**Index**

1. Abstract
   1. Introduction
   2. Objective of Research
2. Problem Statement
3. Data Collection

3.1.Data pre-processing

4.Model Building

5.Application Building

7. Findings and Suggestions

8. Conclusion

9. Reference

**Abstract**

Predicting the performance level of cars is an important and interesting problem. The main goal of the current study is to predict the performance of the car to improve certain behavior of the vehicle. This can significantly help to improve the systems fuel consumption and increase the efficiency. The performance analysis of the car based on the engine type, no of engine cylinders, fuel type and horsepower etc.These are the factors on which the health of the car can be predicted. It is an on-going process of obtaining, researching, analyzing and recording the health based on the above three factors. The performance objectives like mileage, dependability, flexibility and cost can be grouped together to play a vital role in prediction engine and engine management system. This approach is the very important step towards understanding the vehicles performance.

**Introduction**

Fuel consumption of automobile has become one of the major challenges of the automobile industry and fleet organization. Consider a long-distance vehicle that runs between a major city and a rural area in a fleet. During that, the vehicle may drive into heavy traffic and different terrain conditions such as going over a hill station, off-roads, muddy area etc.,. Besides, the traffic intensity, weather conditions, and the load of the automobile may vary depending on the day of the week. Every driver will have their way of driving the vehicle. Fuel efficiency of vehicles can be affected by careless maintenance and improper driving habits such as time duration of cruise mode, number of breaks applied, acceleration speed by the driver etc. There is no existing reliable machine learning technique for predicting fuel consumption in automobile vehicles effectively. For controlling and maintaining the fuel consumption rate on average, a machine learning technique is introduced in this paper. In this paper, ensembles tacking, a machine learning theory has been developed which merges the machine learning models into one predictive model and gives more accurate results higher than the expectation of the machine learning models alone.

**Objective of Research**

The given data set is about Prediction Of Car mileage. The given data is to analyzed. Various objectives of the given data set are as follows:

1. To study the given data
2. To apply data cleaning methods to remove unknown data from the given data set.
3. Test the designed model’s working
4. Draw conclusions from the developed model
5. Predict whether the mileage of car will be satisfied or not

**Problem Statement**

The basic idea of analysing the Auto mpg dataset is to get a fair idea about the factors affecting the aggregate fuel consumption of each car.

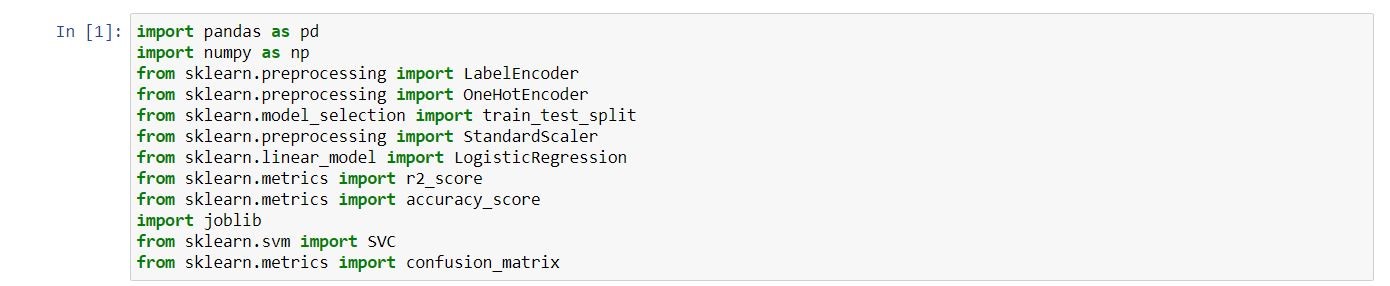
Fuel consumption of car is affected by various factors such as Model year, Horsepower, Number of Cylinders present, displacement, Weight and Acceleration of the car.We need to find which factors mostly affect the Fuel consumption of a car in order to improve the mpg value.Hence build a model to predict the mpg value of each car.

**Data Collection**

The given data set is related to Car mileage. It was taken from the website **kaggle.com**. The website provides various datasets from various domains.

**Data pre-processing**

**Importing required Libraries**:

**Pandas:** It is a python library mainly used for data manipulation.

**NumPy:** This python library is used for numerical analysis.

**Matplotlib and Seaborn:** Both are the data visualization library used for plotting graph which will help us for understanding the data.

