Model Development Phase Template

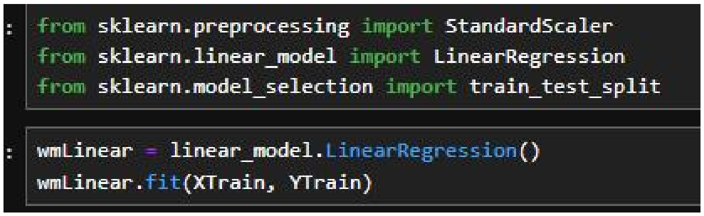
Date	11th july 2024
Team ID	SWTID1720435231
Project Title	Walmart Sales Analysis For Retail Industry With Machine Learning
Maximum Marks	4 Marks

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.

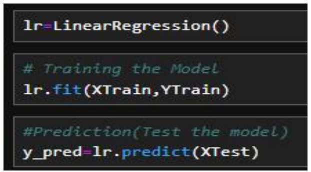
Initial Model Training Code:

Linear regression:



```
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

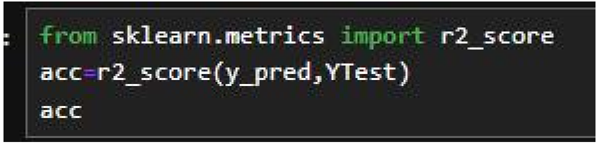
```
wmLinear = linear_model.LinearRegression()
wmLinear.fit(XTrain, YTrain)
```



```
lr=LinearRegression()
```

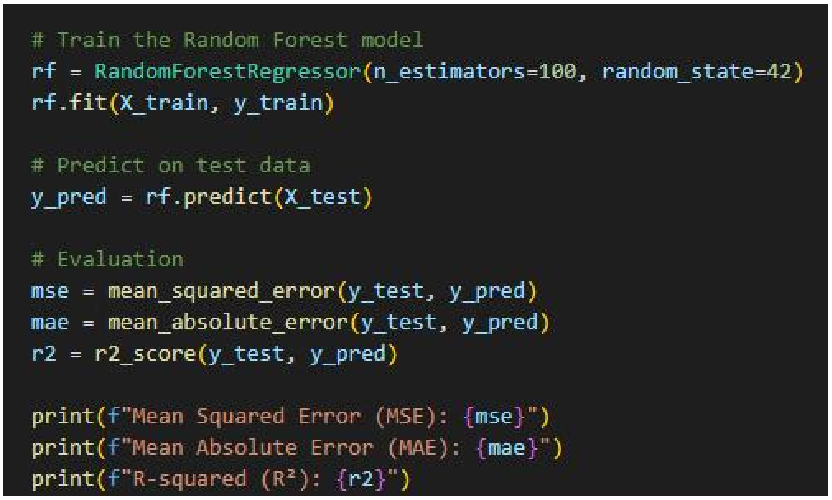
```
# Training the Model
lr.fit(XTrain,YTrain)
```

