Model Optimization and Tuning Phase Template

Date	5th July 2024
Team ID	739958
Project Title	Food demand forecasting for food delivery company
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
KNN	KNN = KNeighborsRegressor() KNN.fit(X_train, y_train) y_pred = KNN.predict(X_val) y_pred[y_pred<0] = 0 from sklearn import metrics print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred))) -	-RMSLE: 67.31466422917168
Decision Tree	- DT = DecisionTreeRegressor() DT.fit(X_train, y_train) y_pred = DT.predict(X_val) y_pred[y_pred<0] = 0 from sklearn import metrics print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))	-RMSLE: 62.6445830592777

Gradient Boosting Regressor	-
	`GB = GradientBoostingRegressor()`
	`GB.fit(X_train, y_train)`
	`y_pred = GB.predict(X_val)`
	`y_pred[y_pred<0] = 0`
	`from sklearn import metrics`
	`print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))`
		-RMSLE:
		98.97455800242957

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric
KNN	-	-
Decision Tree	-	-
Gradient Boosting	-	-

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Gradient Boosting	-The Gradient Boosting 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.