Model Development Phase Template

Date	18 July 2024
Team ID	SWTID1720086535
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:

In [22]:
```
from sklearn import svm
from sklearn.linear_model import LogisticRegression, LogisticRegressionCV, RidgeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from xgboost import XGBClassifier
from sklearn.preprocessing import Normalizer
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_score, confusion_matrix

def model_evaluation(x_train,y_train,x_test,y_test):
    lr=LogisticRegression(random_state=1234)
    lr.fit(x_train,y_train)
    print('LOGISTIC REGRESSION')
    print('Train Score:',lr.score(x_train,y_train))
    print('Test Score:',lr.score(x_test,y_test))
    print()

    lcv=LogisticRegressionCV(random_state=1234)
    lcv.fit(x_train,y_train)
    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)
    print('XGBOOST')
    print('Train Score:',xgb.score(x_train,y_train))
    print('Test Score:',xgb.score(x_test,y_test))
    print()

    rc=RidgeClassifier(random_state=1234)
    rc.fit(x_train,y_train)
    print('RIDGE CLASSIFIER')
    print('Train Score:',rc.score(x_train,y_train))
    print('Test Score:',rc.score(x_test,y_test))
    print()

    kn=KNeighborsClassifier()
    kn.fit(x_train,y_train)
    print('K NEIGHBORS CLASSIFIER')
    print('Train Score:',kn.score(x_train,y_train))
    print('Test Score:',kn.score(x_test,y_test))
    print()

    rf=RandomForestClassifier(random_state=1234)
    rf.fit(x_train,y_train)
    print('RANDOM FOREST CLASSIFIER')
    print('Train Score:',rf.score(x_train,y_train))
    print('Test Score:',rf.score(x_test,y_test))
    print()

    svc=svm.SVC(random_state=1234)
    svc.fit(x_train,y_train)
    print('SVM CLASSIFIER')
    print('Train Score:',svc.score(x_train,y_train))
    print('Test Score:',svc.score(x_test,y_test))
    print()

    return lr,lcv,xgb,rc,kn,rf,svc
```

In [23]:
```
lr,lcv,xgb,rc,kn,rf,svc = model_evaluation(xnorm_train,y_train,xnorm_test,y_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
logistic regression	`print(classification_report(y_test,y_pred))` precision recall f1-score support 0 0.56 0.56 0.56 896 1 0.70 0.69 0.70 1304 accuracy 0.64 2200 macro avg 0.63 0.63 0.63 2200 weighted avg 0.64 0.64 0.64 2200	64%	`print(confusion_matrix(y_test,y_pred))` [[503 393] [398 906]]
logistic regression CV	`print(classification_report(y_test,y_pred))` precision recall f1-score support 0 0.56 0.52 0.54 896 1 0.69 0.72 0.70 1304 accuracy 0.64 2200 macro avg 0.62 0.62 0.62 2200 weighted avg 0.63 0.64 0.64 2200	64%	`print(confusion_matrix(y_test,y_pred))` [[463 433] [362 942]]
XGBoost	`print(classification_report(y_test,y_pred))` precision recall f1-score support 0 0.57 0.64 0.60 896 1 0.73 0.67 0.70 1304 accuracy 0.66 2200 macro avg 0.65 0.65 0.65 2200 weighted avg 0.66 0.66 0.66 2200	66%	`print(confusion_matrix(y_test,y_pred))` [[573 323] [436 868]]
ridge classifier	`print(classification_report(y_test,y_pred))` precision recall f1-score support 0 0.56 0.66 0.61 896 1 0.74 0.65 0.69 1304 accuracy 0.65 2200 macro avg 0.65 0.65 0.65 2200 weighted avg 0.66 0.65 0.66 2200	65%	`print(confusion_matrix(y_test,y_pred))` [[593 303] [462 842]]

K nearest neighbors	`print(classification_report(y_test,y_pred))` precision recall f1-score support 0 0.55 0.57 0.56 896 1 0.70 0.68 0.69 1304 accuracy 0.63 2200 macro avg 0.62 0.62 0.62 2200 weighted avg 0.64 0.63 0.64 2200	63%	`print(confusion_matrix(y_test,y_pred))` [[511 385] [420 884]]
random forest	`print(classification_report(y_test,y_pred))` precision recall f1-score support 0 0.56 0.66 0.61 896 1 0.74 0.65 0.69 1304 accuracy 0.65 2200 macro avg 0.65 0.65 0.65 2200 weighted avg 0.66 0.65 0.66 2200	66%	`print(confusion_matrix(y_test,y_pred))` [[593 303] [462 842]]
support vector classifier	`print(classification_report(y_test,y_pred))` precision recall f1-score support 0 0.56 0.82 0.66 896 1 0.82 0.56 0.66 1304 accuracy 0.66 2200 macro avg 0.69 0.69 0.66 2200 weighted avg 0.71 0.66 0.66 2200	66%	`print(confusion_matrix(y_test,y_pred))` [[734 162] [578 726]]