Model Development Phase

Date	21 March 2024
Team ID	738220
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:

```
from sklearn.ensemble import RandomForestRegressor
# Create a Random Forest Regressor model
rf_model = RandomForestRegressor(n_estimators=150, max_depth=30,
                                 min_samples_split=5, min_samples_leaf=1,
                                 random_state=42)

# Train the model on the training set
rf_model.fit(X_train, y_train)

# Make predictions on the test set
rf_predictions = rf_model.predict(X_test)

# Calculate the R^2 score of the model
rf_score = rf_model.score(X_test, y_test) * 100

# Print the R^2 score
print(f"Random Forest R^2 Score: {rf_score:.2f}%")
```

```python
from sklearn.tree import DecisionTreeRegressor
# Create a decision tree regressor model
dt_model = DecisionTreeRegressor(random_state=42)

# Train the model on the training set
dt_model.fit(X_train, y_train)

# Make predictions on the test set
dt_predictions = dt_model.predict(X_test)

# Calculate the R^2 score of the model
dt_score = dt_model.score(X_test, y_test) * 100

# Print the R^2 score
print(f"Decision Tree R^2 Score: {dt_score:.2f}%")

# Calculate the Mean Absolute Error (MAE)
dt_mae = mean_absolute_error(y_test, dt_predictions)

# Calculate the Root Mean Squared Error (RMSE)
dt_rmse = np.sqrt(mean_squared_error(y_test, dt_predictions))

# Print the MAE and RMSE values
print(f"Decision Tree MAE: {dt_mae:.2f}")
print(f"Decision Tree RMSE: {dt_rmse:.2f}")
```

```python
# Calculate the training accuracy for the Random Forest model
rf_train_accuracy = rf_model.score(X_train, y_train) * 100

# Calculate the training accuracy for the Decision Tree model
dt_train_accuracy = dt_model.score(X_train, y_train) * 100

# Print the training accuracy for both models
print(f"Random Forest Training Accuracy: {rf_train_accuracy:.2f}%")
print(f"Decision Tree Training Accuracy: {dt_train_accuracy:.2f}%")
```

```python
import xgboost as xgb
from sklearn.metrics import mean_squared_error, mean_absolute_error

# Create an XGBoost regressor model
xgb_model = xgb.XGBRegressor(objective='reg:squarederror', nthread=4, n_estimators=1000, max_depth=5, learning_r

# Train the model on the training set
xgb_model.fit(X_train, y_train)

# Make predictions on the test set
xgb_predictions = xgb_model.predict(X_test)

# Calculate the R^2 score of the model
xgb_score = xgb_model.score(X_test, y_test) * 100

# Print the R^2 score
print(f"XGBoost R^2 Score: {xgb_score:.2f}%")

# Calculate the Mean Absolute Error (MAE)
xgb_mae = mean_absolute_error(y_test, xgb_predictions)

# Calculate the Root Mean Squared Error (RMSE)
xgb_rmse = np.sqrt(mean_squared_error(y_test, xgb_predictions))

# Print the MAE and RMSE values
print(f"XGBoost MAE: {xgb_mae:.2f}")
print(f"XGBoost RMSE: {xgb_rmse:.2f}")

# Calculate the training accuracy for the XGBoost model
xgb_train_accuracy = xgb_model.score(X_train, y_train) * 100

# Print the training accuracy
print(f"XGBoost Training Accuracy: {xgb_train_accuracy:.2f}%")
```

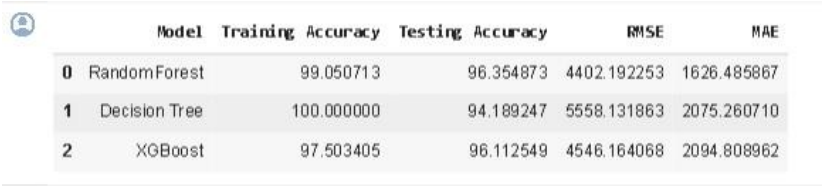
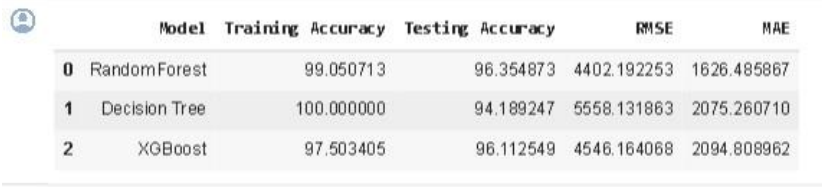
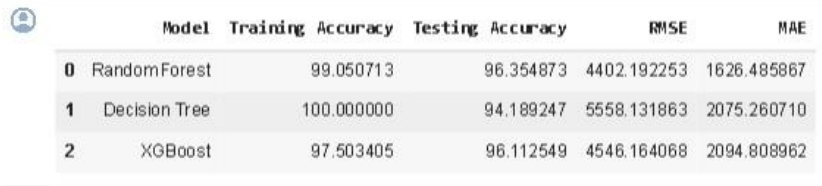
```python
# Calculate the Root Mean Squared Error (RMSE)
arima_rmse = np.sqrt(mean_squared_error(test_data, forecast))

# Calculate the Mean Squared Error (MSE)
arima_mse = mean_squared_error(test_data, forecast)

# Calculate the Mean Absolute Error (MAD)
arima_mad = mean_absolute_error(test_data, forecast)

# Print the RMSE, MSE, and MAD
print(f"RMSE: {rmse:.2f}")
print(f"MSE: {mse:.2f}")
print(f"MAD: {mad:.2f}")
```

Model Validation and Evaluation Report:

Model	Classification Report
Random Regression	
Decision Tree	
Xgboost	

	Model	Training Accuracy	Testing Accuracy	RMSE	MAE
0	RandomForest	99.050713	96.354873	4402.192253	1626.485867
1	Decision Tree	100.000000	94.189247	5558.131863	2075.260710
2	XGBoost	97.503405	96.112549	4546.164068	2094.808962

Arima	+---------------+--------------------+--------------------+--------------------+---------------------+ | Model | Training Accuracy | Test Accuracy | RMSE | MAE /MAD (Arima) | +---------------+--------------------+--------------------+--------------------+---------------------+ | Random Forest | 99.05071258942313 | 96.35487250338254 | 4402.192252783296 | 1626.4858674570846 | | Decision Tree | 100.0 | 94.18924713750879 | 5558.131862816404 | 2075.2607103422015 | | XGBooost | 97.50340544072975 | 96.1125490042831 | 4546.164067935629 | 2094.8089620184737 | | Arima | - | - | 686.6408829232673 | 448.5092795868826 | +---------------+--------------------+--------------------+--------------------+---------------------+