

Project Development Phase
Model Performance Test

Date	19 November 2022
Team ID	PNT2022TMID02351
Project Name	Machine Learning Based VehiclePerformance Analyzer
Maximum Marks	10 Marks

PERFORMANCE METRICS

S. No.	PARAMETER	VALUES	SCREENSHOT
1.	Metrics	Regression Model: MAE-,MSE-,RMSE-,R2 score- Classification Model: Confusion Matrix, Accuray Score- & Classification Report-	<p>Decision tree regression</p> <p>Required</p> <p>R squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.</p> <p>R-squared = Explained variation / Total variation</p> <p>Mean Squared Error (MSE)</p> <p>The Mean Squared Error measures the average of the squares of errors; that is, the difference between actual value (y) and the estimated value (ŷ).</p> <pre>In [45]: from sklearn.metrics import r2_score, mean_squared_error In [46]: r2_score(y_test, y_pred) Out[46]: 0.8578094522389582 In [47]: mean_squared_error(y_test, y_pred) Out[47]: 0.142195475994183 In [48]: np.sqrt(mean_squared_error(y_test, y_pred)) Out[48]: 0.377081619498938</pre> <p>Random Forest Regressor</p> <pre>In [49]: from sklearn.ensemble import RandomForestRegressor In [50]: x11 = dataset.iloc[:,1:8].values y11 = dataset.iloc[:,8].values In [51]: from sklearn.model_selection import train_test_split x_train, x_test, y_train, y_test = train_test_split(x11,y11,test_size=0.2,random_state=0) In [52]: rf= RandomForestRegressor(n_estimators=30,random_state=0) rf.fit(x_train,y_train) Out[52]: RandomForestRegressor(n_estimators=30, random_state=0) In [53]: y1_pred=rf.predict(x_test1) y1_pred Out[53]: array([[14.3 , 24.34333333, 14.18333333, 20.26666667, 18.43333333, 20.11666667, 34.96 , 11.3 , 15.36666667, 26.12333333, 36.81333333, 16.5 , 18.95666667, 27.21333333, 16.47666667, 22.64333333, 27.89333333, 27.17 , 16.06666667, 34.64333333, 15.88333333, 13.3 , 23.48333333, 28.71666667, 32.12 , 27.23333333, 34.48666667, 30.88 , 31.76333333, 15.89333333, 19.87666667, 33.32333333, 18.55 , 32.66 , 28.35666667, 24.2 , 18.92 , 16.48666667, 35.24 , 12.3 , 13.4 , 15.4 , 27.18666667, 32.61333333, 29.86666667, 22.1 , 19.83 , 34.8 , 22.11333333, 29.86666667, 34.84 , 25.36666667, 16.34 , 27.4 , 15.4 , 12.36888887, 18.58888887, 25.12888887, 31.78333333, 16.24 , 18.87 , 25.77666667, 18.86666667, 21.53333333, 15.36666667, 15.11666667, 13.46666667, 17.28333333, 24.85666667, 14. , 35.61333333, 13.3 , 23.81333333, 18.2 , 23.98333333, 29.51666667, 17.1 , 38.97 , 29.67666667, 14.35]]) In [54]: from sklearn.metrics import r2_score accuracy = r2_score(y_test1, y1_pred) accuracy Out[54]: 0.899792555413947</pre> <p>Mean Squared error</p>

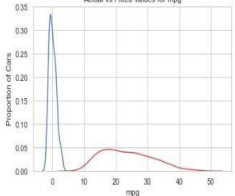
			<pre> In [60]: from sklearn.metrics import r2_score, mean_squared_error In [61]: r2_score(y_test, y_pred2) Out[61]: -0.04347826086956519 In [62]: mean_squared_error(y_test, y_pred2) Out[62]: 0.6 In [63]: np.sqrt(mean_squared_error(y_test, y_pred2)) Out[63]: 0.7745966692414834 </pre> <p>Linear regression</p> <pre> In [44]: sns.distplot(dataset['mpg'], hist=False, color='r', label='Actual Value') sns.distplot(y_pred, hist=False, color='b', label='Fitted Values', ax=ax1) plt.title('Actual vs Fitted Values for mpg') plt.xlabel('mpg') plt.ylabel('Proportion of Cars') plt.show() plt.close() </pre> <p> <small>C:\Users\Hav\Anaconda3\lib\site-packages\seaborn\distributions.py:2618: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'kdeplot' (an axes-level function for kernel density plots).</small> <small>warnings.warn(msg, FutureWarning)</small> <small>C:\Users\Hav\Anaconda3\lib\site-packages\seaborn\distributions.py:2618: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'kdeplot' (an axes-level function for kernel density plots).</small> <small>warnings.warn(msg, FutureWarning)</small> </p>  <p> <small>We can see that the fitted values are reasonably close to the actual values, since the two distributions overlap a bit. However, there is definitely some room for improvement.</small> </p>
	Accuracy	Training accuracy- 0.8999792555413947	<pre> In [54]: from sklearn.metrics import r2_score accuracy = r2_score(y_test1, y1_pred) accuracy Out[54]: 0.8999792555413947 </pre>

Figure 9.1 – Performance Metrics