# PREDICTING CAR PRICE USING MACHINE LEARNING

****

**Submitted By :**

Abhay Das – AM.EN.U4CSE20301

Devesh Kumar- AM.EN.U4CSE20321

Sreesankar S- AM.EN.U4CSE20366

Vaishakh S Krishnan- AM.EN.U4CSE20372

Vignesh Nair: AM.EN.U4CSE20374

**ABSTRACT**

*Informal Description(Ill Posed Problem)*

We have to predict the car price based upon the kilometres run, years of service, the number of previous owners, brand, fuel type, transmission type.

*Formal Description*

We are making the use of data available with price of the cars in the present market keeping all the important factors in mind and calculating an estimated price in the present date without asking help from a third party. Our model will calculate the estimated price will depend on the various factors and current news.

We will make a model with the right amount of test data so that it wouldn’t depend on some bad reviews or results made by few customers.

**INTRODUCTION**

**Motivation:**

Cars are something which mean so much to a person, being a car enthusiast myself. I thought if I were to buy a new car it would cost a lot from a showroom. On the other hand, we all know that a car loses at least 15% of its value the moment it is out of a showroom.

Realistically speaking second hand cars might look like a better deal in the long run if we can manage to find a great car for the value it is bought. It might seem easy for me to predict a car’s price as I am familiar with the car prices, the demand, liking and the safety reasons a car might have but to a normal person it might all confuse him and he would instead get a new one or a person who wants to buy a second vehicle who doesn’t have any prior knowledge of markets and how they should be priced will be taken advantage of.

So, we provide a model based on various factors so that a customer can enter all the details of the car and the model would predict an estimated value of the car.

**Benefits of Solution:**

If benefits us personally, the proposed framework for our model includes data cleaning, data pre-processing, applying classifiers on the data and finally comparing the results from the different classification models we used. So, by the end of this project if we have done it Successfully, we can say that we have got idea of them.

**Solution Use:**

When we work on these sorts of data, we can make a model which can give us a good prediction on the price of the Car Price based on other variables. We are going to use Linear Regression and some other ML algorithms for this dataset and see if it gives us a good accuracy or not.

**Dataset Finalization**

# Dataset :Vehicle dataset

**Dataset 1**: [Link](https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho?select=car+data.csv)

Dataset 2 : [Link](https://drive.google.com/file/d/1r0T8qlqxmcVtuBH0tiqZ0GIrNirFfnBX/view?usp=share_link)

Dataset 3 : [Link](https://drive.google.com/file/d/1eBBK1BihxTpQgsjTICsTSy9-ZPbHSjPL/view?usp=share_link)

*1.What is the data about?*

The data used in this project was downloaded from Kaggle. It was uploaded on Kaggle by Austin Reese who Kaggle.com user. Austin Reese scraped this data from craigslist with non-profit purpose. It contains most all relevant information that Craigslist provides on car sales including columns like price, condition, manufacturer, latitude/longitude, and 22 other categories.

*2.What are the number of features and describe each of the features and explain the importance?*

**Feature Engineering**

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

Note: Here we are going to extracting the new feature from the existing features for predicting the output. That’s the technique will prove your domain knowledge. Here as well as we are going to do handing the categorical features in the car price dataset.

