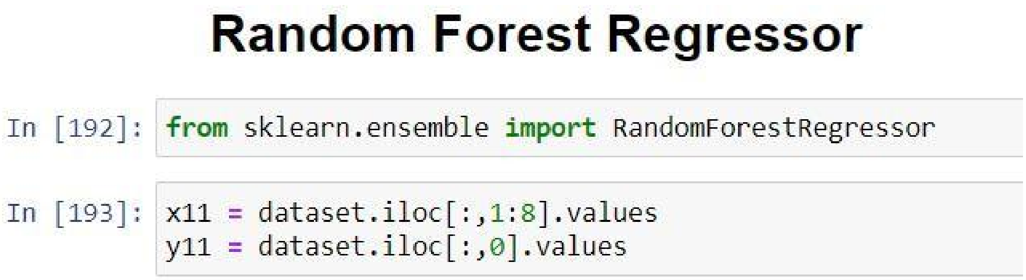
Regression model

Team id - PNT2022TMID37180
Project name - Machine Learning based Vehicle Performance Analyzer

1. Build The Model With The Random Forest Regressor.

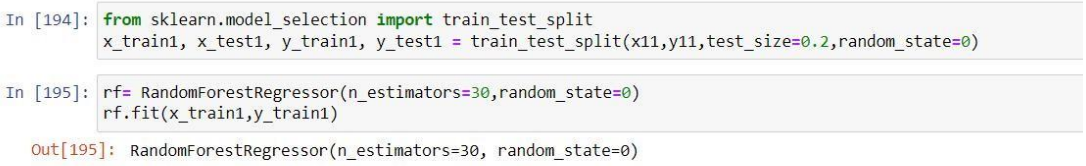


Random Forest Regressor

```
In [192]: from sklearn.ensemble import RandomForestRegressor
```

```
In [193]: x11 = dataset.iloc[:,1:8].values
          y11 = dataset.iloc[:,0].values
```

We use x_train1 and y_train1 obtained above in the train_test_split section to train our decision tree regression model. We're using the fit method and passing the parameters as shown below.

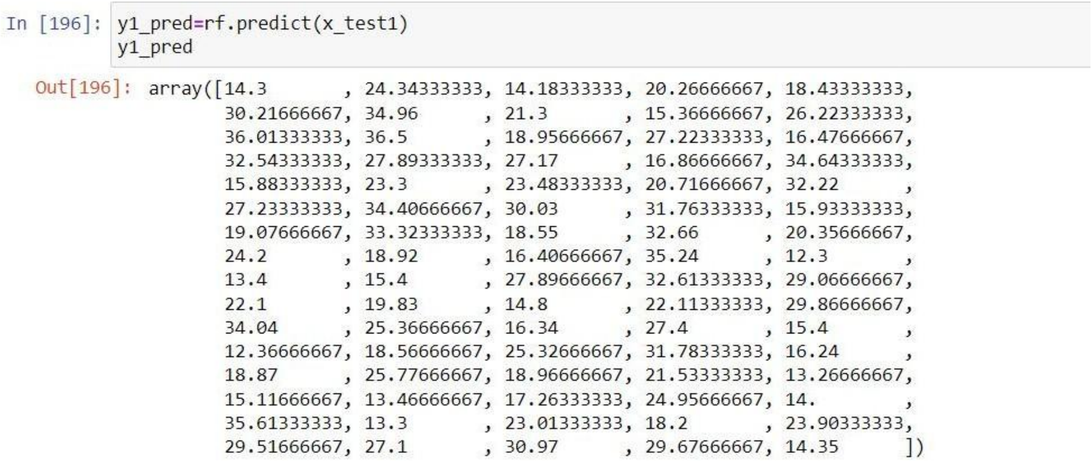


```
In [194]: from sklearn.model_selection import train_test_split
          x_train1, x_test1, y_train1, y_test1 = train_test_split(x11,y11,test_size=0.2,random_state=0)

In [195]: rf= RandomForestRegressor(n_estimators=30,random_state=0)
          rf.fit(x_train1,y_train1)

Out[195]: RandomForestRegressor(n_estimators=30, random_state=0)
```

2. Predict The Values

Once the model is trained, it's ready to make predictions. We can use the predict method on the model and pass x_test1 as a parameter to get the output as y1_pred.