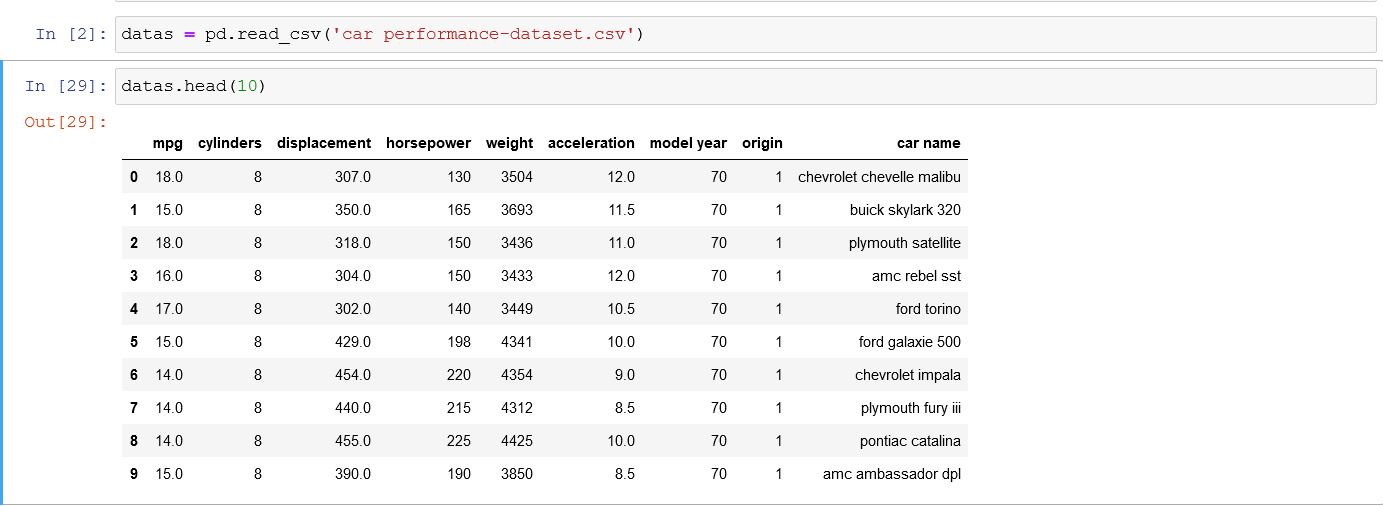
**Accuracy score:** used in classification type problem and for finding accuracy it is used.

**R2 Score:** Coefficient of Determination or R² is another metric used for evaluating the performance of a regression model. The metric helps us to compare our current model with a constant baseline and tells us how much our model is better.

**Train\_test\_split:** used for splitting data arrays into training data and for testing data.

**Scikit-learn (Sklearn)** is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistence interface in Python.

**Importing the dataset:**

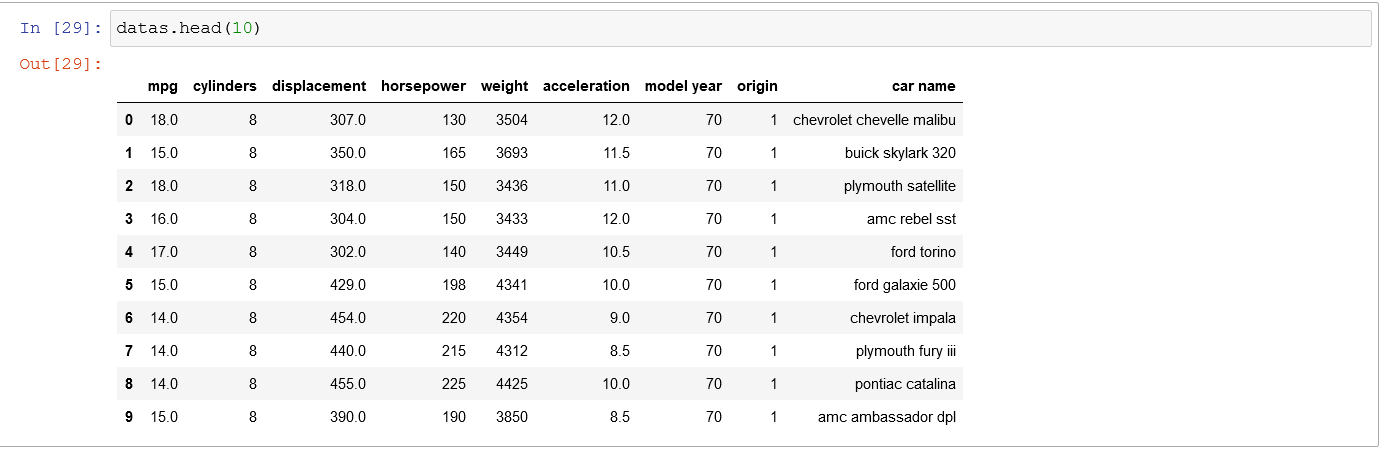


* You might have your data in .csv files, .excel files or .tsv files or something else. But the goal is the same in all cases. If you want to analyse that data using pandas, the first step will be to read it into a data structure that’s compatible with pandas.
* Let’s load a .csv data file into pandas. There is a function for it, called **read\_csv().**We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).
* Path names on Windows tend to have backslashes in them. But we want them to mean actual backslashes, not special characters.

