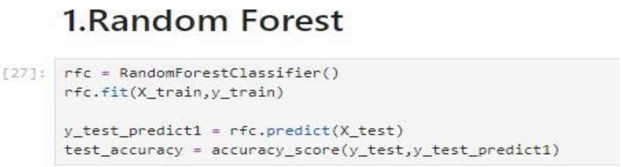
Model Development Phase Template

Date	15 July 2024
Team ID	team-740084
Project Title	Online Payments Fraud Detection
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

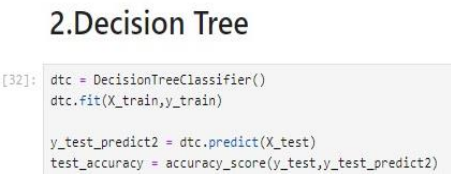
The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code (5 marks):



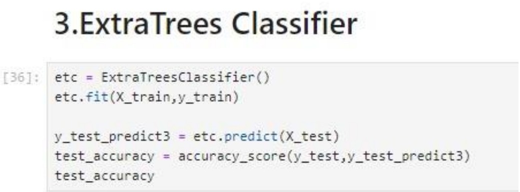
1.Random Forest

```
[27]: rfc = RandomForestClassifier()
      rfc.fit(X_train,y_train)

      y_test_predict1 = rfc.predict(X_test)
      test_accuracy = accuracy_score(y_test,y_test_predict1)
```



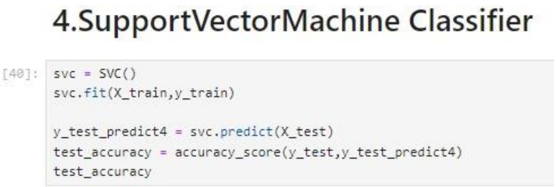
2.Decision Tree

```
[32]: dtc = DecisionTreeClassifier()
      dtc.fit(X_train,y_train)

      y_test_predict2 = dtc.predict(X_test)
      test_accuracy = accuracy_score(y_test,y_test_predict2)
```



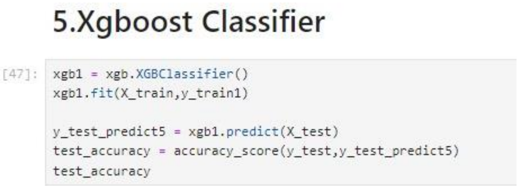
3.ExtraTrees Classifier

```
[36]: etc = ExtraTreesClassifier()
      etc.fit(X_train,y_train)

      y_test_predict3 = etc.predict(X_test)
      test_accuracy = accuracy_score(y_test,y_test_predict3)
      test_accuracy
```



