Model Optimization and Tuning Phase Template

Date	11 July 2024
Team ID	SWTID1720359900
Project Title	Machine Learning Approach For Predicting The Price Of Natural Gas
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

Linear Regression	`lr.get_params()`
	`{'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'positive': False}`
	`from sklearn.model_selection import GridSearchCV`
	`param_space = {'copy_X': [True,False], 'fit_intercept': [True,False], 'n_jobs': [1,5,10,15,None], 'positive': [True,False]}`
	`grid_search = GridSearchCV(lr, param_space, cv=5)`
	`grid_search.fit(X_train, y_train)`
		`# Parameter which gives the best results`
		`print(f"Best Hyperparameters: {grid_search.best_params_}")`
		`# Accuracy of the model after using best parameters`
		`print(f"Best Score: {grid_search.best_score_}")`
		`Best Hyperparameters: {'copy_X': True, 'fit_intercept': True, 'n_jobs': 1, 'positive': False}`
		`Best Score: 0.032457590729715526`

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric

Decision Tree	`from sklearn.tree import DecisionTreeRegressor`
	`from sklearn.metrics import mean_squared_error`
	`regressor = DecisionTreeRegressor(random_state=42)`
	
	`regressor.fit(X_train, y_train)`
	
	`y_pred = regressor.predict(X_test)`
	
	`mse = mean_squared_error(y_test, y_pred)`
	`print(f"Mean Squared Error (MSE): {mse}")`
	
	`Mean Squared Error (MSE): 0.09586742424242424`
		`from sklearn.metrics import r2_score`
		
		`r2 = r2_score(y_test, y_pred)`
		
		`print(f'R-squared (R2) Score: {r2:.4f}')`
		
		`R-squared (R2) Score: 0.9814`

Linear Regression	`from sklearn.linear_model import LinearRegression`
	
	`lr = LinearRegression()`
	
	`lr.fit(X_train,y_train)`
	
	`LinearRegression`
	`LinearRegression()`
	
	`ypred = lr.predict(X_test)`
	
	`ypred`
	
	`array([4.45282588, 4.05787683, 3.74130395, ..., 4.14557995, 4.11678668, 3.98965119])`
		`from sklearn.metrics import r2_score`
		`acc = r2_score(y_pred,y_test)`
		`acc`
		
		`0.9808376083629289`

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Decision Tree	As it gives better accuracy and performance than linear regression.