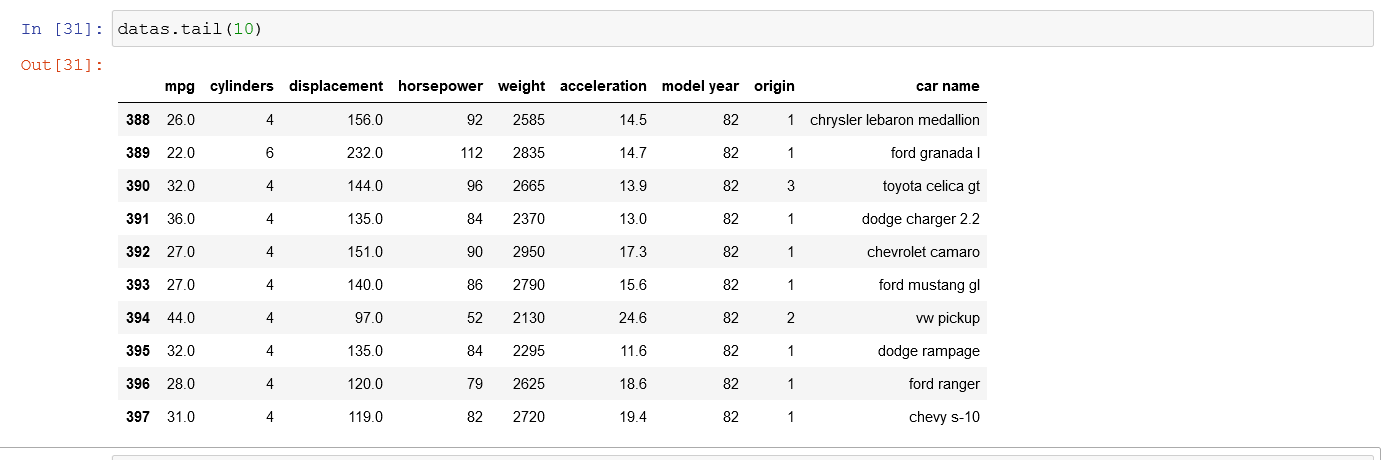
**Data Visualization:**

Exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods and used for determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

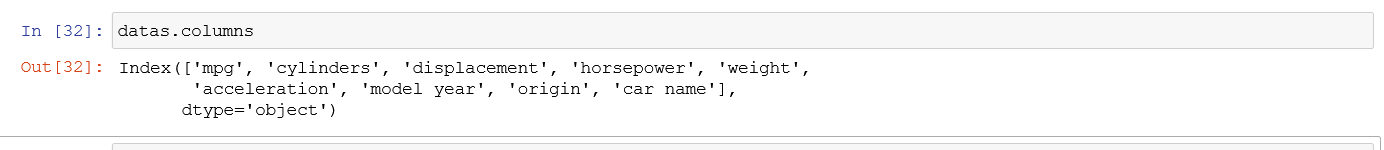
* To check first five rows of dataset, we have a function call **head( ).**



* This head () function returns the first 5 rows for the object based on position, it is useful for quickly testing if your object has the right type of data in it.
* To check last five rows of dataset, we have a function call **tail().**



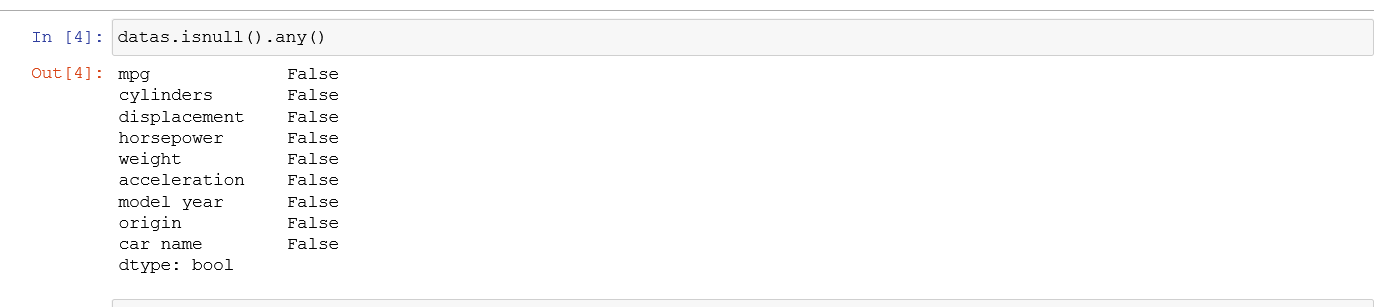
* For finding the names of the columns present in the dataset we make use of **columns**



**Taking care of Missing Data:**

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. Obviously, you could remove the entire line of data but what if you are unknowingly removing crucial information? Of course we would not want to do that. One of the most common ideas to handle the problem is to take a mean of all the values for continuous and for categorical we make use of mode values and replace the missing data.

