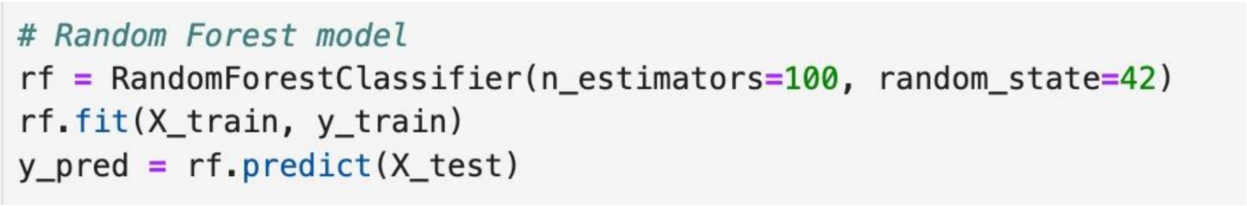
Model Development Phase Template

Date	11 July 2024
Team ID	**SWTID1720163281**
Project Title	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	4 Marks

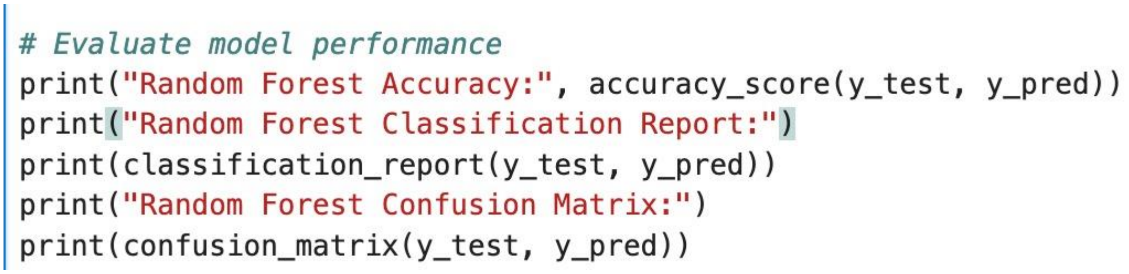
Initial Model Training Code, Model Validation and Evaluation Report

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.

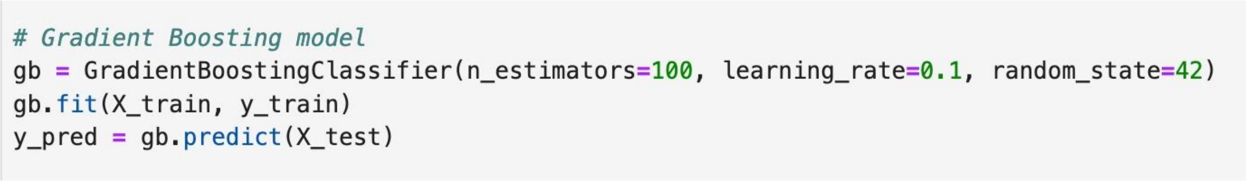
Initial Model Training Code:



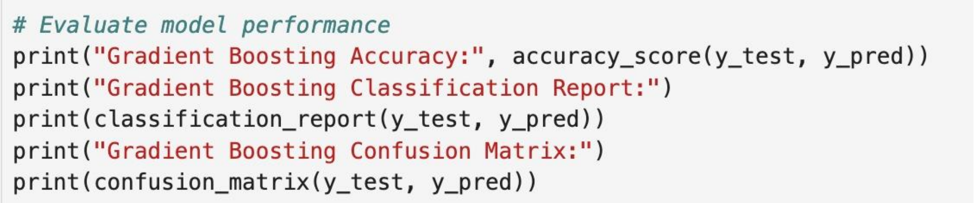
```
# Random Forest model
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```



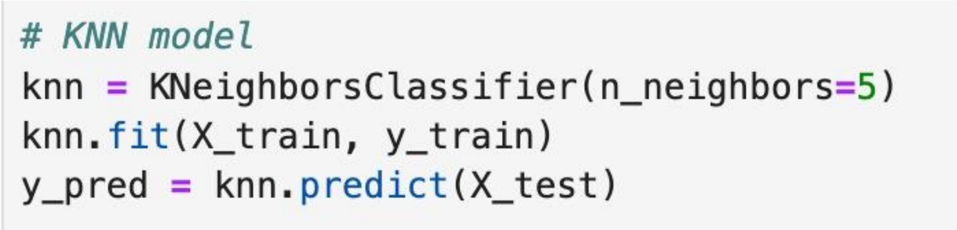
```
# Evaluate model performance
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred))
print("Random Forest Classification Report:")
print(classification_report(y_test, y_pred))
print("Random Forest Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```



