Model Optimization and Tuning Phase Template

Date	17 July 2024
Team ID	SWTID1720527361
Project Title	TrafficTelligence: Advanced Traffic Volume Estimation 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

XG Boost	`model=xgb.XGBRegressor()`
	
	`parameters={`
	`'max_depth': [3, 5, 8],`
	`'min_child_weight': [1, 3, 5],`
	`'eta': [0.1, 0.3, 0.5],`
	`'subsample': [0.6, 0.8, 1],`
	`'colsample_bytree': [0.6, 0.8, 1]`
	`}`
		`y_pred=clf.predict(x_test)`
		`print("Best Score: ", r2_score(y_test, y_pred))`
		`clf.best_params_`
		
		`Best Score: 0.9676877994811365`
		`{'colsample_bytree': 1,`
		`'eta': 0.3,`
		`'max_depth': 8,`
		`'min_child_weight': 1,`
		`'subsample': 1}`

Random Forest Regressor	`#model Initialization`
	`regressor = RandomForestRegressor()`
	
	`#Parameters`
	`parameters={`
	`'n_estimators':[20, 50, 100],`
	`'bootstrap':[True, False]`
	`}`
		`y_pred=clf.predict(x_test)`
		`print("Best Score: ", r2_score(y_test, y_pred))`
		`print("Best Values: ", clf.best_params_)`
		
		Best Score: 0.9556679960267289
		Best Values: {'bootstrap': True, 'n_estimators': 100}

Polynomial Regression	`model=LinearRegression()`
	
	`parameters={`
	`'fit_intercept':[ True, False],`
	`'positive':[True, False]`
	`}`
		`y_pred=clf.predict(x_test)`
		`print("Best Score: ", r2_score(y_test, y_pred))`
		`print("Best Values: ", clf.best_params_)`
		
		Best Score: 0.7686065818544895
		Best Values: {'fit_intercept': True, 'positive': False}

SVR	`model=SVR()`
	
	`parameters={`
	`'C': [0.1, 1, 10],`
	`'kernel': ['linear', 'rbf'],`
	`'gamma': [0.1, 1, 10],`
	`'epsilon': [0.1, 0.5, 1]`
	`}`
		`y_pred=clf.predict(x_test)`
		`print("Best Score: ", r2_score(y_test, y_pred))`
		`clf.best_params_`
		
		Best Score: 0.6402522031519096
		{'kernel': 'rbf', 'gamma': 10, 'epsilon': 0.5, 'C': 10}

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric

XG Boost	`from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error`
	`print("Mean Square Error: ", mean_squared_error(y_test, y_pred))`
	`print("Mean Absolute Error: ", mean_absolute_error(y_test, y_pred))`
	`print("R-square Score: ", r2_score(y_test, y_pred))`
	
	Mean Square Error: 120958.54825379612
	Mean Absolute Error: 228.5786688810355
	R-square Score: 0.9563201748182905
		`y_pred=clf.predict(x_test)`
		`print("Mean Square Error: ", mean_squared_error(y_test, y_pred))`
		`print("Mean Absolute Error: ", mean_absolute_error(y_test, y_pred))`
		`print("R-square Score: ", r2_score(y_test, y_pred))`
		
		Mean Square Error: 87733.82381560856
		Mean Absolute Error: 199.1112719822503
		R-square Score: 0.9676877994811365

Random Forest Regressor	`from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error`
	`print("Mean Square Error: ", mean_squared_error(y_test, y_pred))`
	`print("Mean Absolute Error: ", mean_absolute_error(y_test, y_pred))`
	`print("R-square Score: ", r2_score(y_test, y_pred))`
	Mean Square Error: 198008.8816750678
	Mean Absolute Error: 280.9588373446771
	R-square Score: 0.9277791335225944
		`y_pred=clf.predict(x_test)`
		`print("Mean Square Error: ", mean_squared_error(y_test, y_pred))`
		`print("Mean Absolute Error: ", mean_absolute_error(y_test, y_pred))`
		`print("R-square Score: ", r2_score(y_test, y_pred))`
		Mean Square Error: 120881.52201975712
		Mean Absolute Error: 219.65486158265864
		R-square Score: 0.9556679960267289

Polynomial Regression	`from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error`
	`print("Mean Square Error: ", mean_squared_error(y_test, y_pred))`
	`print("Mean Absolute Error: ", mean_absolute_error(y_test, y_pred))`
	`print("R-square Score: ", r2_score(y_test, y_pred))`
	Mean Square Error: 646496.8829842781
	Mean Absolute Error: 588.6552844192978
	R-square Score: 0.7605639174654056
		`y_pred=clf.predict(x_test)`
		`print("Mean Square Error: ", mean_squared_error(y_test, y_pred))`
		`print("Mean Absolute Error: ", mean_absolute_error(y_test, y_pred))`
		`print("R-square Score: ", r2_score(y_test, y_pred))`
		Mean Square Error: 629758.8109993833
		Mean Absolute Error: 591.4450058414657
		R-square Score: 0.7686065818544895

SVR	`from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error`
	`print("Mean Square Error: ", mean_squared_error(y_test, y_pred))`
	`print("Mean Absolute Error: ", mean_absolute_error(y_test, y_pred))`
	`print("R-square Score: ", r2_score(y_test, y_pred))`
	Mean Square Error: 2104039.9111552383
	Mean Absolute Error: 1256.7067758496808
	R-square Score: 0.23504232546490522
		`y_pred=clf.predict(x_test)`
		`print("Mean Square Error: ", mean_squared_error(y_test, y_pred))`
		`print("Mean Absolute Error: ", mean_absolute_error(y_test, y_pred))`
		`print("R-square Score: ", r2_score(y_test, y_pred))`
		Mean Square Error: 974323.6860184855
		Mean Absolute Error: 760.1124997263398
		R-square Score: 0.6402522031519096

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
XG Boost	This model had highest R2-Score before optimization and also it has highest R2-Score after optimization of 96.8%. It is selected for its highest performance among all other mode after hypertuning.