**DATASET-1**

**Import Necessary Packages**

**Table

Description automatically generated**

**Load Dataset:**

A picture containing table

Description automatically generated

***Inference*:** Here we are loading the dataset and using head() we can print the first 5 rows of the data set.

**Data Preprocessing:**

Text

Description automatically generated with medium confidence

***Inference*:** info we are using to get a summary of the dataframe.

A picture containing table

Description automatically generated

***Inference*:** To check the number of null values present in each column.

**Chart, bar chart

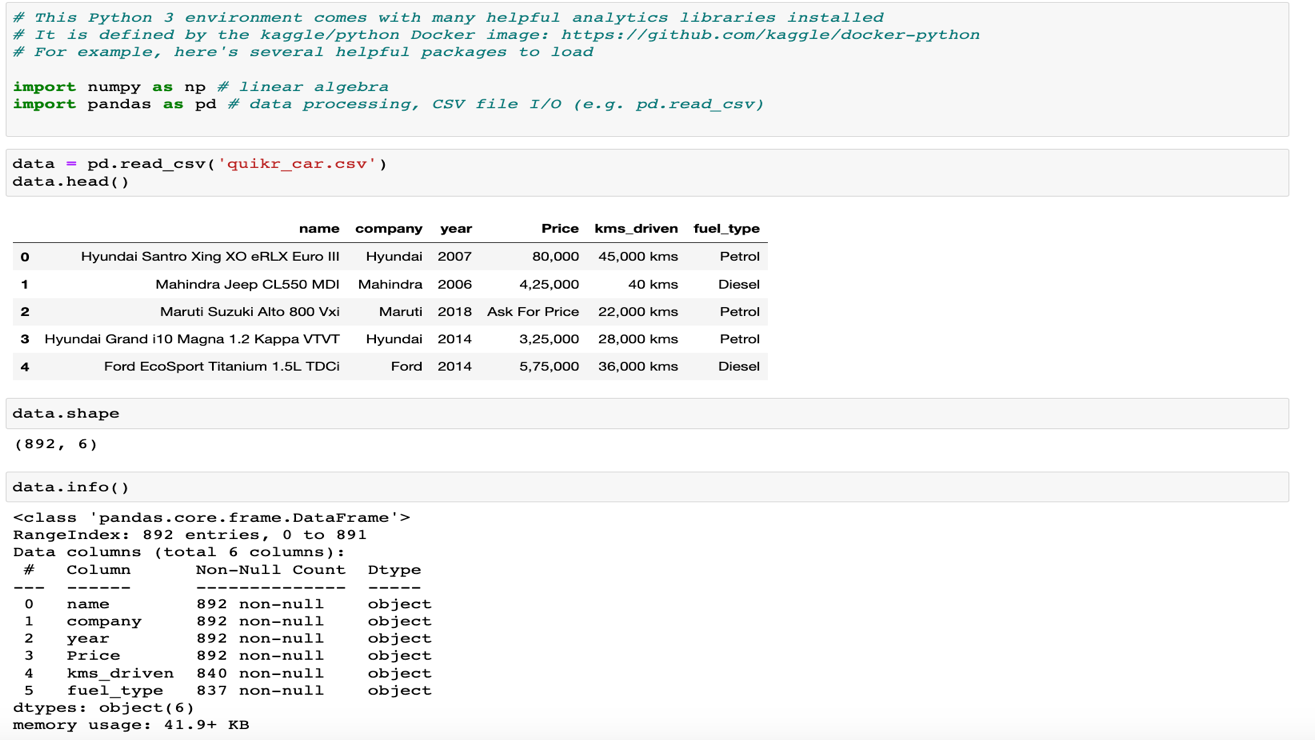
Description automatically generated**

Show the counts of observations in each categorical bin using bars

***Inference*:** Here we have used countplot() to show the counts of observations in each categorical bin using bars.

**DATASET 2**

**Load data set**



Graphical user interface, application

Description automatically generated

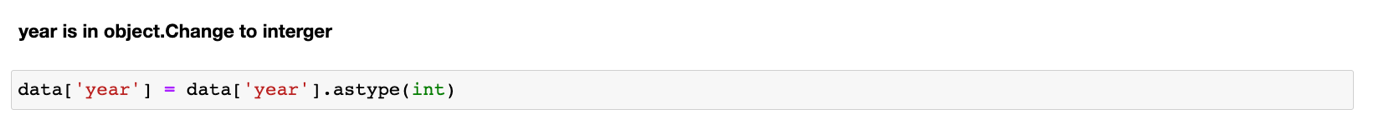
**Data Pre-processing**

Here there are many value in year column that not in year type.

