Model Development Phase Template

Date	10 June 2014
Team ID	team-739671
Project Title	Ecommerce Shipping Prediction using Machine Learning
Maximum Marks	10 Marks

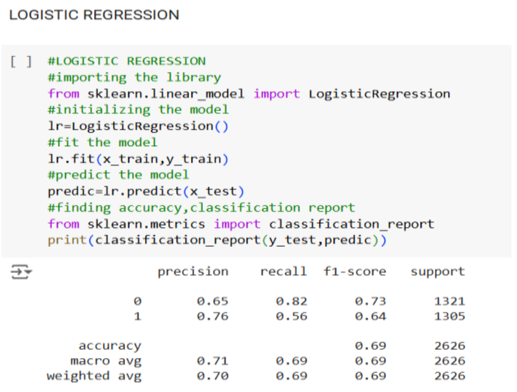
Initial Model Training Code, Model Validation and Evaluation Report

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):

Paste the screenshot of the model training code

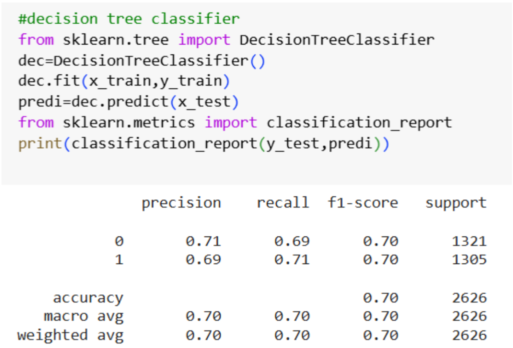
Model Validation and Evaluation Report (5 marks):

<table>
<tr><th>Model</th><th>Summary</th><th>Training and Validation Performance Metrics</th></tr>
<tr><td>Model 1</td><td>Logistic regression model typically include accuracy, precision, recall, F1 score to evaluate its predictive performance and generalization capability.</td><td>


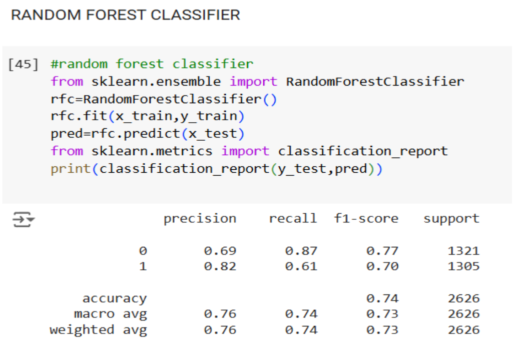
LOGISTIC REGRESSION

```
[ ] #LOGISTIC REGRESSION
    #importing the library
    from sklearn.linear_model import LogisticRegression
    #initializing the model
    lr=LogisticRegression()
    #fit the model
    lr.fit(x_train,y_train)
    #predict the model
    predic=lr.predict(x_test)
    #finding accuracy,classification report
    from sklearn.metrics import classification_report
    print(classification_report(y_test,predic))
```

```
              precision    recall  f1-score   support

           0       0.65      0.82      0.73      1321
           1       0.76      0.56      0.64      1305

    accuracy                           0.69      2626
   macro avg       0.71      0.69      0.69      2626
weighted avg       0.70      0.69      0.69      2626
```
</td></tr>
</table>

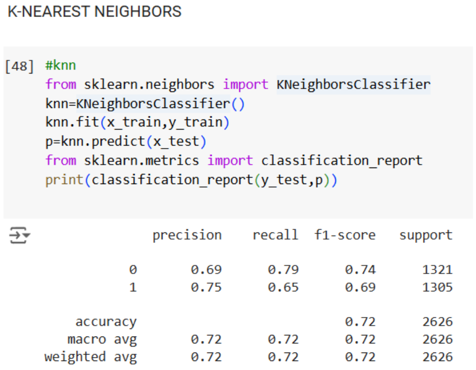
<table>
<tr>
<td>Model 2</td>
<td>Decision tree classifier model commonly include accuracy, precision, recall, F1 score which help assess the model's prediction accuracy and generalizability</td>
<td>


```
#decision tree classifier
from sklearn.tree import DecisionTreeClassifier
dec=DecisionTreeClassifier()
dec.fit(x_train,y_train)
predi=dec.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test,predi))
```

```
              precision    recall  f1-score   support

           0       0.71      0.69      0.70      1321
           1       0.69      0.71      0.70      1305

    accuracy                           0.70      2626
   macro avg       0.70      0.70      0.70      2626
weighted avg       0.70      0.70      0.70      2626
```
</td>
</tr>
<tr>
<td>Model 3</td>
<td>Random forest classifier model often encompass accuracy, precision, recall, F1 score to measure its prediction quality and robustness.</td>
<td>