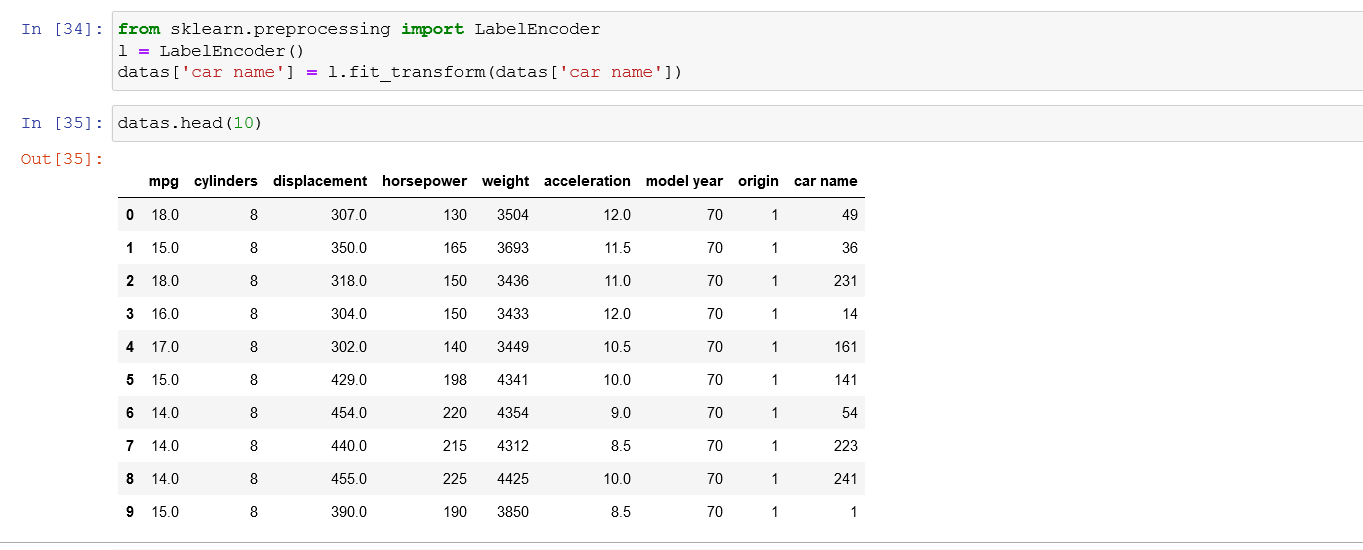
* We will be using **isnull().any()** method to see which column has missing values.



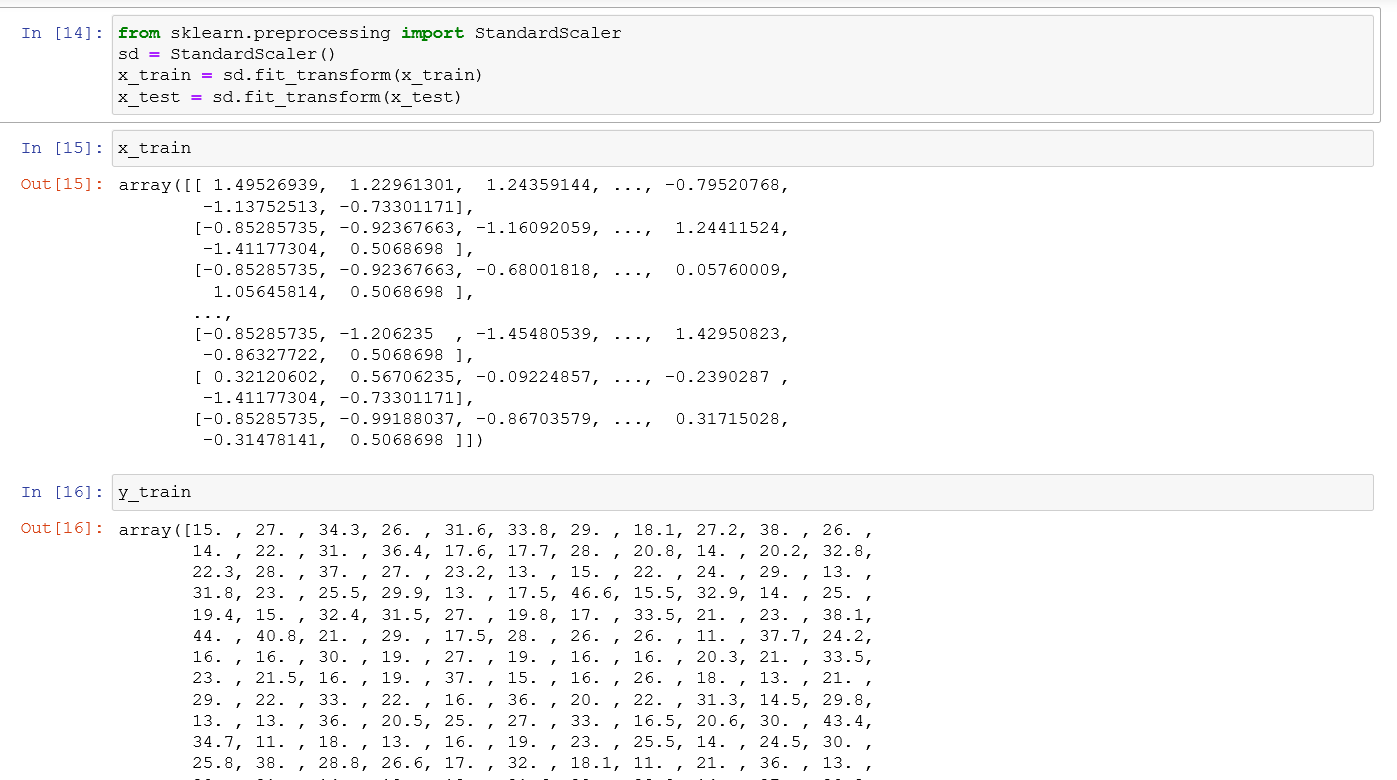
**Label encoding:**

Typically, any structured dataset includes multiple columns with combination of numerical as well as categorical variables. A machine can only understand the numbers. It cannot understand the text. That’s essentially the case with [Machine Learning algorithms](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/?utm_source=blog&utm_medium=one-hot-encoding-vs-label-encoding-using-scikit-learn) too. We need to convert each text category to numbers in order for the machine to process those using mathematical equations.