Graphical user interface, table

Description automatically generated

**year is in object. Change to integer**



**Price has Ask for Price**

Graphical user interface, table

Description automatically generated with medium confidence

**Price has commas in its prices and is in object**



**kms\_driven has object values with kms at last.**

****

Graphical user interface

Description automatically generated with medium confidence

**It has nan values and two rows have 'Petrol' in them**

****

**fuel\_type has nan values**



### ***Inference:***

### name and company had spammed data...but with the previous cleaning, those rows got removed.

#### Company does not need any cleaning now. Changing car names. Keeping only the first three words

**Resetting the index of the final cleaned data**

A picture containing text

Description automatically generated

## **Cleaned Data**

Graphical user interface, table

Description automatically generated

**Data Visualisation**

### Checking relationship of Company with Price

Chart

Description automatically generated

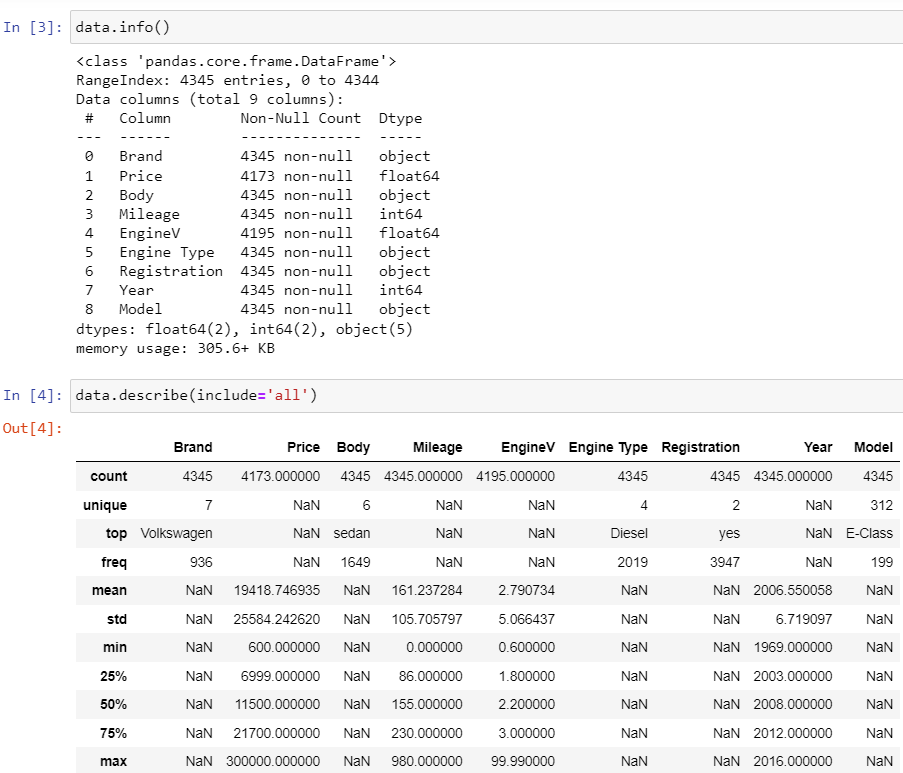
### Relationship of Price with FuelType, Year and Company mixed

