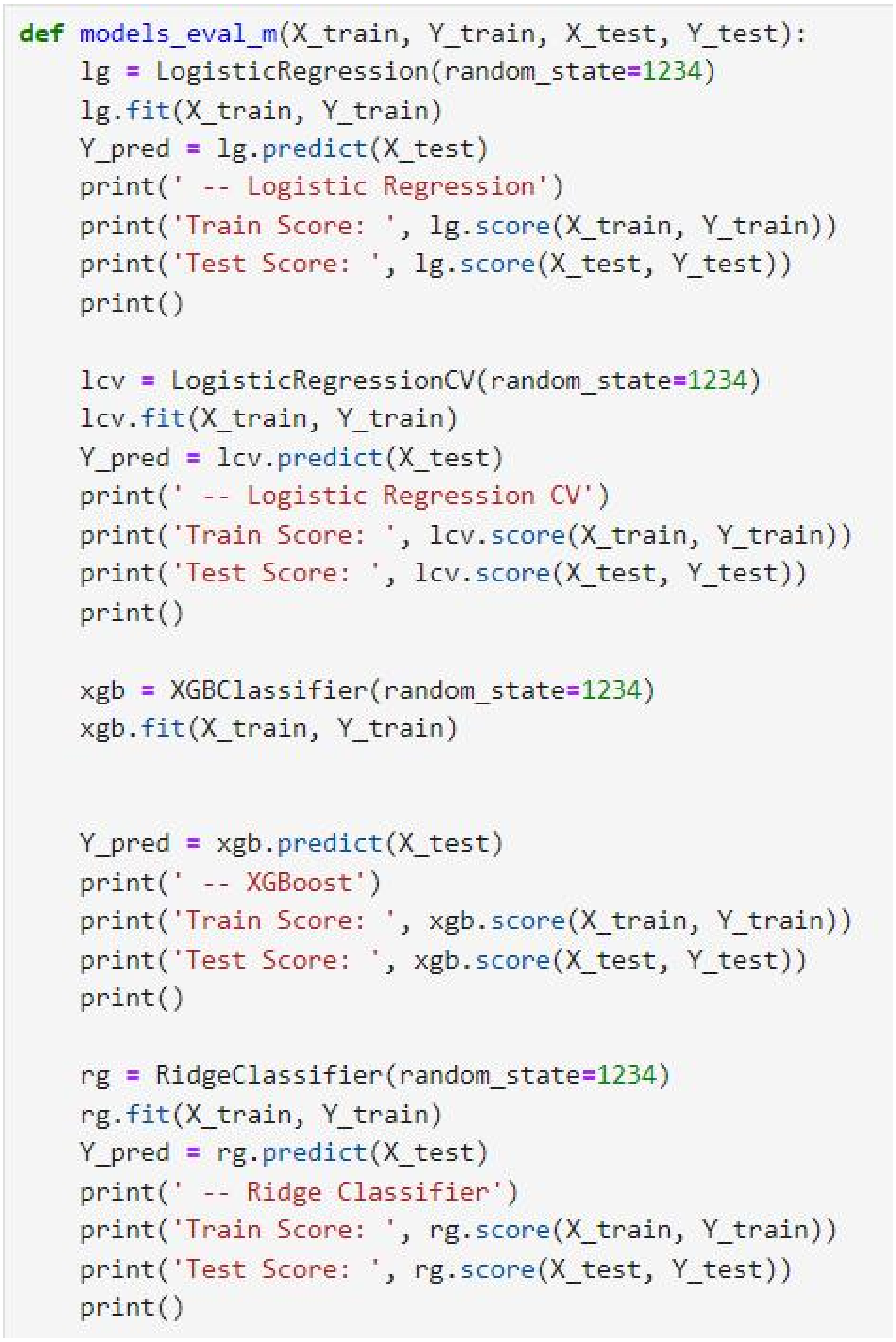
Model Development Phase Template

Date	28 June 2024
Team ID	SWTID1720196555
Project Title	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	4 Marks

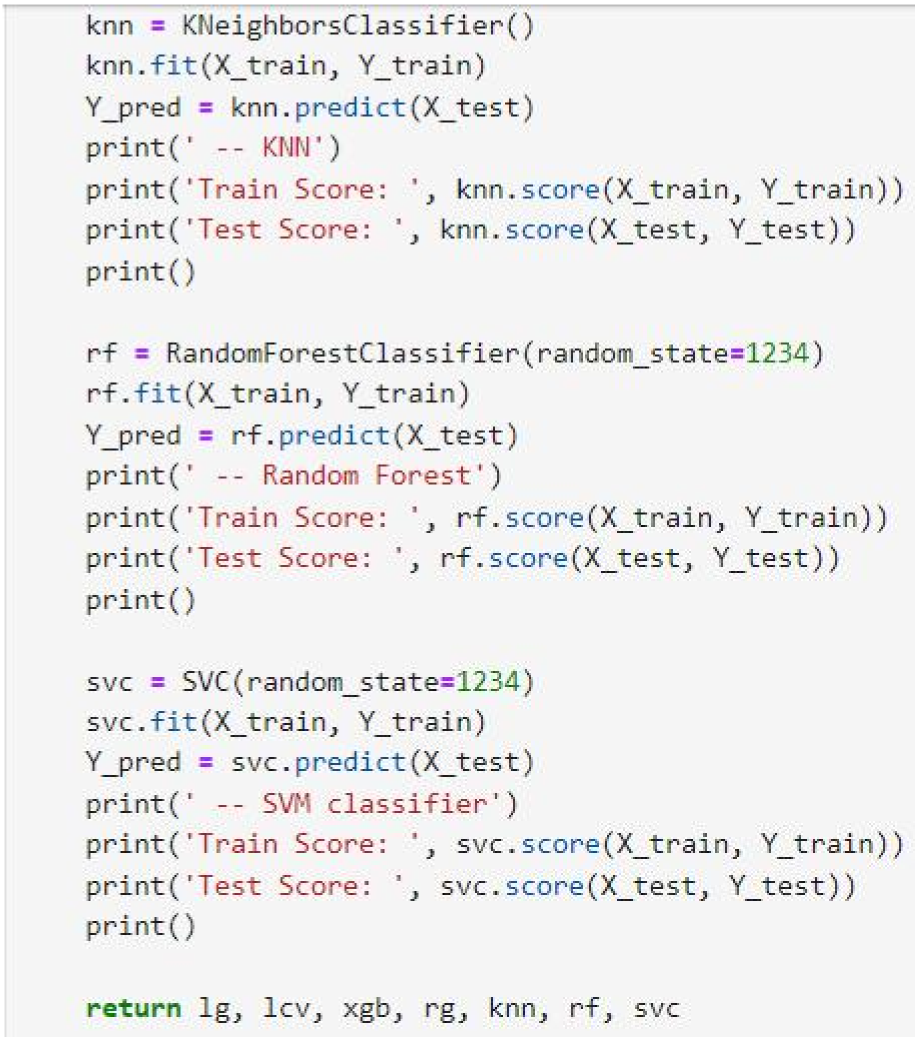
Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

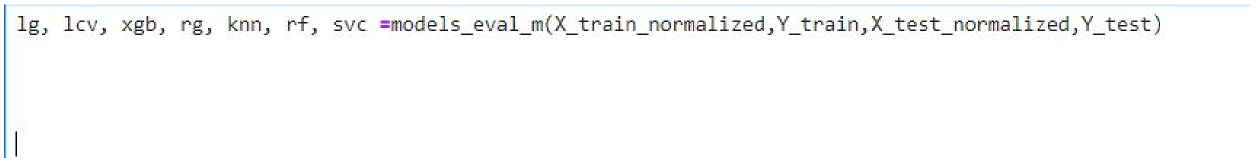
Initial Model Training Code:



```python
def models_eval_m(X_train, Y_train, X_test, Y_test):
    lg = LogisticRegression(random_state=1234)
    lg.fit(X_train, Y_train)
    Y_pred = lg.predict(X_test)
    print(' -- Logistic Regression')
    print('Train Score: ', lg.score(X_train, Y_train))
    print('Test Score: ', lg.score(X_test, Y_test))
    print()

    lcv = LogisticRegressionCV(random_state=1234)
    lcv.fit(X_train, Y_train)
    Y_pred = lcv.predict(X_test)
    print(' -- Logistic Regression CV')
    print('Train Score: ', lcv.score(X_train, Y_train))
    print('Test Score: ', lcv.score(X_test, Y_test))
    print()

    xgb = XGBClassifier(random_state=1234)
    xgb.fit(X_train, Y_train)


    Y_pred = xgb.predict(X_test)
