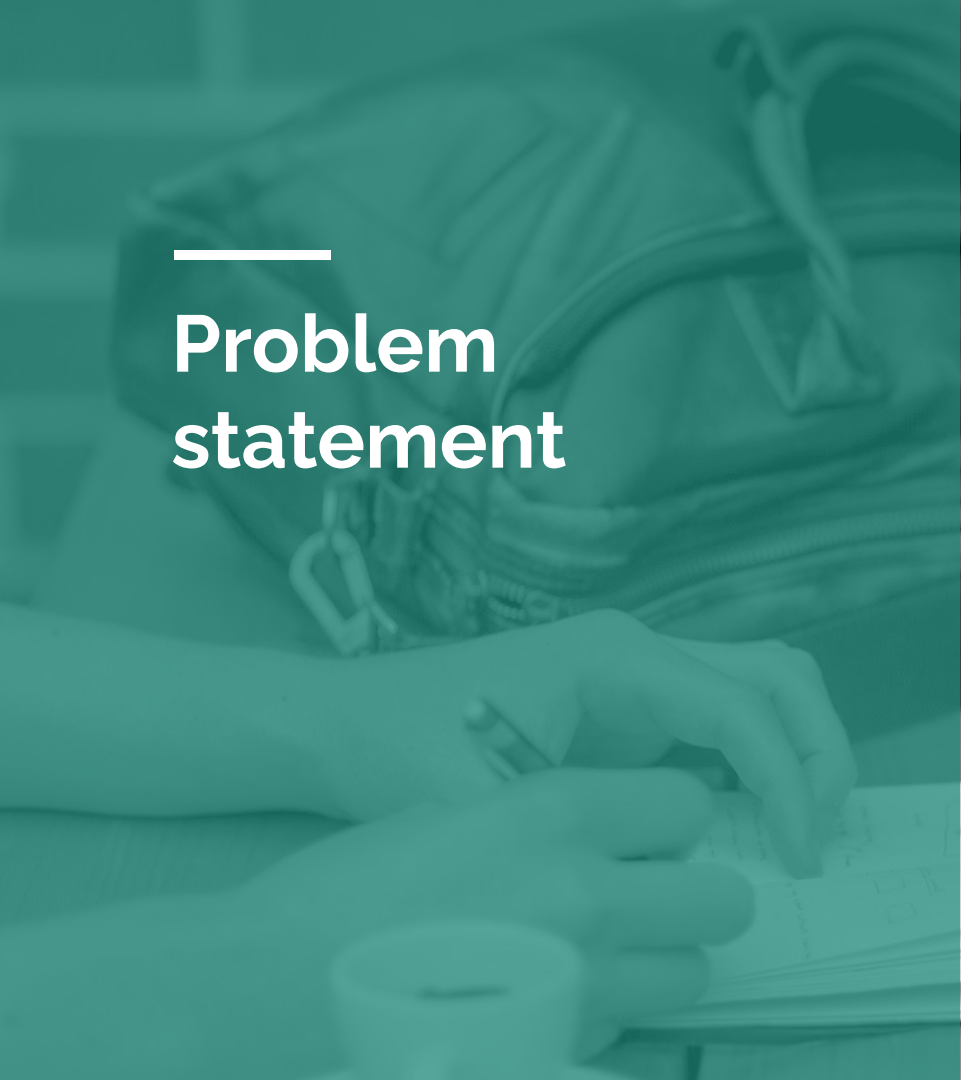
Data Science WorkFlow

Problem statement, Analysis, Approach
and Results.

Outline

The Problem



Problem statement

Using Auto MPG dataset to predict the fuel efficiency with multiple regression models.

The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes.

Prediction Target: MPG

Description of DataSet

Attribute Information:

- mpg: continuous
- cylinders: multi-valued discrete
- displacement: continuous
- horsepower: continuous
- weight: continuous
- acceleration: continuous
- model year: multi-valued discrete
- origin: multi-valued discrete

A relatively small dataset, with 398 examples.

Exploratory Analysis

Exploratory Analysis

01

Analyzing the origin column tells us that it is a categorical column.

Value Count:

We can then use one-hot encoder to encode the categorical values to model acceptable binary values.