Notice that the prediction output is an array of real numbers corresponding to the input array.

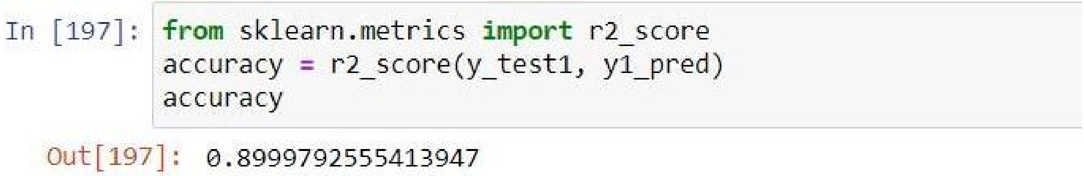


```
In [196]: y1_pred=rf.predict(x_test1)
          y1_pred

Out[196]: array([14.3      , 24.34333333, 14.18333333, 20.26666667, 18.43333333,
                 30.21666667, 34.96      , 21.3      , 15.36666667, 26.22333333,
                 36.01333333, 36.5      , 18.95666667, 27.22333333, 16.47666667,
                 32.54333333, 27.89333333, 27.17      , 16.86666667, 34.64333333,
                 15.88333333, 23.3      , 23.48333333, 20.71666667, 32.22      ,
                 27.23333333, 34.40666667, 30.03      , 31.76333333, 15.93333333,
                 19.07666667, 33.32333333, 18.55      , 32.66      , 20.35666667,
                 24.2      , 18.92      , 16.40666667, 35.24      , 12.3      ,
                 13.4      , 15.4      , 27.89666667, 32.61333333, 29.06666667,
                 22.1      , 19.83      , 14.8      , 22.11333333, 29.86666667,
                 34.04      , 25.36666667, 16.34      , 27.4      , 15.4      ,
                 12.36666667, 18.56666667, 25.32666667, 31.78333333, 16.24      ,
                 18.87      , 25.77666667, 18.96666667, 21.53333333, 13.26666667,
                 15.11666667, 13.46666667, 17.26333333, 24.95666667, 14.       ,
                 35.61333333, 13.3      , 23.01333333, 18.2      , 23.90333333,
                 29.51666667, 27.1      , 30.97      , 29.67666667, 14.35      ])
```

3. Accuracy

For that we need to import the r2_score method from sklearn.metrics package. We can use the r2_score method on the model and pass y_test1 and y1_pred as a parameter to get the accuracy.

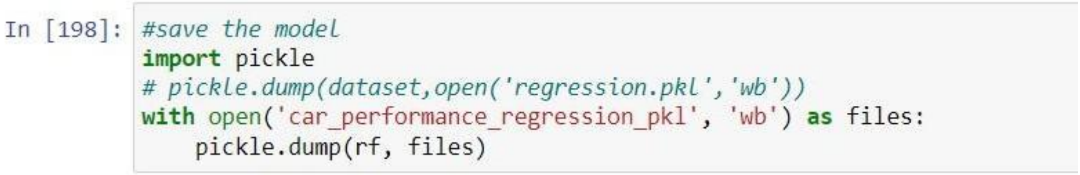
In regression models, R2 corresponds to the squared correlation between the observed outcome values and the predicted values by the model. The higher the R-squared, the better the model.



```
In [197]: from sklearn.metrics import r2_score
          accuracy = r2_score(y_test1, y1_pred)
          accuracy

Out[197]: 0.8999792555413947
```

4. Save the Regression Model

Save the model by importing pickle file.



```
In [198]: #save the model
          import pickle
          # pickle.dump(dataset,open('regression.pkl','wb'))
          with open('car_performance_regression_pkl', 'wb') as files:
              pickle.dump(rf, files)
```