    print(' -- XGBoost')
    print('Train Score: ', xgb.score(X_train, Y_train))
    print('Test Score: ', xgb.score(X_test, Y_test))
    print()

    rg = RidgeClassifier(random_state=1234)
    rg.fit(X_train, Y_train)
    Y_pred = rg.predict(X_test)
    print(' -- Ridge Classifier')
    print('Train Score: ', rg.score(X_train, Y_train))
    print('Test Score: ', rg.score(X_test, Y_test))
    print()
```

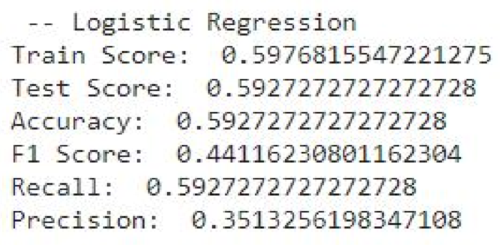
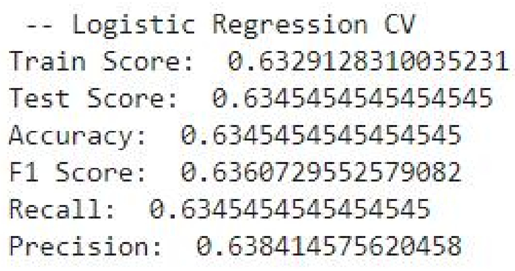
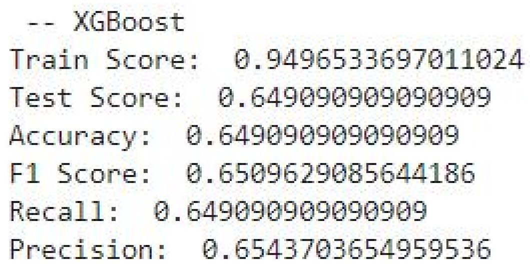
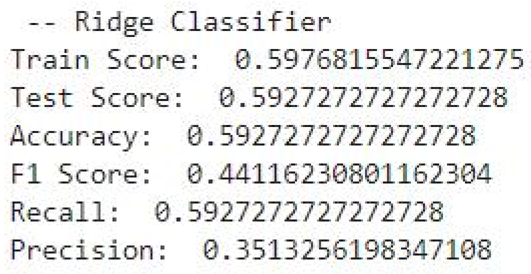


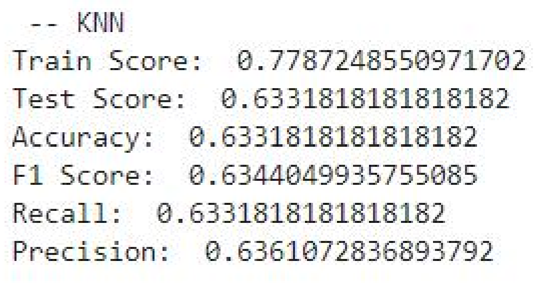
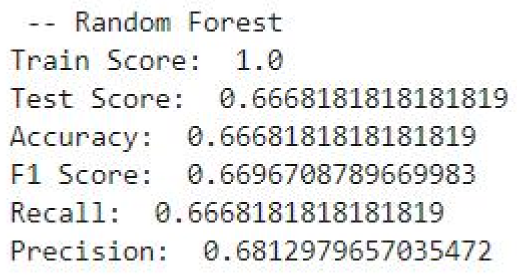
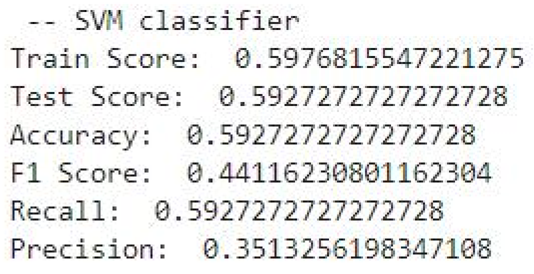
```
    knn = KNeighborsClassifier()
    knn.fit(X_train, Y_train)
    Y_pred = knn.predict(X_test)
    print(' -- KNN')
    print('Train Score: ', knn.score(X_train, Y_train))
    print('Test Score: ', knn.score(X_test, Y_test))
    print()

    rf = RandomForestClassifier(random_state=1234)
    rf.fit(X_train, Y_train)
    Y_pred = rf.predict(X_test)