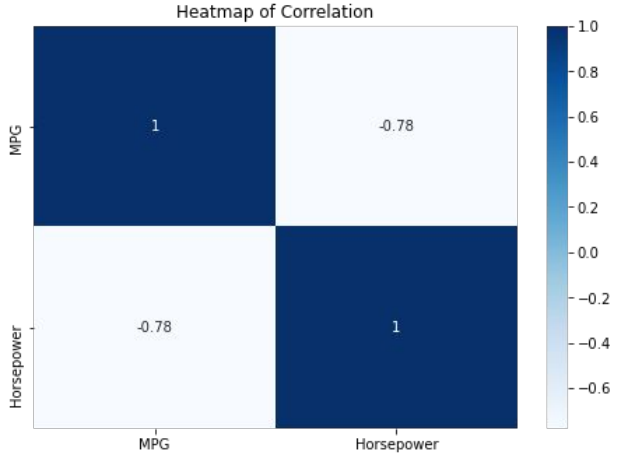
Exploratory Analysis

02

Analyzing the other columns we can see that Horsepower is the only column having null values which needs to be addressed.

Correlation:

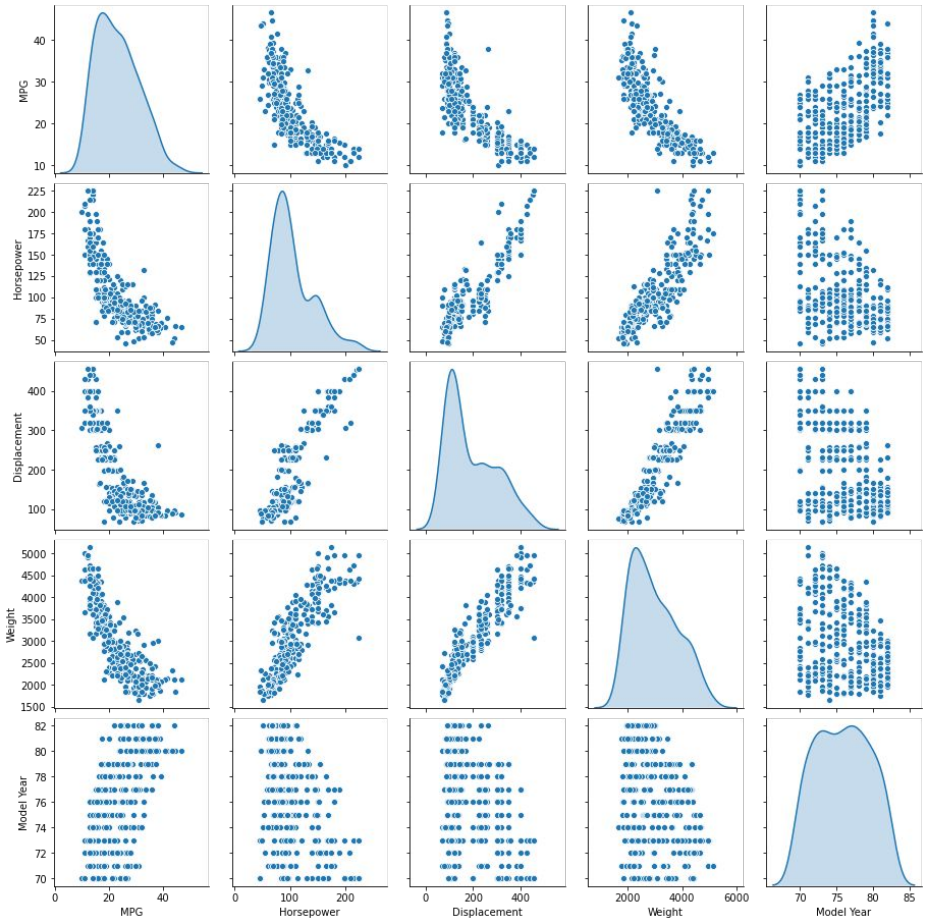
Input Feature Horsepower has a high correlation, and hence we'll drop the rows from the dataset instead of imputing.



Exploratory Analysis

03

From a correlation plot, we are able to find out the correlation between the input features and the ones that'll influence our model more.



