

Model Development Phase Template

Date	14 July 2024
Team ID	SWTID1720151584
Project Title	E-Commerce 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:

Support Vector Machine:

```
svm_model = svm.SVC(gamma='auto',C=5,kernel='rbf')
svm_model.fit(X_train,y_train)
y_pred = svm_model.predict(X_test)
print(classification_report(y_test,y_pred))
```

Logistic Regression:

```
lr=LogisticRegression()
lr.fit(X_train,y_train)
predLR=lr.predict(X_test)
print(classification_report(y_test,predLR))
```

Decision Tree Classifier:

```
df=DecisionTreeClassifier(criterion='entropy',random_state=0)
df.fit(X_train,y_train)
preddf=df.predict(X_test)
print(classification_report(y_test,preddf))
```

K-neighbors Classifier:

```
knn=KNeighborsClassifier()  
knn.fit(X_train,y_train)  
predknn=knn.predict(X_test)  
print(classification_report(y_test,predknn))
```

Naive Bayes Classifier:

```
from sklearn.naive_bayes import GaussianNB  
nb = GaussianNB()  
  
nb.fit(X_train,y_train)  
prednb = nb.predict(X_test)  
print(classification_report(prednb,y_test))
```

XG Boost Classifier:

```
import xgboost as xgb  
xg=xgb.XGBClassifier()  
xg.fit(X_train,y_train)  
predxg = xg.predict(X_test)  
print(classification_report(prednb,y_test))
```

Ada Boost and Gradient Boost Classifier:

```
from sklearn.ensemble import AdaBoostClassifier, GradientBoostingClassifier
```

```
ab=AdaBoostClassifier()  
gb=GradientBoostingClassifier()
```

```
ab.fit(X_train,y_train)
```

```
▼ AdaBoostClassifier  
AdaBoostClassifier()
```

```
gb.fit(X_train,y_train)
```

```
▼ GradientBoostingClassifier  
GradientBoostingClassifier()
```

```
pred2= ab.predict(X_test)  
pred3= gb.predict(X_test)
```

Random Forest Classifier:

```
rf = RandomForestClassifier()
rf.fit(X_train,y_train)
predrf = rf.predict(X_test)
print(classification_report(predrf,y_test))
```

Artificial Neural Network:

```
ann = Sequential()
```

WARNING:tensorflow:From C:\Users\ramak\Anaconda\Lib\site-packages\keras\src\backend\tensorflow_v1_get_default_graph.py:107: tf.nn.conv2d is deprecated and will be removed in a future version. Use tf.nn.conv2d_v2 instead.

```
ann.add(Dense(14,activation='relu'))
ann.add(Dense(26,activation='relu'))
ann.add(Dense(26,activation='relu'))
ann.add(Dense(1,activation='sigmoid'))
ann.compile(loss="binary_crossentropy", optimizer='adam',metrics=['accuracy'])
```

WARNING:tensorflow:From C:\Users\ramak\Anaconda\Lib\site-packages\keras\src\optimizer\tf_optimizer.py:110: tf.nn.conv2d is deprecated and will be removed in a future version. Use tf.nn.conv2d_v2 instead.

```
ann.fit(X_train, y_train, epochs=50, batch_size=15)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Support vector machine	<pre> precision recall f1-score support 0 0.55 0.90 0.68 1312 1 0.88 0.51 0.65 1988 accuracy 0.66 3300 macro avg 0.71 0.70 0.66 3300 weighted avg 0.75 0.66 0.66 3300 </pre>	66%	<pre> [[1175 137] [976 1012]] </pre>
Logistic Regression	<pre> precision recall f1-score support 0 0.54 0.58 0.56 1312 1 0.71 0.67 0.69 1988 accuracy 0.64 3300 macro avg 0.63 0.63 0.63 3300 weighted avg 0.64 0.64 0.64 3300 </pre>	64%	<pre> [[766 546] [649 1339]] </pre>
Decision Tree Classifier	<pre> precision recall f1-score support 0 0.56 0.59 0.58 1312 1 0.72 0.70 0.71 1988 accuracy 0.66 3300 macro avg 0.64 0.64 0.64 3300 weighted avg 0.66 0.66 0.66 3300 </pre>	66%	<pre> [[774 538] [600 1388]] </pre>

K-neighbors Classifier	<pre> precision recall f1-score support 0 0.55 0.61 0.58 1312 1 0.72 0.67 0.70 1988 accuracy macro avg 0.64 0.64 0.65 3300 weighted avg 0.65 0.65 0.65 3300 </pre>	65%	[[802 510] [655 1333]]
Naive Bayes Classifier	<pre> precision recall f1-score support 0 0.99 0.53 0.69 2446 1 0.42 0.98 0.59 854 accuracy macro avg 0.71 0.76 0.64 3300 weighted avg 0.84 0.65 0.66 3300 </pre>	65%	[[1297 1149] [15 839]]
XG boost Classifier	<pre> precision recall f1-score support 0 0.99 0.53 0.69 2446 1 0.42 0.98 0.59 854 accuracy macro avg 0.71 0.76 0.64 3300 weighted avg 0.84 0.65 0.66 3300 </pre>	65%	[[1297 1149] [15 839]]
Ada Boost Classifier	<pre> precision recall f1-score support 0 0.57 0.76 0.65 1312 1 0.79 0.62 0.69 1988 accuracy macro avg 0.68 0.69 0.67 3300 weighted avg 0.70 0.67 0.68 3300 </pre>	67%	[[995 317] [765 1223]]
Gradient Boost Classifier	<pre> precision recall f1-score support 0 0.57 0.87 0.69 1312 1 0.86 0.56 0.68 1988 accuracy macro avg 0.72 0.71 0.68 3300 weighted avg 0.75 0.68 0.68 3300 </pre>	68%	[[1137 175] [870 1118]]
Random Forest Classifier	<pre> precision recall f1-score support 0 0.68 0.57 0.62 1570 1 0.66 0.76 0.71 1730 accuracy macro avg 0.67 0.66 0.66 3300 weighted avg 0.67 0.67 0.67 3300 </pre>	67%	[[894 676] [418 1312]]
Artificial Neural Network	<pre> precision recall f1-score support 0 0.56 0.69 0.62 1312 1 0.76 0.64 0.69 1988 accuracy macro avg 0.66 0.67 0.66 3300 weighted avg 0.68 0.66 0.66 3300 </pre>	66%	[[910 402] [722 1266]]