    print(' -- Random Forest')
    print('Train Score: ', rf.score(X_train, Y_train))
    print('Test Score: ', rf.score(X_test, Y_test))
    print()

    svc = SVC(random_state=1234)
    svc.fit(X_train, Y_train)
    Y_pred = svc.predict(X_test)
    print(' -- SVM classifier')
    print('Train Score: ', svc.score(X_train, Y_train))
    print('Test Score: ', svc.score(X_test, Y_test))
    print()

    return lg, lcv, xgb, rg, knn, rf, svc
```



```
lg, lcv, xgb, rg, knn, rf, svc =models_eval_m(X_train_normalized,Y_train,X_test_normalized,Y_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy
Logistic Regression	 -- Logistic Regression Train Score: 0.5976815547221275 Test Score: 0.5927272727272728 Accuracy: 0.5927272727272728 F1 Score: 0.44116230801162304 Recall: 0.5927272727272728 Precision: 0.3513256198347108	59%
Logistic Regression CV	 -- Logistic Regression CV Train Score: 0.6329128310035231 Test Score: 0.6345454545454545 Accuracy: 0.6345454545454545 F1 Score: 0.6360729552579082 Recall: 0.6345454545454545 Precision: 0.638414575620458	63.45%
XGBoost	 -- XGBoost Train Score: 0.9496533697011024 Test Score: 0.6490909090909090 Accuracy: 0.6490909090909090 F1 Score: 0.6509629085644186 Recall: 0.6490909090909090 Precision: 0.6543703654959536	64.9%
Ridge Classifier	 -- Ridge Classifier Train Score: 0.5976815547221275 Test Score: 0.5927272727272728 Accuracy: 0.5927272727272728 F1 Score: 0.44116230801162304 Recall: 0.5927272727272728 Precision: 0.3513256198347108	59.27%

KNN	 -- KNN Train Score: 0.7787248550971702 Test Score: 0.6331818181818182 Accuracy: 0.6331818181818182 F1 Score: 0.6344049935755085 Recall: 0.6331818181818182 Precision: 0.6361072836893792	63.31%
Random Forest	 -- Random Forest Train Score: 1.0 Test Score: 0.6668181818181819 Accuracy: 0.6668181818181819 F1 Score: 0.6696708789669983 Recall: 0.6668181818181819 Precision: 0.6812979657035472	66.6%
SVM classifier	 -- SVM classifier Train Score: 0.5976815547221275 Test Score: 0.5927272727272728 Accuracy: 0.5927272727272728 F1 Score: 0.44116230801162304 Recall: 0.5927272727272728 Precision: 0.3513256198347108	59.27%