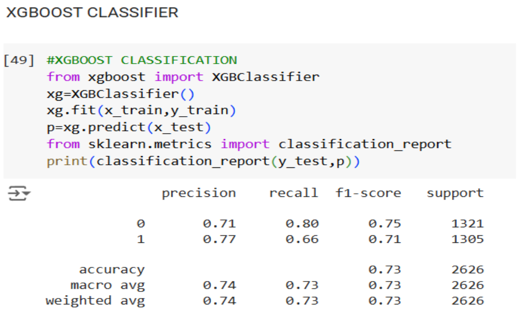
RANDOM FOREST CLASSIFIER

```
[45] #random forest classifier
     from sklearn.ensemble import RandomForestClassifier
     rfc=RandomForestClassifier()
     rfc.fit(x_train,y_train)
     pred=rfc.predict(x_test)
     from sklearn.metrics import classification_report
     print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.69      0.87      0.77      1321
           1       0.82      0.61      0.70      1305

    accuracy                           0.74      2626
   macro avg       0.76      0.74      0.73      2626
weighted avg       0.76      0.74      0.73      2626
```
</td>
</tr>
<tr>
<td>Model 4</td>
<td>K-nearest neighbors classifier model typically include accuracy, precision, recall, F1 score to evaluate its prediction performance and generalization ability</td>
<td>


K-NEAREST NEIGHBORS

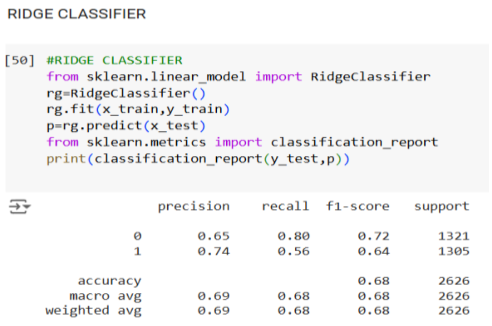
```
[48] #knn
     from sklearn.neighbors import KNeighborsClassifier
     knn=KNeighborsClassifier()
     knn.fit(x_train,y_train)
     p=knn.predict(x_test)
     from sklearn.metrics import classification_report
     print(classification_report(y_test,p))
```

```
              precision    recall  f1-score   support

           0       0.69      0.79      0.74      1321
           1       0.75      0.65      0.69      1305

    accuracy                           0.72      2626
   macro avg       0.72      0.72      0.72      2626
weighted avg       0.72      0.72      0.72      2626
```
</td>
</tr>
<tr>
<td>Model 5</td>
<td>XGBoost classifier model typically include accuracy, precision, recall, F1 score used to evaluate the model's predictive performance and ability to generalize</td>
<td>


XGBOOST CLASSIFIER

```
[49] #XGBOOST CLASSIFICATION
     from xgboost import XGBClassifier
     xg=XGBClassifier()
     xg.fit(x_train,y_train)
     p=xg.predict(x_test)
     from sklearn.metrics import classification_report
     print(classification_report(y_test,p))
```

```
              precision    recall  f1-score   support

           0       0.71      0.80      0.75      1321
           1       0.77      0.66      0.71      1305

    accuracy                           0.73      2626
   macro avg       0.74      0.73      0.73      2626
weighted avg       0.74      0.73      0.73      2626
```
</td>
</tr>
</table>

Model 6	Ridge classifier model typically include accuracy, precision, recall, F1 score, and mean squared error to evaluate its prediction performance and generalization.	 RIDGE CLASSIFIER

```
[50] #RIDGE CLASSIFIER
     from sklearn.linear_model import RidgeClassifier
     rg=RidgeClassifier()
     rg.fit(x_train,y_train)
     p=rg.predict(x_test)
     from sklearn.metrics import classification_report
     print(classification_report(y_test,p))
```

```
              precision    recall  f1-score   support

           0       0.65      0.80      0.72      1321
           1       0.74      0.56      0.64      1305

    accuracy                           0.68      2626
   macro avg       0.69      0.68      0.68      2626
weighted avg       0.69      0.68      0.68      2626
```