146
```

```
#evaluation of Decision tree model
roc_auc_dt = auc(fpr_dt,tpr_dt)
print("ROC AUC : ",roc_auc_dt)
```

```
ROC AUC :  0.8991334378801971
```

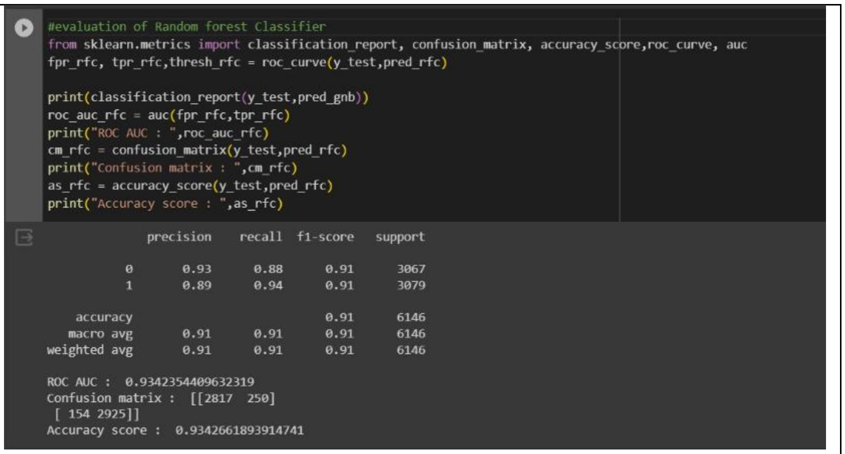
```
#evaluation of Decision tree model
cm_dt = confusion_matrix(y_test,pred_dt)
print("Confusion matrix : ",cm_dt)
as_dt = accuracy_score(y_test,pred_dt)
print("Accuracy score : ",as_dt)
```

```
Confusion matrix :  [[2521  546]
 [  73 3006]]
Accuracy score :  0.8992840872111942
```



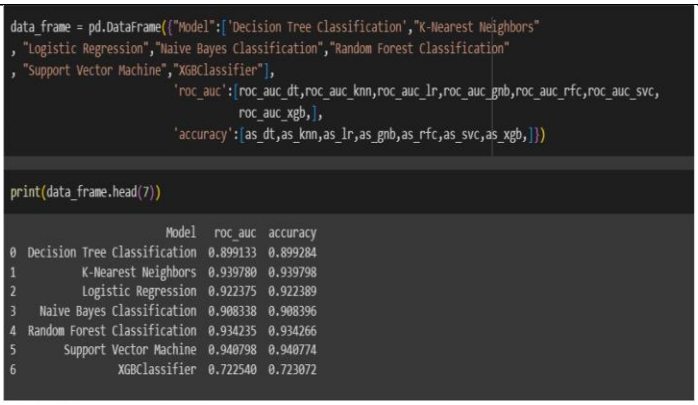
```
#evaluation of KNN
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,roc_curve, auc
fpr_knn, tpr_knn,thresh_knn = roc_curve(y_test,pred_knn)

print(classification_report(y_test,pred_knn))
roc_auc_knn = auc(fpr_knn,tpr_knn)
print("ROC AUC : ",roc_auc_knn)
cm_knn = confusion_matrix(y_test,pred_knn)
print("Confusion matrix : ",cm_knn)
as_knn = accuracy_score(y_test,pred_knn)
print("Accuracy score : ",as_knn)
```

```
              precision    recall  f1-score   support

           0       0.95      0.93      0.94      3067
           1       0.93      0.95      0.94      3079

    accuracy                           0.94      6146
   macro avg       0.94      0.94      0.94      6146
weighted avg       0.94      0.94      0.94      6146

ROC AUC :  0.9397802228523462
Confusion matrix :  [[2854  213]
 [ 157 2922]]
Accuracy score :  0.9397982427595184
```



```
#evaluation of Random forest Classifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,roc_curve, auc
fpr_rfc, tpr_rfc,thresh_rfc = roc_curve(y_test,pred_rfc)

print(classification_report(y_test,pred_gnb))
roc_auc_rfc = auc(fpr_rfc,tpr_rfc)
print("ROC AUC : ",roc_auc_rfc)
cm_rfc = confusion_matrix(y_test,pred_rfc)
print("Confusion matrix : ",cm_rfc)
as_rfc = accuracy_score(y_test,pred_rfc)
print("Accuracy score : ",as_rfc)
```

```
              precision    recall  f1-score   support

           0       0.93      0.88      0.91      3067
           1       0.89      0.94      0.91      3079

    accuracy                           0.91      6146
   macro avg       0.91      0.91      0.91      6146
weighted avg       0.91      0.91      0.91      6146

ROC AUC :  0.9342354409632319
Confusion matrix :  [[2817  250]
 [ 154 2925]]
Accuracy score :  0.9342661893914741
```

<table>
<tr>
<td>2.</td>
<td>Tune the Model</td>
<td>Hyperparameter Tuning - Validation Method -</td>
<td>

<pre>
data_frame = pd.DataFrame({"Model":['Decision Tree Classification',"K-Nearest Neighbors"
, "Logistic Regression","Naive Bayes Classification","Random Forest Classification"
, "Support Vector Machine","XGBClassifier"],
                          'roc_auc':[roc_auc_dt,roc_auc_knn,roc_auc_lr,roc_auc_gnb,roc_auc_rfc,roc_auc_svc,
                                     roc_auc_xgb,],
                          'accuracy':[as_dt,as_knn,as_lr,as_gnb,as_rfc,as_svc,as_xgb,]})

print(data_frame.head(7))

                          Model   roc_auc accuracy
0 Decision Tree Classification 0.899133 0.899284
1   K-Nearest Neighbors 0.939780 0.939798
2   Logistic Regression 0.922375 0.922389
3   Naive Bayes Classification 0.908338 0.908396
4 Random Forest Classification 0.934235 0.934266
5   Support Vector Machine 0.940798 0.940774
6   XGBClassifier 0.722540 0.723072
</pre>
</td>
</tr>
</table>