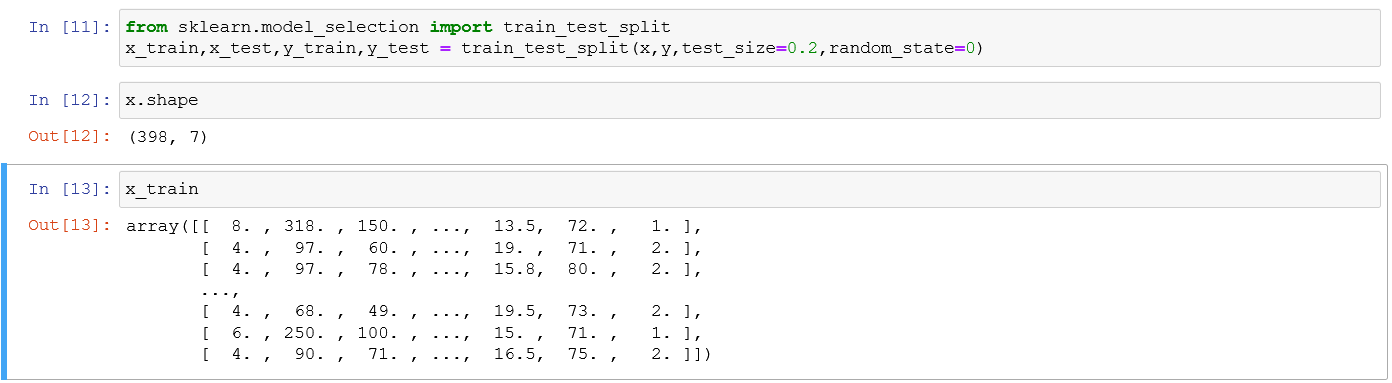
**Label Encoding** is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.



**Feature Scaling:**



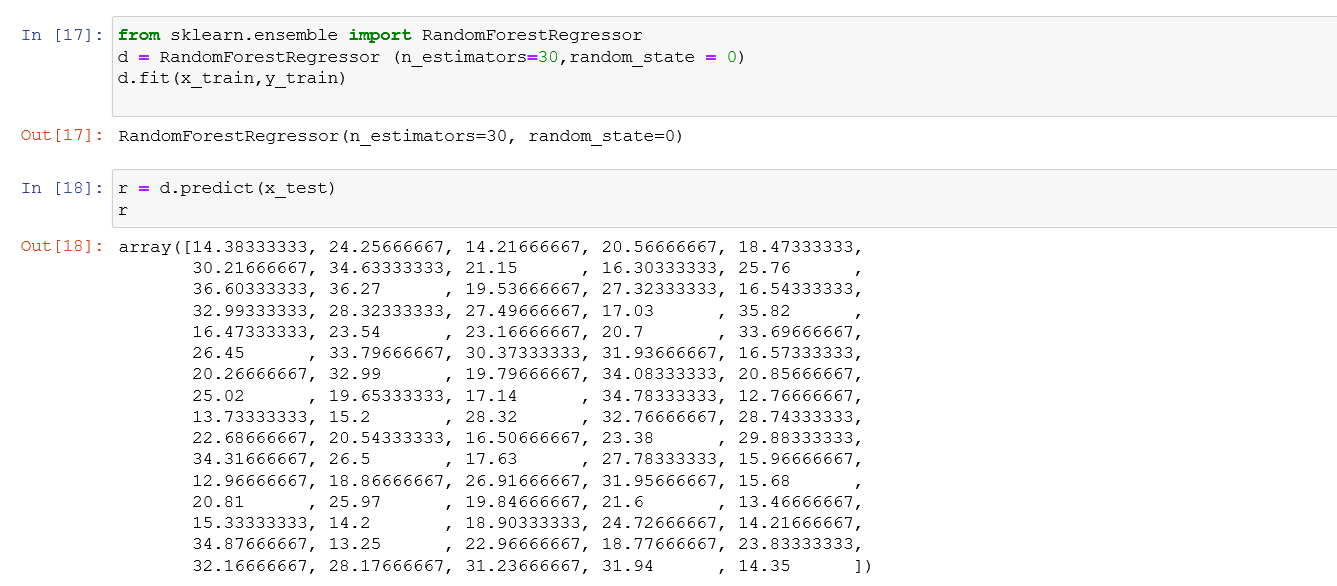
**Splitting Data into Train and Test:**

* When you are working on a model and you want to train it, you obviously have a dataset. But after training, we have to test the model on some test dataset. For this, you will a dataset which is different from the training set you used earlier. But it might not always be possible to have so much data during the development phase. In such cases, the solution is to split the dataset into two sets, one for training and the other for testing.
* Now split our dataset into train set and test using train\_test\_split class from scikit learn library.

**Model Building:**

Training and testing the model:

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values.

