# Model Optimization and Tuning Phase Template

| | |
|---|---|
| Date | 10 July 2024 |
| Team ID | 740064 |
| Project Title | Trip-Based Modelling of Fuel Consumption in Modern Fleet Vehicles Using Machine Learning |
| Maximum Marks | 10 Marks |

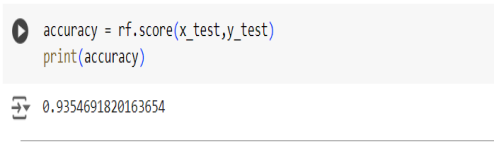
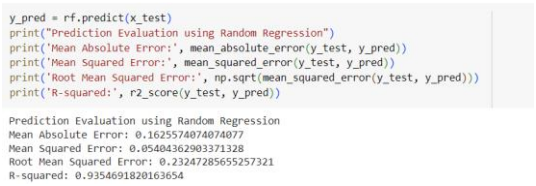
**Model Optimization and Tuning Phase**

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 |
|---|---|---|
| Linear Regression | No Hyperparameters used | ---------- |
| Lasso Regression | No Hyperparameters used | ---------- |
| SVM | No Hyperparameters used | ---------- |
| Decision Tree | No Hyperparameters used | ---------- |
| Random Forest | No Hyperparameters used | ---------- |

| Model | Accuracy | Metrics |
|---|---|---|
| Linear Regression | `print(linReg.coef_,linReg.intercept_)`<br>`[ 0.00523674 -0.02371772 -0.14711979 -0.03724498  0.41456804  0.61676684 -0.06407861] 9.389308142257129`<br>`accuracy = linReg.score(x_test,y_test)`<br>`print(accuracy)`<br>`0.1134733714697449` | `[58] print("Prediction Evaluation using Linear Regression")`<br>`print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))`<br>`print('Mean Squared Error:', mean_squared_error(y_test, y_pred))`<br>`print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))`<br>`print('R-squared:', r2_score(y_test, y_pred))`<br>Prediction Evaluation using Linear Regression<br>Mean Absolute Error: 0.6635761182069623<br>Mean Squared Error: 0.742453260904708<br>Root Mean Squared Error: 0.8616572757800562<br>R-squared: 0.1134733714697449 |
| Lasso Regression | `accuracy = lassoReg.score(x_test,y_test)`<br>`print(accuracy)`<br>`0.1456141532515728` | `y_pred = lassoReg.predict(x_test)`<br>`print("Prediction Evaluation using lasso Regression")`<br>`print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))`<br>`print('Mean Squared Error:', mean_squared_error(y_test, y_pred))`<br>`print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))`<br>`print('R-squared:', r2_score(y_test, y_pred))`<br>Prediction Evaluation using lasso Regression<br>Mean Absolute Error: 0.6296444264267669<br>Mean Squared Error: 0.7155358198781405<br>Root Mean Squared Error: 0.8458935038633058<br>R-squared: 0.1456141532515728 |
| SVM | `[43] #SVM MODEL`<br>`[44] svr = SVR().fit(x,y)`<br>`[45] y_pred = svr.predict(x_test)`<br>`[46] accuracy = svr.score(x_test,y_test)`<br>`print(accuracy)`<br>`0.4176454053391483` | `y_pred = svr.predict(x_test)`<br>`print("Prediction Evaluation using svr Regression")`<br>`print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))`<br>`print('Mean Squared Error:', mean_squared_error(y_test, y_pred))`<br>`print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))`<br>`print('R-squared:', r2_score(y_test, y_pred))`<br>Prediction Evaluation using svr Regression<br>Mean Absolute Error: 0.49633196595520446<br>Mean Squared Error: 0.48771357102448615<br>Root Mean Squared Error: 0.698364926828722<br>R-squared: 0.4176454053391483 |
| Decision Tree | `[48] dt = DecisionTreeRegressor(random_state = 0)`<br>`dt.fit(x,y)`<br>`DecisionTreeRegressor(random_state=0)`<br>`[49] y_pred = dt.predict(x_test)`<br>`print(y_pred)`<br>`[5.6 5.1 5.55 4.6 4.3 7.1 5. 6.8 5.1 4.7 4.3 4.4 5.4 6.2 7.4 4.7 5.1 4.6 4.9 5. 4.2 4.6 4.9 4.6 4.8 5.1 4.6 2.8 4.5 4.2 4.8 5.2 3.8 4.2 4.9 4.4 4.1 4.6 2.6 5.4 5.4 4.9 4.3 5.5 6.5 4.2 6. 5. 3.6 4.9 3.9 4.9 5.7 5. 4.9 3.9 5.9 5.1 5.3 4.1 6.8 4.1 4.7 3.9 5.3 6.4 5.9 4.1 4. 5.2 5. 4.7 5.7 3.9 4.9 4.7 4.7 4.4 5.2 4.6 4.5 5.8 4.1 5.1 4.6 5.2 5. 4.6 4.3 4.1 5. 4. 5.4 4.8 4. 5.6 4.6 4.4 5.55 4.7 4.5 5. 3.7 3.7 4.4 4.3 4. 5.6 7.9 4. 3.9 7.25 6.1 3.7 5.1 5.1 4.1 ]`<br>`accuracy = dt.score(x_test,y_test)`<br>`print(accuracy)`<br>`0.0864521202267205` | `y_pred = dt.predict(x_test)`<br>`print("Prediction Evaluation using decisiontree Regression")`<br>`print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))`<br>`print('Mean Squared Error:', mean_squared_error(y_test, y_pred))`<br>`print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))`<br>`print('R-squared:', r2_score(y_test, y_pred))`<br>Prediction Evaluation using decisiontree Regression<br>Mean Absolute Error: 0.01666666666666666<br>Mean Squared Error: 0.011346153846153837<br>Root Mean Squared Error: 0.10651832633943249<br>R-squared: 0.9864521202267205 |

| | | |
|---|---|---|
| Random Forest | 4.9075 4.945 4.295 ]<br>  <br> |  <br> |

**Performance Metrics Comparison Report (2 Marks):**

**Final Model Selection**
**Justification (2 Marks):**

| Final Model Selection | Reasoning |
|---|---|
| Decision Tree | Decision Tree model was selected for its superior performance, exhibiting high accuracy than any other models .<br><br>We chose the decision tree because it gives very accurate predictions, can handle complex patterns in data, and avoids overfitting. It works well with different types of data and allows us to see which features are most important. This makes it a reliable and effective model for our task |