```
#Prediction(Test the model)
y_pred=lr.predict(XTest)
```



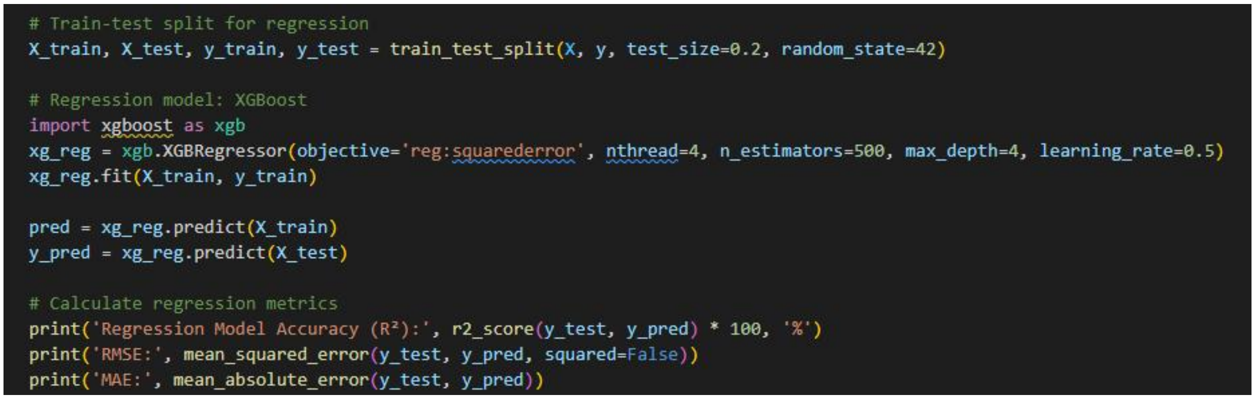
```
from sklearn.metrics import r2_score
acc=r2_score(y_pred,YTest)
acc
```

Random forest:



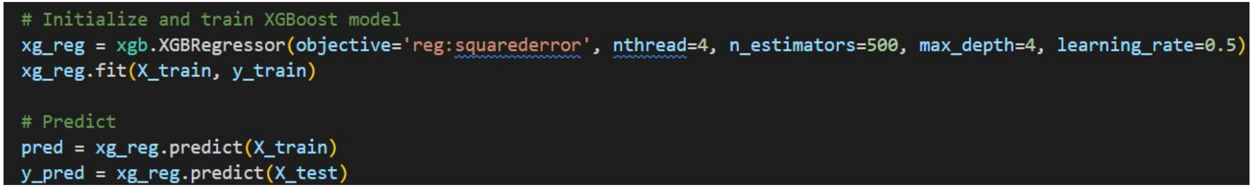
```
# Train the Random Forest model
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)

# Predict on test data
y_pred = rf.predict(X_test)

# Evaluation
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse}")
print(f"Mean Absolute Error (MAE): {mae}")
print(f"R-squared (R²): {r2}")
```

Arima:

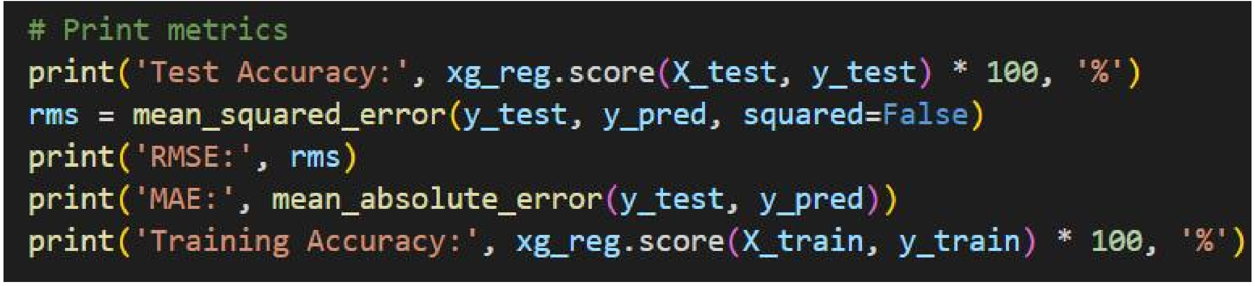


```
# Train-test split for regression
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Regression model: XGBoost
import xgboost as xgb
xg_reg = xgb.XGBRegressor(objective='reg:squarederror', nthread=4, n_estimators=500, max_depth=4, learning_rate=0.5)
xg_reg.fit(X_train, y_train)

pred = xg_reg.predict(X_train)
y_pred = xg_reg.predict(X_test)

# Calculate regression metrics
print('Regression Model Accuracy (R²):', r2_score(y_test, y_pred) * 100, '%')
print('RMSE:', mean_squared_error(y_test, y_pred, squared=False))
print('MAE:', mean_absolute_error(y_test, y_pred))
```

Xgboost:

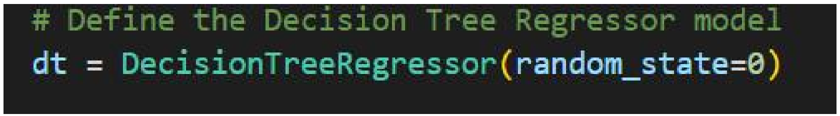