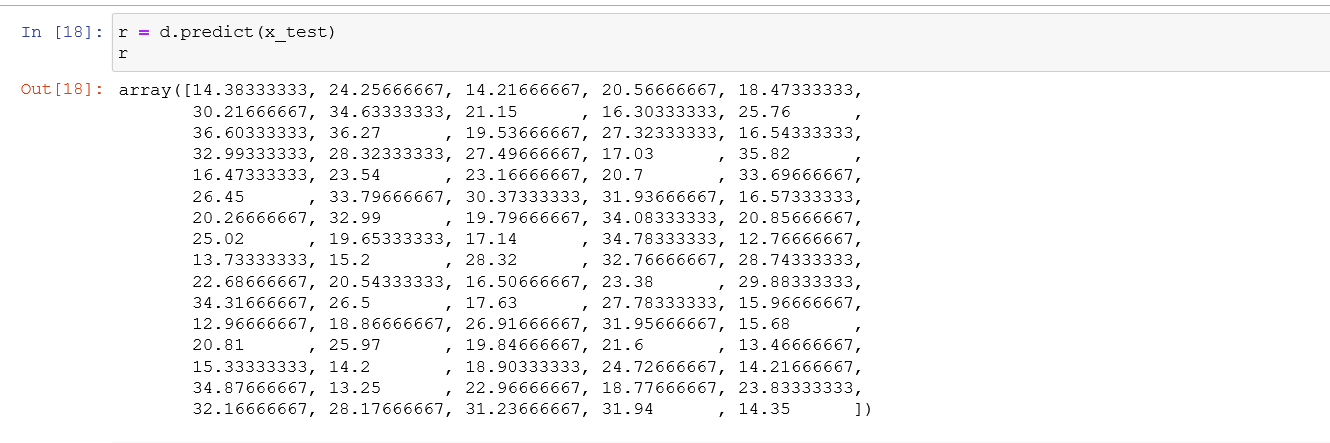
**#random forest regression**

**# SVR Regression**

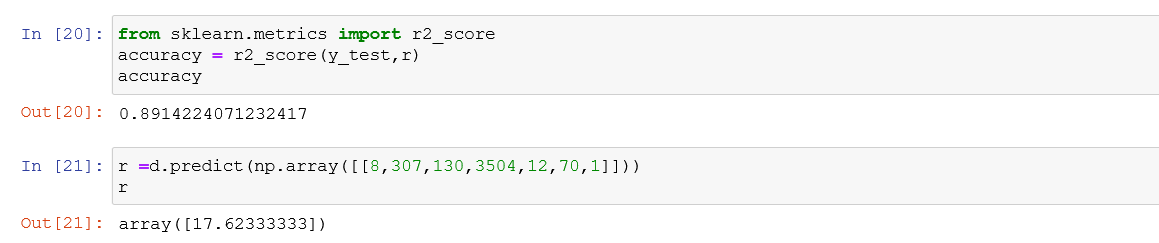
**Predict the values:**

Once the model is trained, it’s ready to make predictions. We can use the **predict** method on the model and pass **x\_test** as a parameter to get the output as **pred.**

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



**Evaluation:**

Finally, we need to check to see how well our model is performing on the test data. There are many evaluation techniques are there. For this, we evaluate **r2\_score** produced by the model

**Saving a model:**

Model is saved so it can be used in future and no need to train it again.

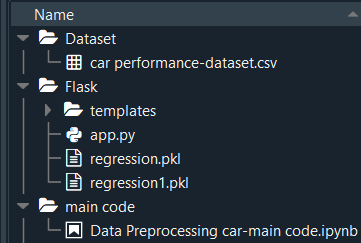
**Application Building:**

Creating a HTML File, flask application.

* Build python code
* Importing Libraries
* Routing to the html Page
* Showcasing prediction on UI
* Run The app in local browser.

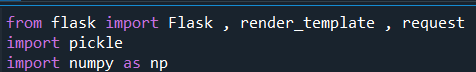
**Project Structure:**

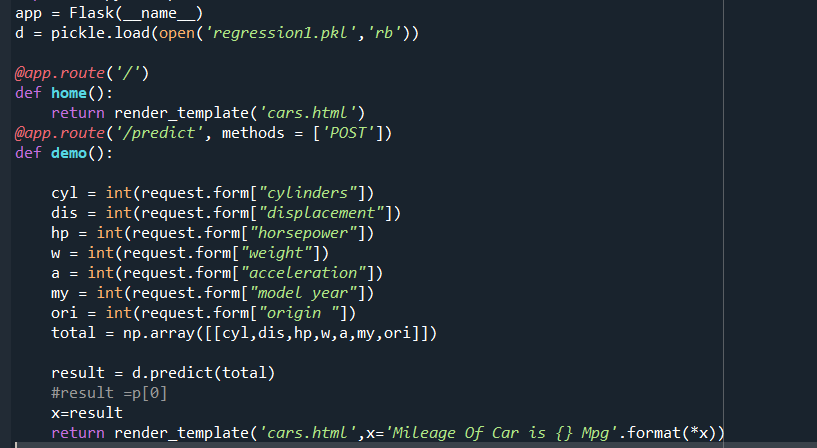
Create a Project folder that contains files as shown below



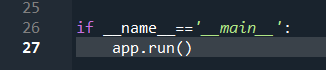
* We are building a Flask Application that needs HTML pages stored in the templates folder
* Templates folder contains index.html
* Static folder contains CSS and image files.