4.SupportVectorMachine Classifier

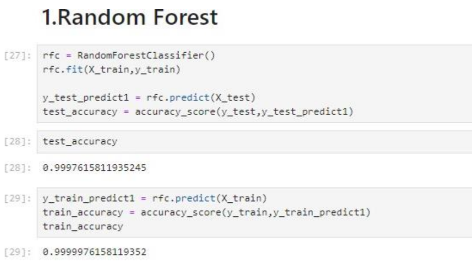
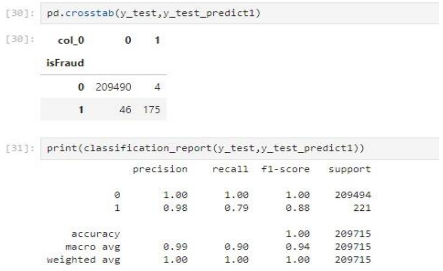
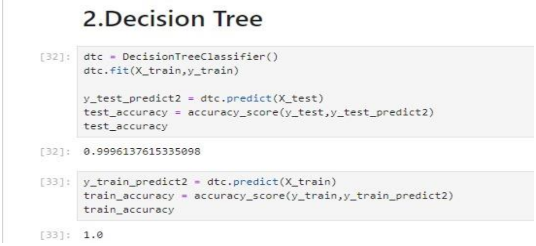
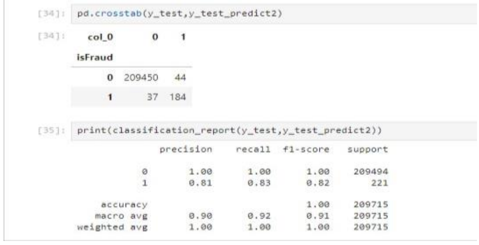
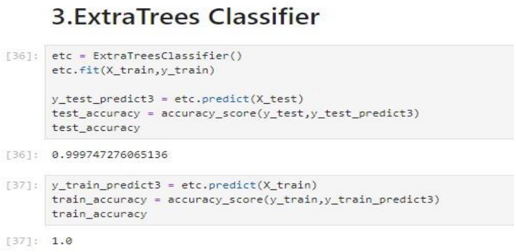
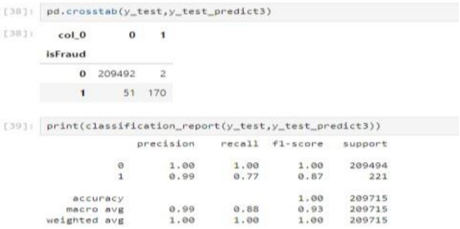
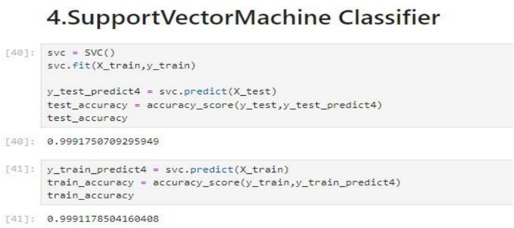
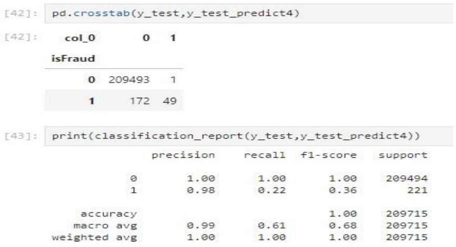
```
[40]: svc = SVC()
      svc.fit(X_train,y_train)

      y_test_predict4 = svc.predict(X_test)
      test_accuracy = accuracy_score(y_test,y_test_predict4)
      test_accuracy
```

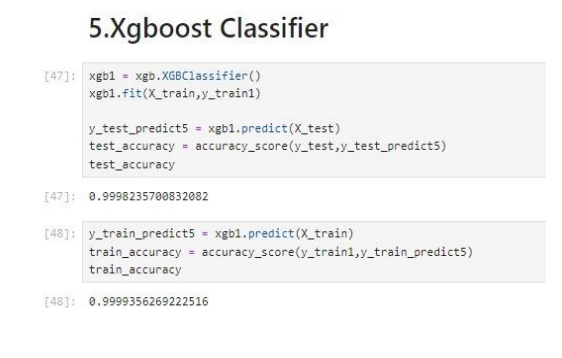
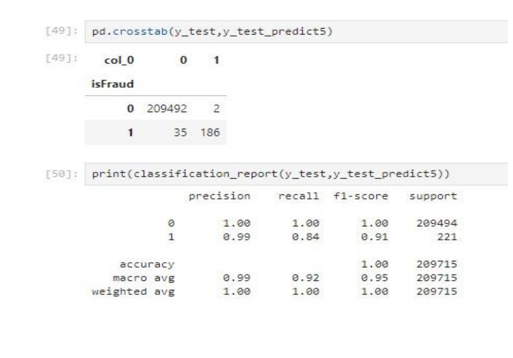


