



Model Optimization and Tuning Phase

Date	22 March 2024
Team ID	738220
Project Title	Walmart Sales Analysis for Retail Industry with Machine Learning
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

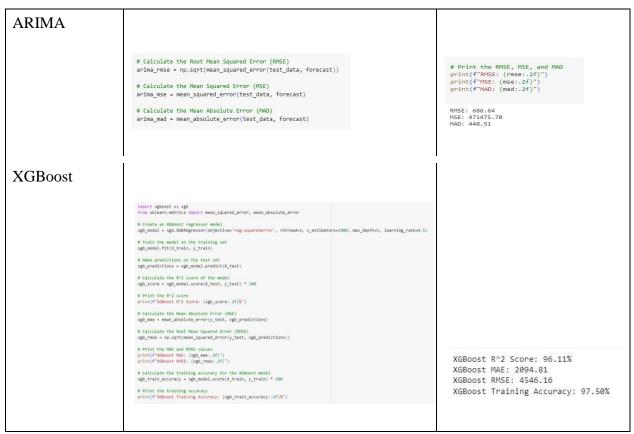
Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Random Forest		<pre># Print the MAE and RMSE values print(f"Random Forest MAE: {rf_mae:.2f}") print(f"Random Forest RMSE: {rf_rmse:.2f}")</pre>
	from sklearn.ensemble import RandomforestRegressor # Create & Random Forest Regressor model rf_model = RandomforestRegressor(n_estimators=150, max_depth=30, min_samples_split=5, min_samples_leaf=1, random_state=62) # Trais the model on the training set rf_model.fick(train, y_train) # Rake predictions on the test set	Random Forest MAE: 1626.49 Random Forest RMSE: 4402.19 # Print the R^2 score print(f"Random Forest R^2 Score: {rf_score:.2f}%")
	of_predictions = rf_model.predict(X_test) # Calculate the 8'2 score of the model rf_score = rf_model.score(_test, y_test) * 100	Random Forest R^2 Score: 96.35%
	# Calculate the Mean Absolute Error (MAE) rf_mae - mean_absolute_error(y_test, rf_predictions) # Calculates the Boot Mean Squared Error (UMSE) rf_mse - np.sqrt(mean_squared_error(y_test, rf_predictions))	Random Forest Training Accuracy: 99.85%
	<pre># Calculate the training accuracy for the Random Forest model rf_train_accuracy = rf_model.score(X_train, y_train) * 100</pre>	





Print the MAE and RMSE values print(f"Decision Tree MAE: {dt_mae:.2f}") print(f"Decision Tree RMSE: {dt_mae:.2f}") print(f"Decision Tree RMSE: {dt_mae:.2f}") print(f"Decision Tree RMSE: {dt_mse:.2f}") # Trais the model on the training set dt_model.fit(X_train, y_train) # Rais predictions on the text set dt_predictions - dt_model.score(X_train) # RMSE predictions - dt_model.score(X_text) # Calculate the R*3 score of the model dt_score - dt_model.score(X_text, y_text) *100 # Print the R*2 score print(f"Decision Tree RMSE: {dt_mse:.2f}") # Calculate the R*3 score of the model dt_score - dt_model.score(X_text, y_text) *100 # Print the R*2 score print(f"Decision Tree RMSE: {dt_mse:.2f}") # Calculate the Read Absolute From (MBE) dt_mse = msem_absolute_error(y_text, dt_predictions)) # Calculate the Rout Rean Squared From (MBE) dt_mse = msem_closedure_error(y_text, dt_predictions)) # Calculate the training accuracy for the Decision Tree model dt_train_accuracy - dt_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



Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric





Decision Tree	+	+			
	Model	Training Accuracy	Test Accuracy	RMSE	MAE /MAD (Arima)
	Decision Tree	100.0	94.18924713750879	5558.131862816404	2075.2607103422015
Random Forest					
	+	.+	+	÷	-+
	Model	Training Accuracy	Test Accuracy	RMSE	MAE /MAD (Arima)
	Random Forest	99.05071258942313	96.35487250338254	4402.192252783296	1626.4858674570846
A D IN // A	İ				
ARIMA	Model	Training Accuracy	Test Accuracy	+ RMSE	MAE /MAD (Arima)
	Arima	-	 -	686.6408829232673	448.5092795868826
XGBoost					
	Model	Training Accuracy	Test Accuracy	RMSE	MAE /MAD (Arima)
	XGBooost	97.50340544072975	96.112549042831	4546.164067935629	2094.8089620184737

Final Model Selection Justification (2 Marks):

Final Model	Reasoning		
Decision Tree	The Decision Tree model was selected for its superior performance, exhibiting high accuracy during hyperparameter tuning. Its ability to handle complex relationships, minimize overfitting, and optimize predictive accuracy aligns with project objectives, justifying its selection as the final model.		