Approach

Description

Finally, after data has been preprocessed, we've normalized the data to 1NF (0-mean; 1-standard deviation) using tensorflow **normalizer** with **adapt** layer.

Now, we'll be creating 4 different models to predict the MPG based on our training data (80%), and validating it against our testing data (20%).

Predicting 1 output for each example, on the line y = mx + c with m as the matrix of features

We're using the Tensorflow-Keras API and sklearn module from python.

- Linear Regression
- Random Forest Regression
- Decision Tree
- Deep Neural Network

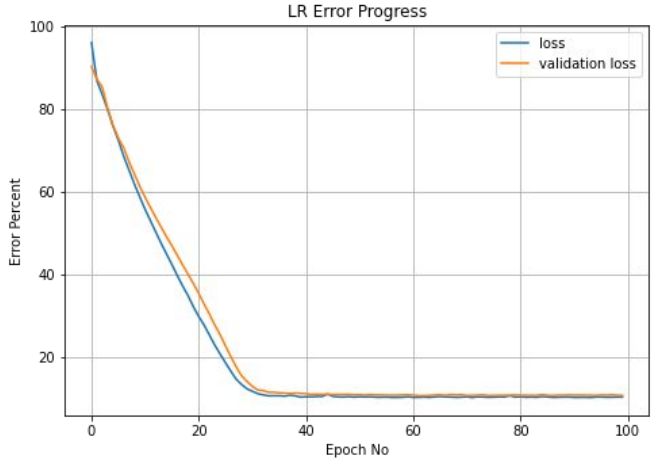
Approach

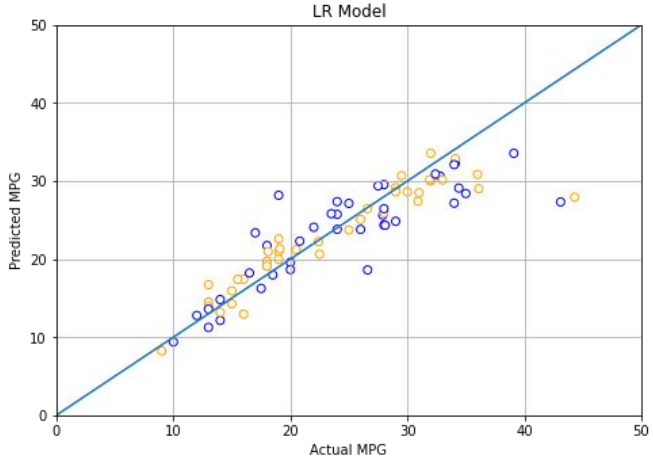
01

We'll use the **Keras Linear Regression** model first, with 1 output layer and a total of 29 parameters and a learning rate of 0.1.

We'll be calculating the error as Mean Absolute Percentage Error

The model gives us an 89% accuracy with a steady decline in loss, finally settling at 10.64%