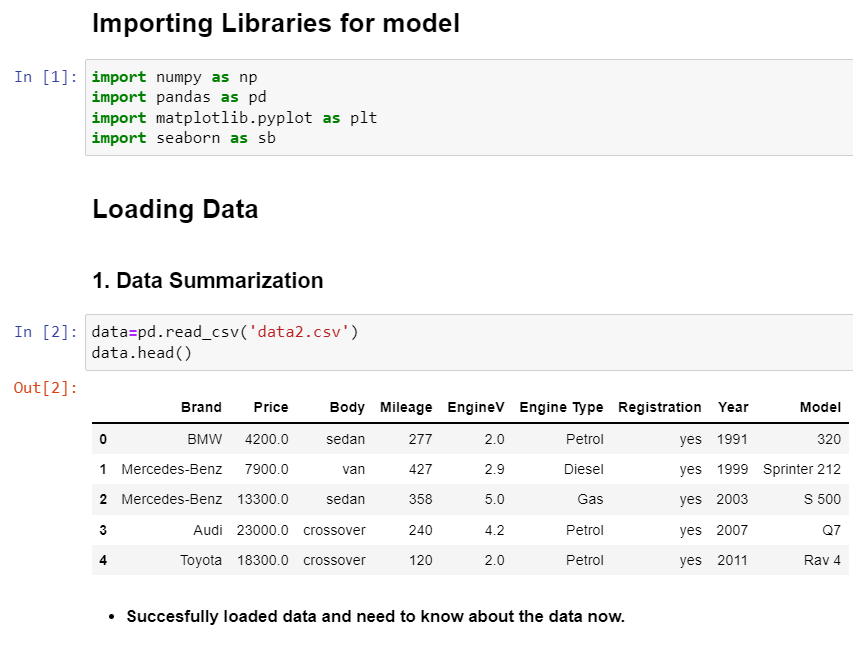
Chart, scatter chart

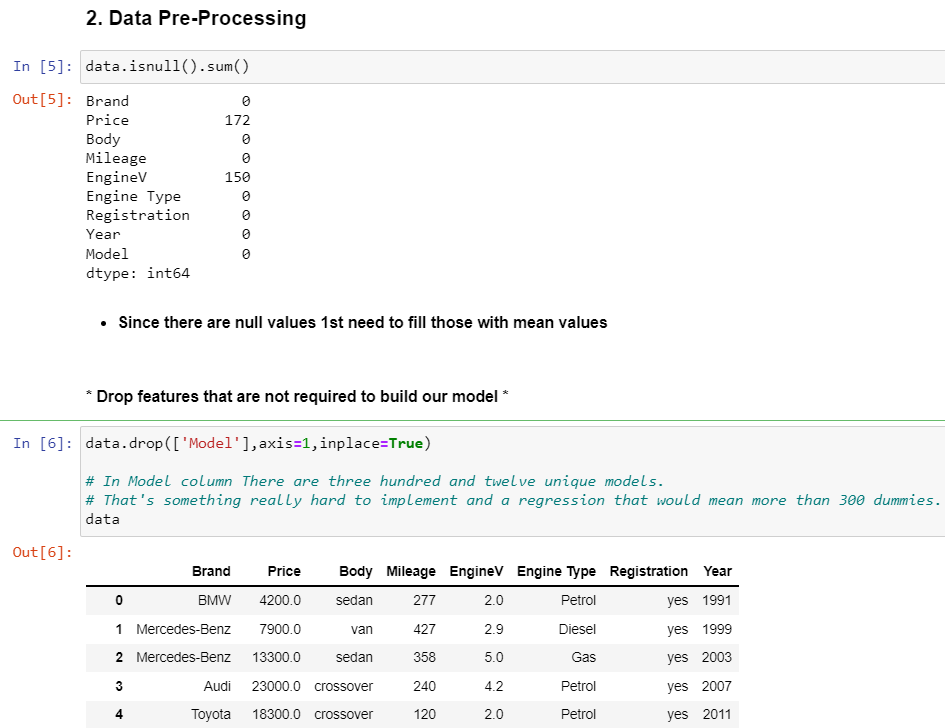
Description automatically generated

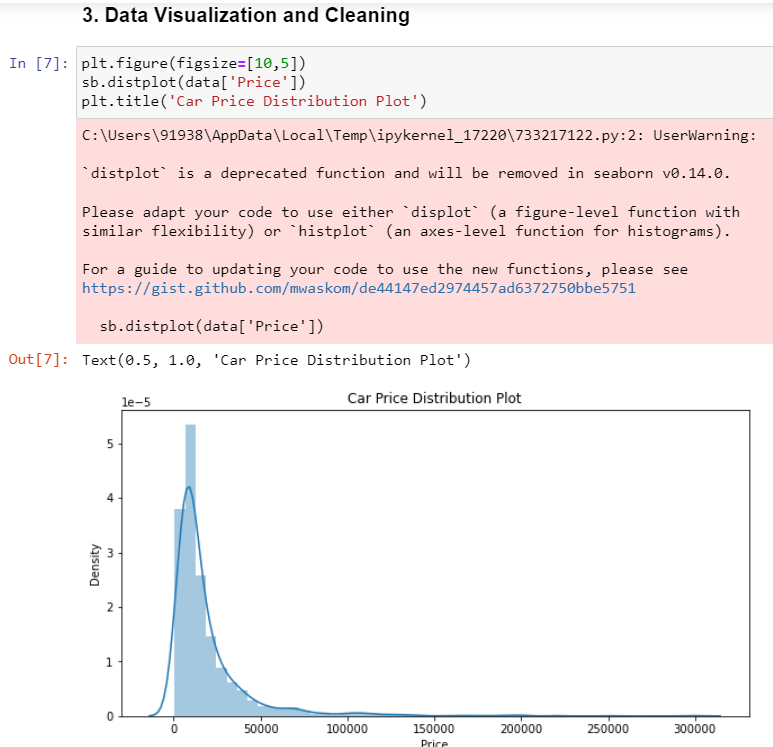