```
# Initialize and train XGBoost model
xg_reg = xgb.XGBRegressor(objective='reg:squarederror', nthread=4, n_estimators=500, max_depth=4, learning_rate=0.5)
xg_reg.fit(X_train, y_train)

# Predict
pred = xg_reg.predict(X_train)
y_pred = xg_reg.predict(X_test)
```

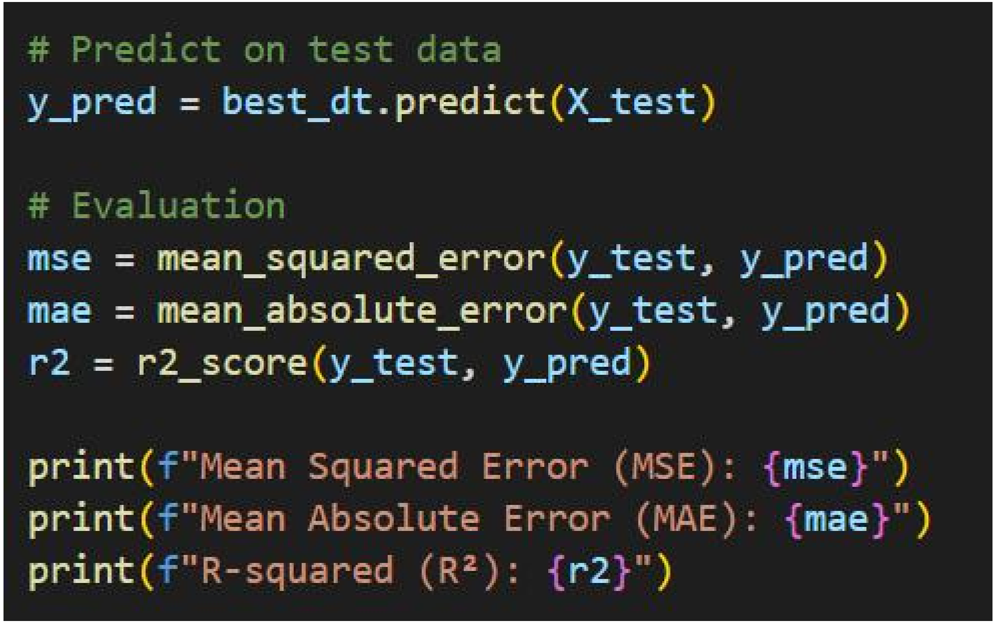


```
# Print metrics
print('Test Accuracy:', xg_reg.score(X_test, y_test) * 100, '%')
rms = mean_squared_error(y_test, y_pred, squared=False)
print('RMSE:', rms)
print('MAE:', mean_absolute_error(y_test, y_pred))
print('Training Accuracy:', xg_reg.score(X_train, y_train) * 100, '%')
```

Decision Tree:

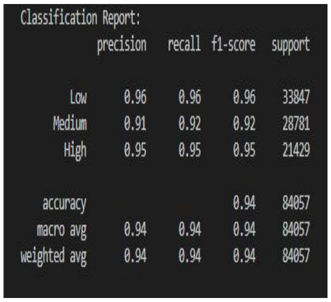
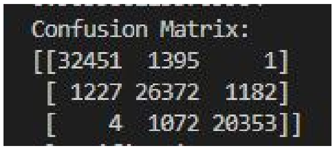


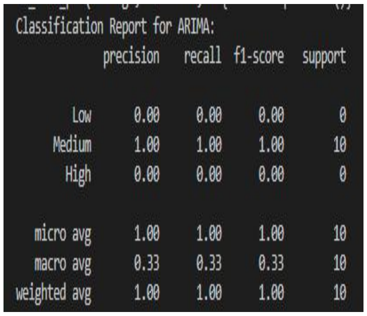
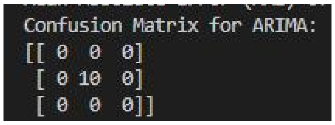
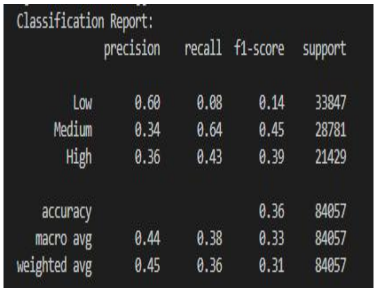
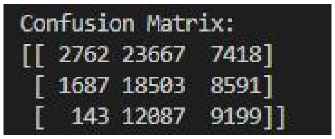
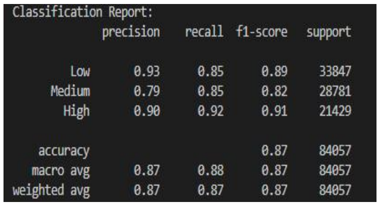
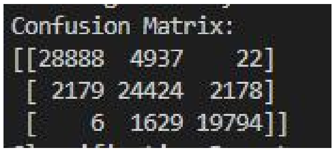
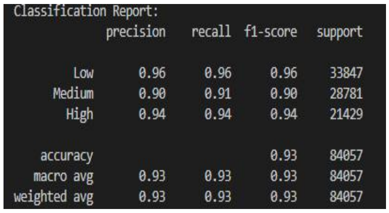
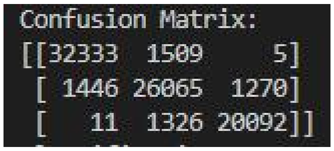
```
# Define the Decision Tree Regressor model
dt = DecisionTreeRegressor(random_state=0)
```



