# Model Development Phase Template

| | |
|---|---|
| Date | 10 July 2024 |
| Team ID | 739652 |
| Project Title | Trip-Based Modelling of Fuel Consumption in Modern Fleet Vehicles Using Machine Learning |
| Maximum Marks | 4 Marks |

**Initial Model Training Code, Model Validation and Evaluation Report**

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

**Initial Model Training Code:**

Paste the screenshot of the model training code

```
[18] from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
```

```
#Splitting Data Into Train And Test
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=42)
```

Linear regression

```
[25] #Linear Regression
```

```
linReg = LinearRegression()
linReg.fit(x_train,y_train)
```

```
LinearRegression
LinearRegression()
```

Lasso Regression

```
[39] #Lasso Regression model
```

```
lassoReg = linear_model.Lasso(alpha = 0.1)
lassoReg.fit(x,y)
```

```
Lasso
Lasso(alpha=0.1)
```

Decision Tree

```
[47] #Decision Tree Model:
```

```
dt = DecisionTreeRegressor(random_state = 0)
dt.fit(x,y)
```

```
DecisionTreeRegressor
DecisionTreeRegressor(random_state=0)
```

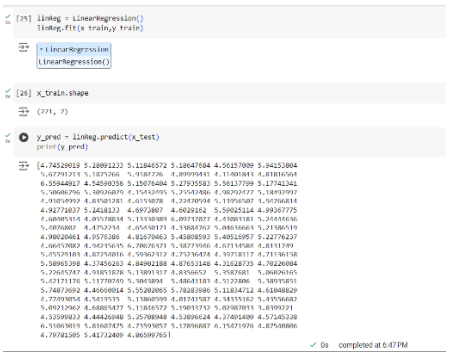
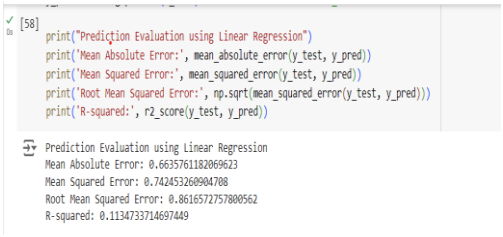
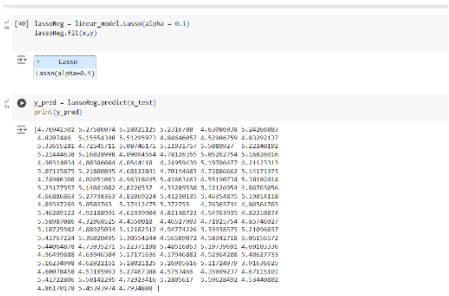
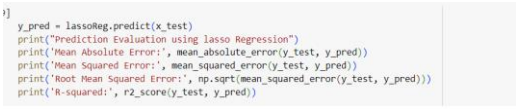
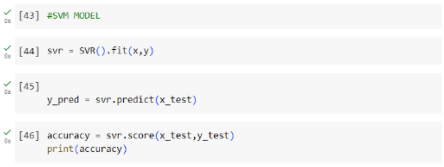
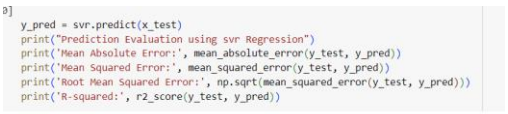
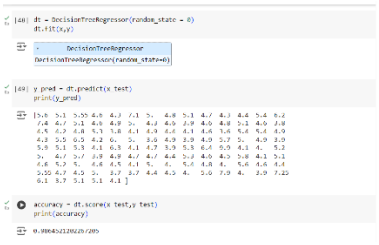
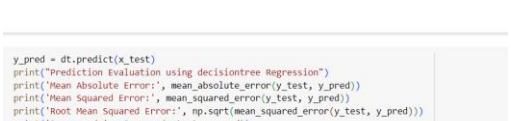
Random Forest

```
[51] #Random Forest Model:
```

```
[52] rf = RandomForestRegressor(n_estimators = 100 , random_state = 0)
     rf.fit(x,y)
```

```
RandomForestRegressor
RandomForestRegressor(random_state=0)
```

## Model Validation and Evaluation Report:

| Model | Training Report | Accuracy | Metrix |
|---|---|---|---|
| Linear Regression |  | 0.11 |  |
| Lasso Regression |  | 0.14 |  |
| SVM |  | 0.41 |  |
| Decision Tree |  | 0.98 |  |

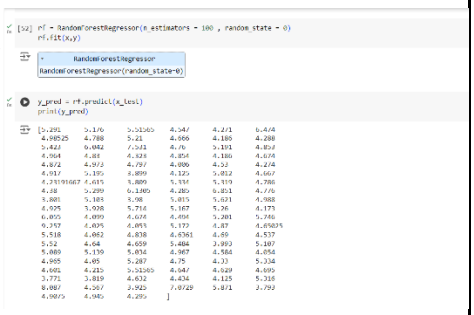
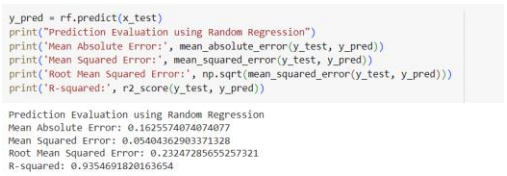
```
[58] print("Prediction Evaluation using Linear Regression")
     print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
     print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
     print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
     print('R-squared:', r2_score(y_test, y_pred))

Prediction Evaluation using Linear Regression
Mean Absolute Error: 0.6635761182069623
Mean Squared Error: 0.742453260904708
Root Mean Squared Error: 0.8616572757800562
R-squared: 0.1134733714697449
```

```
y_pred = lassoReg.predict(x_test)
print("Prediction Evaluation using lasso Regression")
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('R-squared:', r2_score(y_test, y_pred))

Prediction Evaluation using lasso Regression
Mean Absolute Error: 0.6296444264267669
Mean Squared Error: 0.7155358198781405
Root Mean Squared Error: 0.8458935038633058
R-squared: 0.1456141532515728
```

```
[43] #SVM MODEL

[44] svr = SVR().fit(x,y)

[45] y_pred = svr.predict(x_test)

[46] accuracy = svr.score(x_test,y_test)
     print(accuracy)

0.4176454053391483
```

```
y_pred = svr.predict(x_test)
print("Prediction Evaluation using svr Regression")
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('R-squared:', r2_score(y_test, y_pred))

Prediction Evaluation using svr Regression
Mean Absolute Error: 0.496331965955204 46
Mean Squared Error: 0.4877135710244 8615
Root Mean Squared Error: 0.6983649268287 22
R-squared: 0.4176454053391483
```

```
y_pred = dt.predict(x_test)
print("Prediction Evaluation using decisiontree Regression")
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('R-squared:', r2_score(y_test, y_pred))

Prediction Evaluation using decisiontree Regression
Mean Absolute Error: 0.016666666666666 66
Mean Squared Error: 0.011346153846153837
Root Mean Squared Error: 0.10651832633943249
R-squared: 0.9864521202267205
```

| | | | |
|---|---|---|---|
| Random forest |  | 0.93 |  |