**DATASET 3**

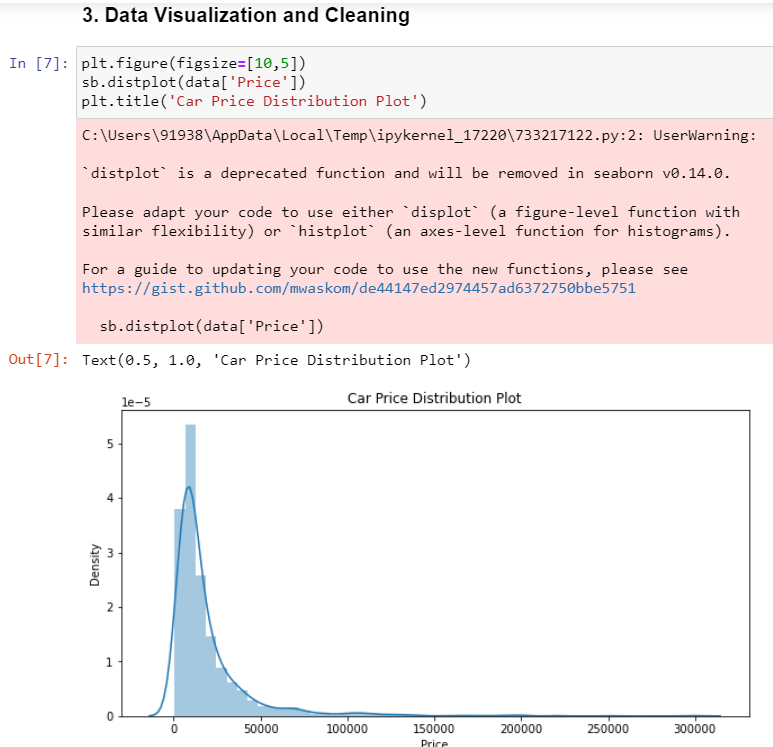
**Data Pre-processing**

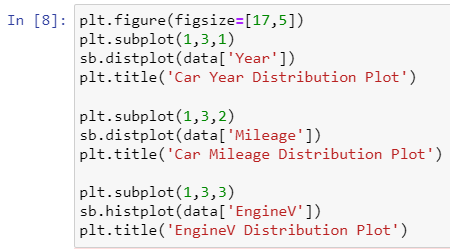
****

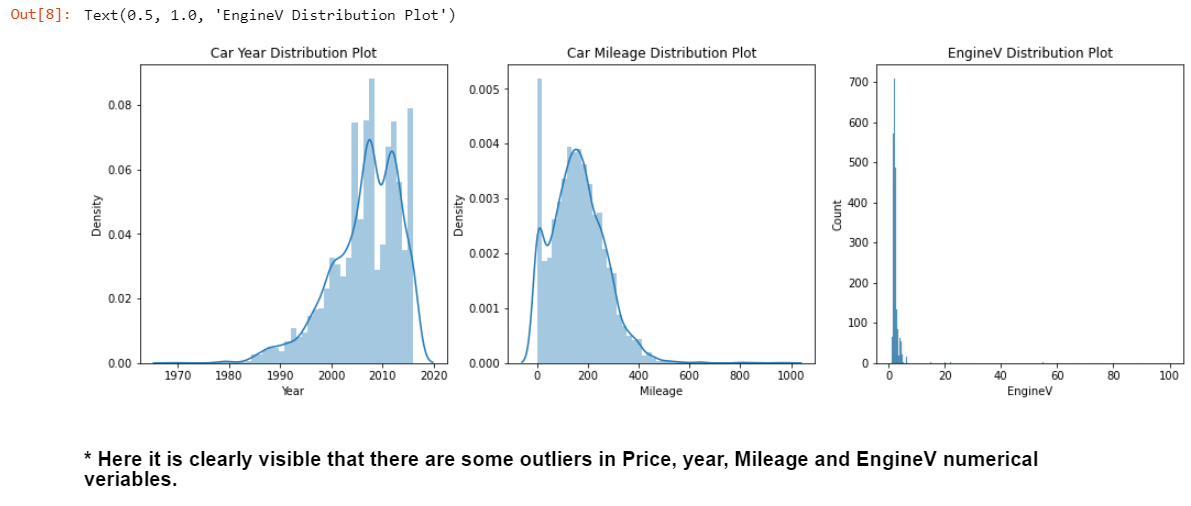