```
# Predict on test data
y_pred = best_dt.predict(X_test)

# Evaluation
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse}")
print(f"Mean Absolute Error (MAE): {mae}")
print(f"R-squared (R²): {r2}")
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random forest Model	 Classification Report: precision recall f1-score support Low 0.96 0.96 0.96 33847 Medium 0.91 0.92 0.92 28781 High 0.95 0.95 0.95 21429 accuracy 0.94 84057 macro avg 0.94 0.94 0.94 84057 weighted avg 0.94 0.94 0.94 84057	Accuracy: **96%**	 Confusion Matrix: [[32451 1395 1] [1227 26372 1182] [4 1072 20353]]

ARIMA Model	 Classification Report for ARIMA: precision recall f1-score support Low 0.00 0.00 0.00 0 Medium 1.00 1.00 1.00 10 High 0.00 0.00 0.00 0 micro avg 1.00 1.00 1.00 10 macro avg 0.33 0.33 0.33 10 weighted avg 1.00 1.00 1.00 10	Accuracy: **97 %**	 Confusion Matrix for ARIMA: [[0 0 0] [0 10 0] [0 0 0]]
Linear Regression Model	 Classification Report: precision recall f1-score support Low 0.60 0.08 0.14 33847 Medium 0.34 0.64 0.45 28781 High 0.36 0.43 0.39 21429 accuracy 0.36 84057 macro avg 0.44 0.38 0.33 84057 weighted avg 0.45 0.36 0.31 84057	Accuracy: **89 %**	 Confusion Matrix: [[2762 23667 7418] [1687 18503 8591] [143 12087 9199]]
Xgboost Model	 Classification Report: precision recall f1-score support Low 0.93 0.85 0.89 33847 Medium 0.79 0.85 0.82 28781 High 0.90 0.92 0.91 21429 accuracy 0.87 84057 macro avg 0.87 0.88 0.87 84057 weighted avg 0.87 0.87 0.87 84057	Accuracy: **93 %**	 Confusion Matrix: [[28888 4937 22] [2179 24424 2178] [6 1629 19794]]
Decision tree Model	 Classification Report: precision recall f1-score support Low 0.96 0.96 0.96 33847 Medium 0.90 0.91 0.90 28781 High 0.94 0.94 0.94 21429 accuracy 0.93 84057 macro avg 0.93 0.93 0.93 84057 weighted avg 0.93 0.93 0.93 84057	Accuracy: **93 %**	 Confusion Matrix: [[32333 1509 5] [1446 26065 1270] [11 1326 20092]]