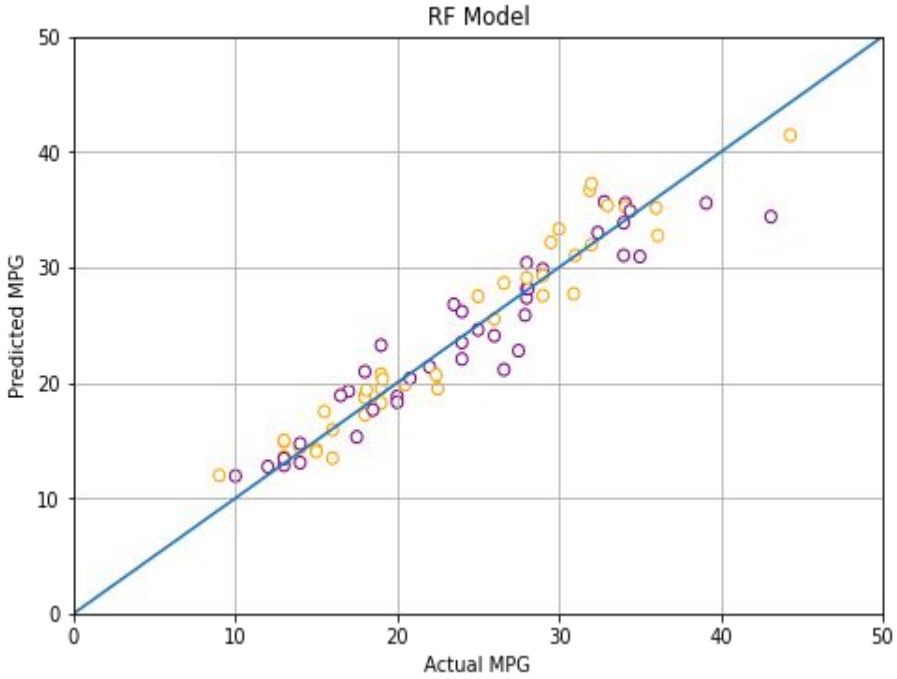


Approach
02

We'll use the sklearn's **Random Forest Regressor**, random_state as 1.

We'll be calculating the error as Mean Absolute Percentage Error.

The model gives us a 92% accuracy with a final loss, settling at 7.82%. The Random Forest regressor shows significant improvement over the Linear Regressor.

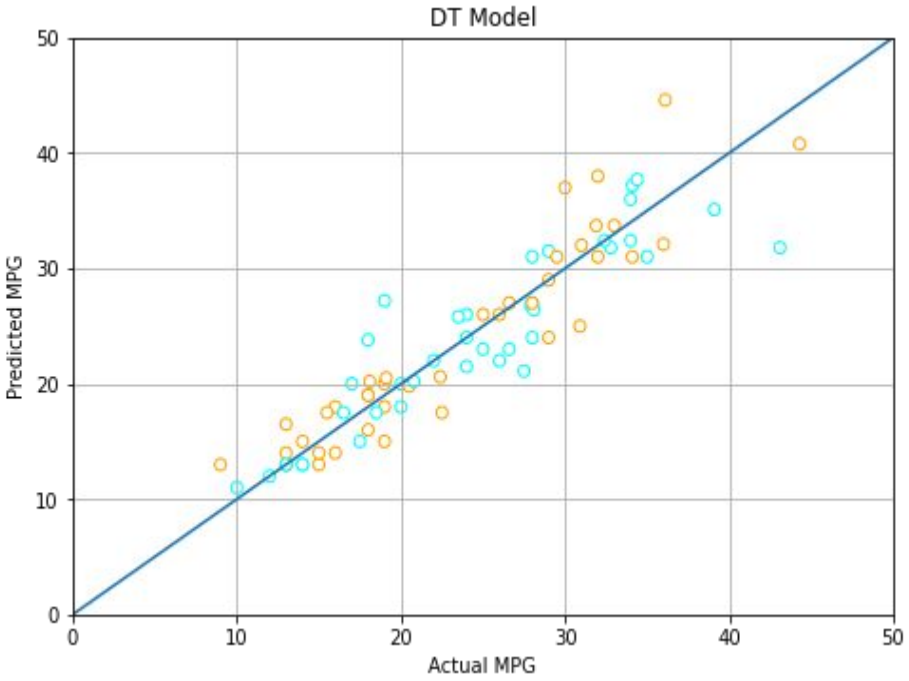


Approach
03

The **Decision Tree** gives a marginally better output than the Linear Regressor.

We'll be calculating the error as Mean Absolute Percentage Error.

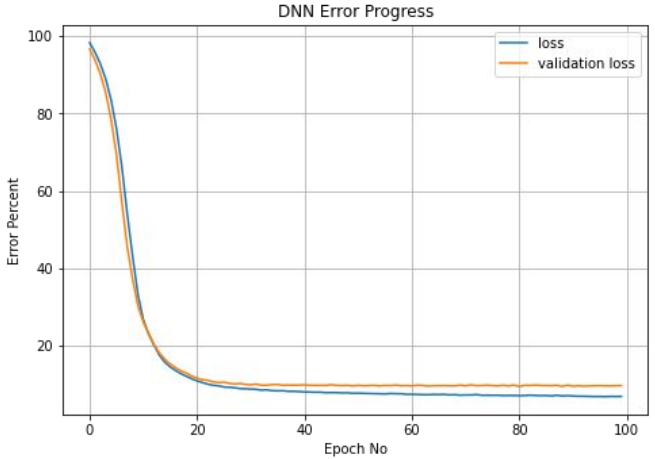
The model gives us a 90% accuracy with a final loss, settling at 10%. However, the Random Forest approach prove to be significantly better.