**Task 1: Importing Libraries**

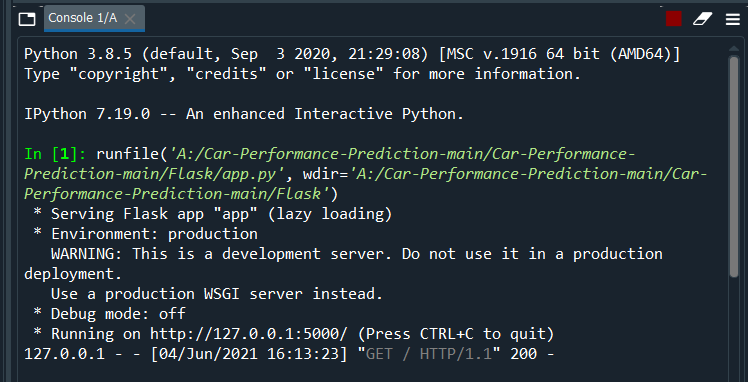


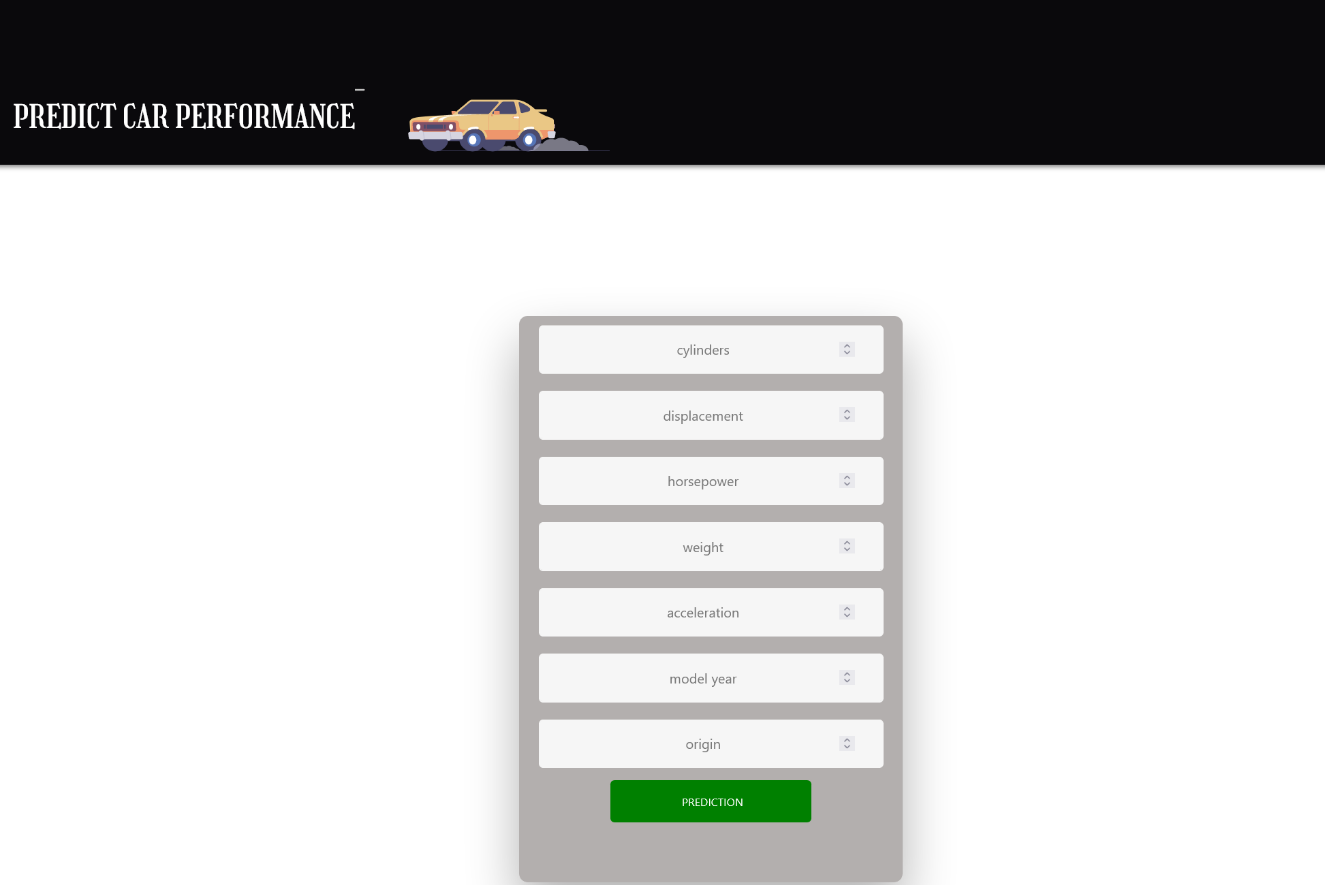
**Task 2: Routing to the html Page** :

