Model Development Phase Template

Date	12 March 2024
Team ID	SWTID1720089323
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:

Paste the screenshot of the model training code

```
#training models witchout without any hyperparameters
def models_eval_mm(x_train,y_train,x_test,y_test):

    #Logistic Regression
    lg = LogisticRegression()
    lg.fit(x_train,y_train)

    #Logistic Regression CV
    lcv = LogisticRegressionCV()
    lcv.fit(x_train,y_train)

    #XGBoost
    xgb = XGBClassifier()
    xgb.fit(x_train,y_train)

    #Ridge Classifier
    rg = RidgeClassifier()
    rg.fit(x_train,y_train)

    #KNN
    knn = KNeighborsClassifier()
    knn.fit(x_train,y_train)

    #Random Forest
    rf = RandomForestClassifier()
    rf.fit(x_train,y_train)

    #SVM classifier
    svc = svm.SVC()
    svc.fit(x_train,y_train)

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

lg,lcv,xgb,rg,knn,rf,svc = models_eval_mm(x_train,y_train,x_test,y_test)
```

```
model_list = {
    'logistic regression':lg,
    'logistic regression CV':lcv,
    'XGBoost':xgb,
    'Ridge classifier':rg,
    'KNN':knn,
    'Random Forest':rf,
    'Support Vector Classifier':svc
}
```

```
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_score, confusion_matrix
def eval(name,model):
    y_pred = model.predict(x_test)
    y_train_pred = model.predict(x_train)
    print("Model : ",name)
    print("For Training data : -")
    print("Accuracy : {:.2f}".format(accuracy_score(y_train, y_train_pred) * 100))
    print("f1 score : {:.2f}".format(f1_score(y_train, y_train_pred) * 100))
    print("Recall score : {:.2f}".format(recall_score(y_train, y_train_pred) * 100))
    print("Precision score : {:.2f}".format(precision_score(y_train, y_train_pred) * 100))
    print("\nFor Test data : -")
    print("Accuracy : {:.2f}".format(accuracy_score(y_test, y_pred) * 100))
    print("f1 score : {:.2f}".format(f1_score(y_test, y_pred) * 100))
    print("Recall score : {:.2f}".format(recall_score(y_test, y_pred) * 100))
    print("Precision score : {:.2f}".format(precision_score(y_test, y_pred) * 100))
    print("------------------------------")
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Logistic Regression	print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.53 0.66 0.59 1312 1 0.73 0.61 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.64 0.63 3300 weighted avg 0.65 0.63 0.63 3300	62.94	print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[870 442] [781 1207]]
Logistic Regression CV	print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.52 0.67 0.59 1312 1 0.73 0.59 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.63 0.62 3300 weighted avg 0.65 0.63 0.63 3300	62.61	print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[884 428] [806 1182]]
XGBoost Classifier	print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.56 0.70 0.62 1312 1 0.76 0.64 0.70 1988 accuracy 0.66 3300 macro avg 0.66 0.67 0.66 3300 weighted avg 0.68 0.66 0.67 3300	66.24	print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[916 396] [718 1270]]

Ridge Classifier	`print(classification_report(y_test, y_pred))` Classification Report: precision recall f1-score support 0 0.53 0.67 0.59 1312 1 0.73 0.60 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.63 0.62 3300 weighted avg 0.65 0.63 0.63 3300	62.82	`print(confusion_matrix(y_test, y_pred))` Confusion Matrix: [[874 438] [789 1199]]
K-Nearest Neighbors Classifier	`print(classification_report(y_test, y_pred))` Classification Report: precision recall f1-score support 0 0.53 0.69 0.60 1312 1 0.74 0.59 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.64 0.63 3300 weighted avg 0.66 0.63 0.63 3300	63.06	`print(confusion_matrix(y_test, y_pred)` Confusion Matrix: [[905 407] [812 1176]]
Random Forest Classifier	`print(classification_report(y_test, y_pred))` Classification Report: precision recall f1-score support 0 0.56 0.76 0.65 1312 1 0.79 0.61 0.69 1988 accuracy 0.67 3300 macro avg 0.68 0.68 0.67 3300 weighted avg 0.70 0.67 0.67 3300	66.58	`print(confusion_matrix(y_test, y_pred))` Confusion Matrix: [[997 315] [776 1212]]
Support Vector Machine Classifier	`print(classification_report(y_test, y_pred))` Classification Report: precision recall f1-score support 0 0.54 0.87 0.66 1312 1 0.85 0.51 0.64 1988 accuracy 0.65 3300 macro avg 0.70 0.69 0.65 3300 weighted avg 0.73 0.65 0.65 3300	65.15	`print(confusion_matrix(y_test, y_pred))` Confusion Matrix: [[1139 173] [977 1011]]