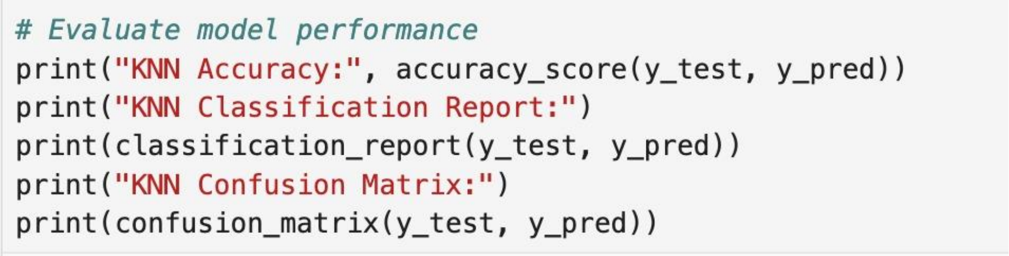
```
# Gradient Boosting model
gb = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, random_state=42)
gb.fit(X_train, y_train)
y_pred = gb.predict(X_test)
```



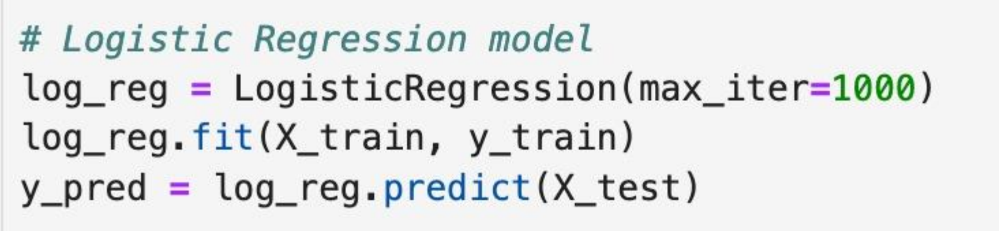
```
# Evaluate model performance
print("Gradient Boosting Accuracy:", accuracy_score(y_test, y_pred))
print("Gradient Boosting Classification Report:")
print(classification_report(y_test, y_pred))
print("Gradient Boosting Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```



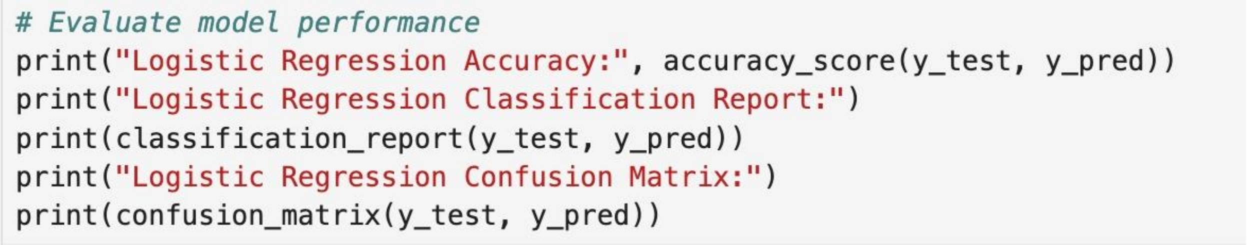
```
# KNN model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```



```
# Evaluate model performance
print("KNN Accuracy:", accuracy_score(y_test, y_pred))
print("KNN Classification Report:")
print(classification_report(y_test, y_pred))
print("KNN Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

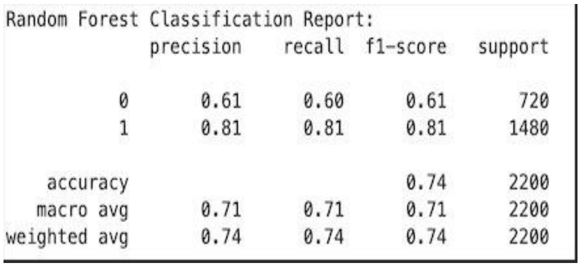
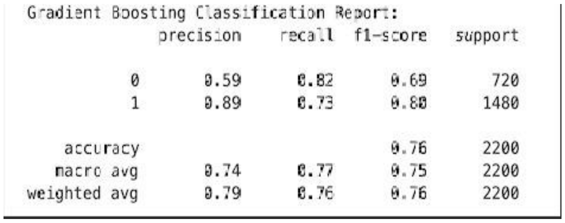
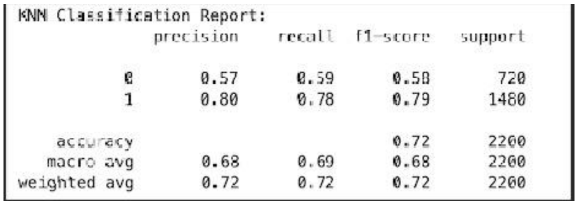


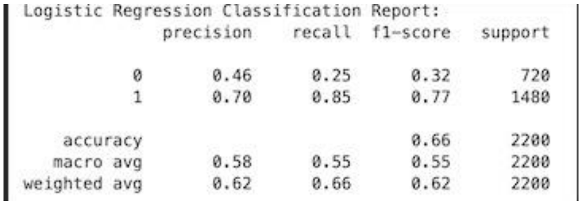
```
# Logistic Regression model
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
y_pred = log_reg.predict(X_test)
```