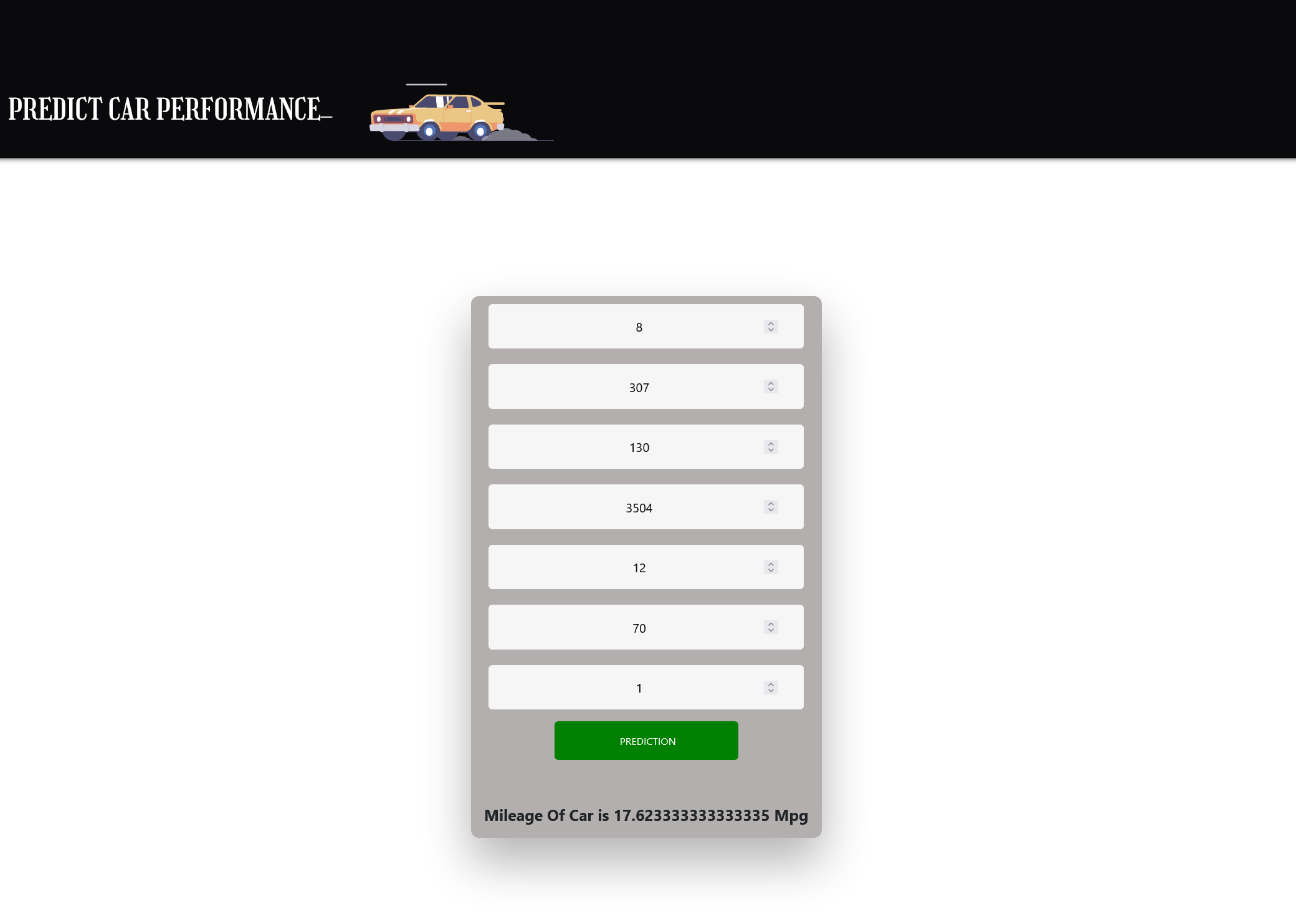
**Task 3: Main Function**



**Activity 3: Run the application**

* Open the anaconda prompt from the start menu.
* Navigate to the folder where your app.py resides.
* Now type “python app.py” command.
* It will show the local host where your app is running on [**http://127.0.0.1:5000/**](http://127.0.0.1:5000/)
* Copy that local host URL and open that URL in the browser. It does navigate me to where you can view your web page.
* Enter the values, click on the predict button and see the result/prediction on the web page

**Output Screen:**

**s**

**Findings and Suggestions**

Through Exploratory Data Analysis,

* The Accuracy for single randomforest regression is 89.1%. It had a R square value for training   is 0.891
* The Accuracy for svr regression is 82.0%. It had a R square value for testing is 0.82
* we have predicted the mpg value by ensemble technique and compared the actual mpg with predicted mpg value.

**Conclusion**

This paper presented a machine learning model developing a stack of regressors predict the automobile vehicles in a fleet. The model relies on eight predictors: mpg, cylinders, horsepower, weight, acceleration, car name, model year, origin. An experiment was completed using ensemble stacking of the regressors and thereby proving that the combination of models will give high accuracy in the prediction

**Reference**

* 1. www.kaggle.com
* 2. www.quora.com
* 3. www.wikkipedia.com