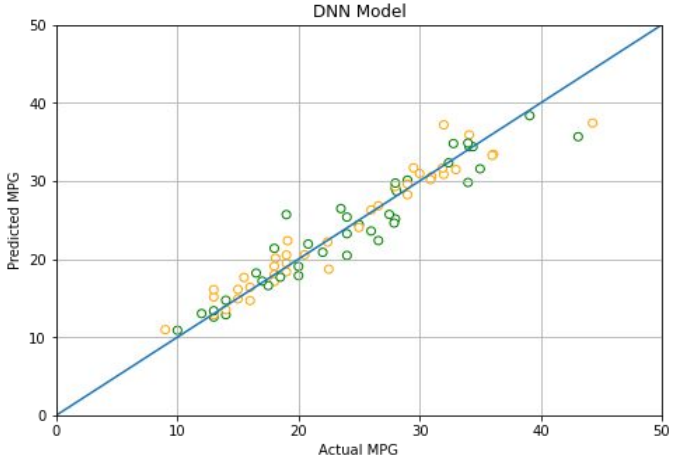


Approach
04

Finally we'll be creating a **Deep Neural Network**, again with Tensorflow's Keras API.
Here we're using a learning rate of 0.001 with two hidden non linear layers with 'relu' non-linearity and 1 output layer.

Calculating the error with Mean Absolute Percentage Error, we get a final loss of 7.14%, with an accuracy of 93% (approx).





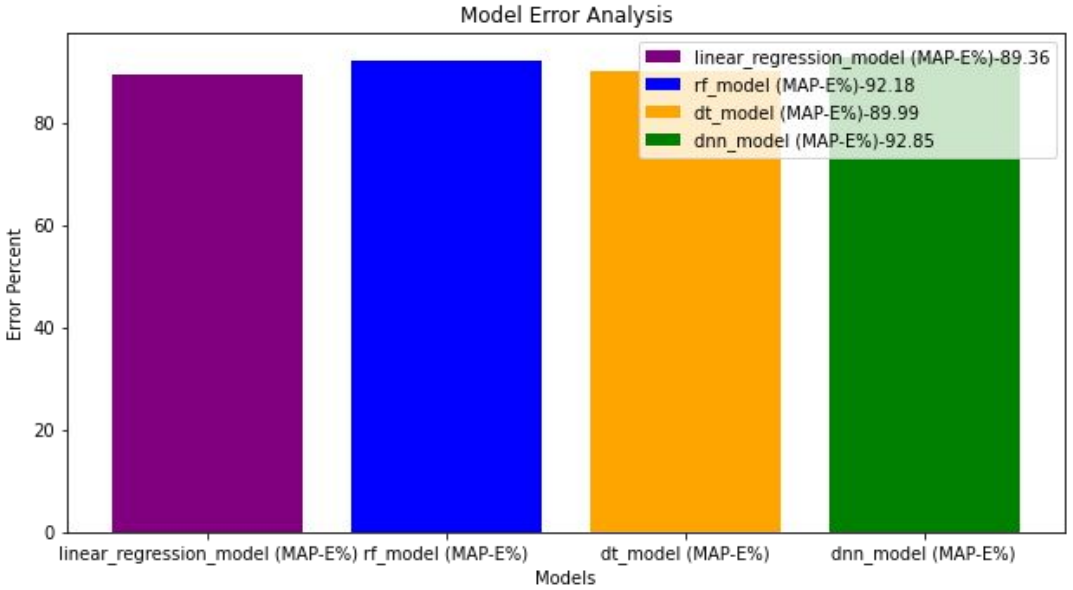
Results

Conclusion

To conclude, all 4 of our models have given us respectable results.

- Linear regression model: 89.36%
- Random forest model: 92.18%
- Decision tree model: 89.99%,
- Deep Neural Network model: 92.85%

However we can see that the DNN model has the highest accuracy whereas the Linear regression model has the lowest.



Thank You.

References

Please check the jupyter notebook: Predict Fuel Efficiency.ipynb

Dataset: http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/