```
# Evaluate model performance
print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred))
print("Logistic Regression Classification Report:")
print(classification_report(y_test, y_pred))
print("Logistic Regression Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest	 Random Forest Classification Report: precision recall f1-score support 0 0.61 0.60 0.61 720 1 0.81 0.81 0.81 1480 accuracy 0.74 2200 macro avg 0.71 0.71 0.71 2200 weighted avg 0.74 0.74 0.74 2200	print("Random Forest Accuracy:", **accuracy_score(y_test, y_pred))** Random Forest Accuracy : 74.318	print("Random Forest Confusion Matrix:") print(confusion_matrix(y_test, y_pred)) Random Forest Confusion Matrix: [[434 286] [279 1201]]
GRADIENT BOOSTING	 Gradient Boosting Classification Report: precision recall f1-score support 0 0.59 0.82 0.69 720 1 0.89 0.73 0.80 1480 accuracy 0.76 2200 macro avg 0.74 0.77 0.75 2200 weighted avg 0.79 0.76 0.76 2200	print("Gradient Boosting Accuracy:**", accuracy_score(y_test, y_pred))** Gradient Boosting Accuracy: 75.772	print("Gradient Boosting Confusion Matrix:") print(confusion_matrix(y_test, y_pred)) Gradient Boosting Confusion Matrix: [[589 131] [402 1078]]
KNN	 KNN Classification Report: precision recall f1-score support 0 0.57 0.59 0.58 720 1 0.80 0.78 0.79 1480 accuracy 0.72 2200 macro avg 0.68 0.69 0.68 2200 weighted avg 0.72 0.72 0.72 2200	print("KNN Accuracy:", **accuracy_score(y_test, y_pred))** KNN Accuracy : 72.000	print("KNN Confusion Matrix:") print(confusion_matrix(y_test, y_pred)) KNN Confusion Matrix: [[425 295] [321 1159]]

Logistic Regression	
	Logistic Regression Classification Report:
	precision recall f1-score support
	0 0.46 0.25 0.32 720
	1 0.70 0.85 0.77 1480
	accuracy 0.66 2200
	macro avg 0.58 0.55 0.55 2200
	weighted avg 0.62 0.66 0.62 2200
		print("Logistic Regression Accuracy:"**,accuracy_score(y_test, y_pred))**
		Logistic Regression : 65.681
			print("Logistic Regression Confusion Matrix:")
			print(confusion_matrix(y_test, y_pred))