5.Xgboost Classifier

```
[47]: xgb1 = xgb.XGBClassifier()
      xgb1.fit(X_train,y_train1)

      y_test_predict5 = xgb1.predict(X_test)
      test_accuracy = accuracy_score(y_test,y_test_predict5)
      test_accuracy
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Random Forest classifier	 1.Random Forest [27]: rfc = RandomForestClassifier() rfc.fit(X_train,y_train) y_test_predict1 = rfc.predict(X_test) test_accuracy = accuracy_score(y_test,y_test_predict1) [28]: test_accuracy [28]: 0.9997615811935245 [29]: y_train_predict1 = rfc.predict(X_train) train_accuracy = accuracy_score(y_train,y_train_predict1) train_accuracy [29]: 0.9999976158119352	 [30]: pd.crosstab(y_test,y_test_predict1) [30]: col_0 0 1 isFraud 0 209490 4 1 46 175 [31]: print(classification_report(y_test,y_test_predict1)) precision recall f1-score support 0 1.00 1.00 1.00 209494 1 0.98 0.79 0.88 221 accuracy 1.00 209715 macro avg 0.99 0.90 0.94 209715 weighted avg 1.00 1.00 1.00 209715
Decision Tree classifier	 2.Decision Tree [32]: dtc = DecisionTreeClassifier() dtc.fit(X_train,y_train) y_test_predict2 = dtc.predict(X_test) test_accuracy = accuracy_score(y_test,y_test_predict2) test_accuracy [32]: 0.99961376153355098 [33]: y_train_predict2 = dtc.predict(X_train) train_accuracy = accuracy_score(y_train,y_train_predict2) train_accuracy [33]: 1.0	 [34]: pd.crosstab(y_test,y_test_predict2) [34]: col_0 0 1 isFraud 0 209450 44 1 37 184 [35]: print(classification_report(y_test,y_test_predict2)) precision recall f1-score support 0 1.00 1.00 1.00 209494 1 0.81 0.83 0.82 221 accuracy 1.00 209715 macro avg 0.90 0.92 0.91 209715 weighted avg 1.00 1.00 1.00 209715
ExtraTrees classifier	 3.ExtraTrees Classifier [36]: etc = ExtraTreesClassifier() etc.fit(X_train,y_train) y_test_predict3 = etc.predict(X_test) test_accuracy = accuracy_score(y_test,y_test_predict3) test_accuracy [36]: 0.999747276065136 [37]: y_train_predict3 = etc.predict(X_train) train_accuracy = accuracy_score(y_train,y_train_predict3) train_accuracy [37]: 1.0	 [38]: pd.crosstab(y_test,y_test_predict3) [38]: col_0 0 1 isFraud 0 209492 2 1 51 170 [39]: print(classification_report(y_test,y_test_predict3)) precision recall f1-score support 0 1.00 1.00 1.00 209494 1 0.99 0.77 0.87 221 accuracy 1.00 209715 macro avg 0.99 0.88 0.93 209715 weighted avg 1.00 1.00 1.00 209715
Support Vector Machine Classifier	 4.SupportVectorMachine Classifier [40]: svc = SVC() svc.fit(X_train,y_train) y_test_predict4 = svc.predict(X_test) test_accuracy = accuracy_score(y_test,y_test_predict4) test_accuracy [40]: 0.9991750709295949 [41]: y_train_predict4 = svc.predict(X_train) train_accuracy = accuracy_score(y_train,y_train_predict4) train_accuracy [41]: 0.9991178504160408	 [42]: pd.crosstab(y_test,y_test_predict4) [42]: col_0 0 1 isFraud 0 209493 1 1 172 49 [43]: print(classification_report(y_test,y_test_predict4)) precision recall f1-score support 0 1.00 1.00 1.00 209494 1 0.98 0.22 0.36 221 accuracy 1.00 209715 macro avg 0.99 0.61 0.68 209715 weighted avg 1.00 1.00 1.00 209715

Xgboost Classifier		

5.Xgboost Classifier

```
[47]: xgb1 = xgb.XGBClassifier()
      xgb1.fit(X_train,y_train1)

      y_test_predict5 = xgb1.predict(X_test)
      test_accuracy = accuracy_score(y_test,y_test_predict5)
      test_accuracy
```

```
[47]: 0.9998235700832082
```

```
[48]: y_train_predict5 = xgb1.predict(X_train)
      train_accuracy = accuracy_score(y_train1,y_train_predict5)
      train_accuracy
```

```
[48]: 0.9999356269222516
```

```
[49]: pd.crosstab(y_test,y_test_predict5)
```

[49]:

col_0	0	1
isFraud		
0	209492	2
1	35	186

```
[50]: print(classification_report(y_test,y_test_predict5))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    209494
           1       0.99      0.84      0.91       221

    accuracy                           1.00    209715
   macro avg       0.99      0.92      0.95    209715
weighted avg       1.00      1.00      1.00    209715
```