****

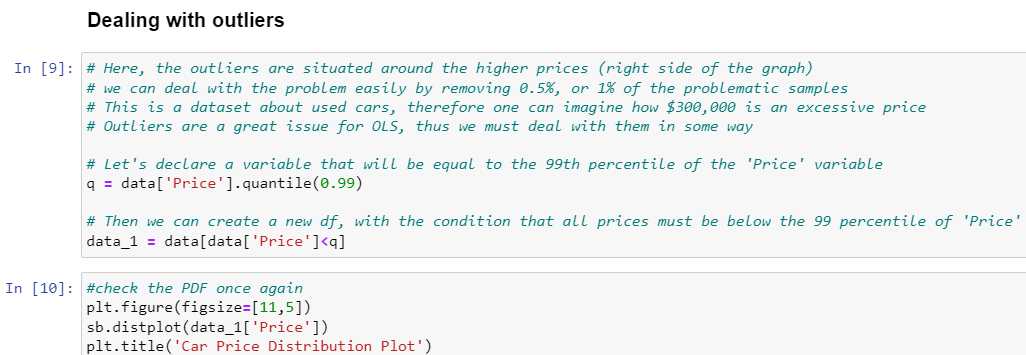
****

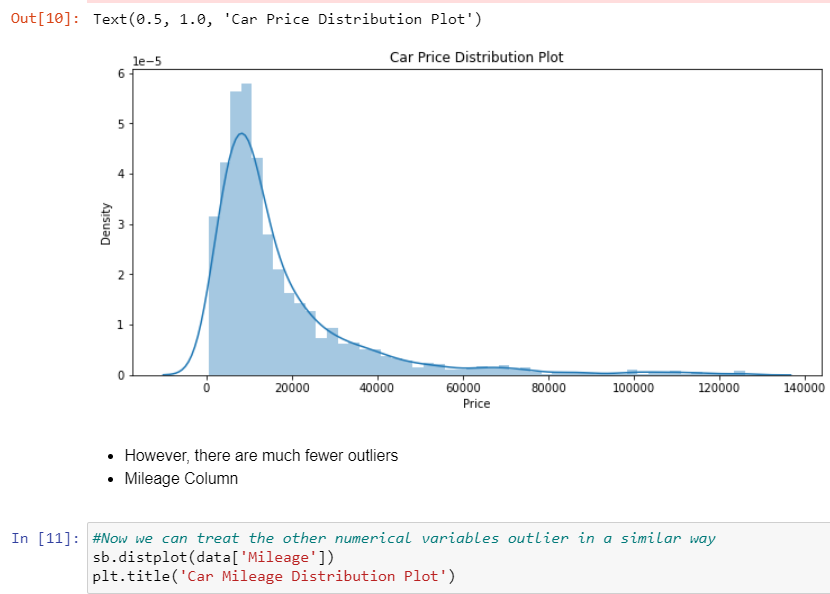
****

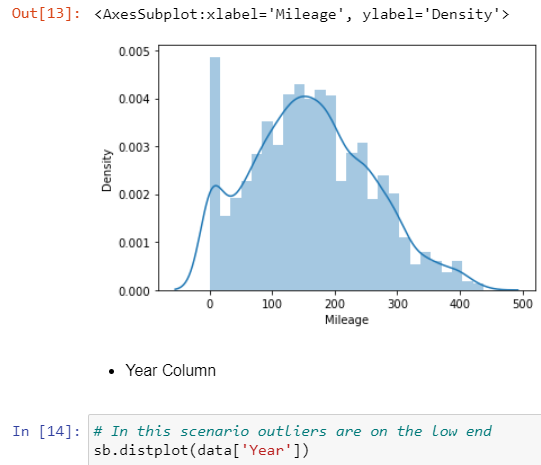
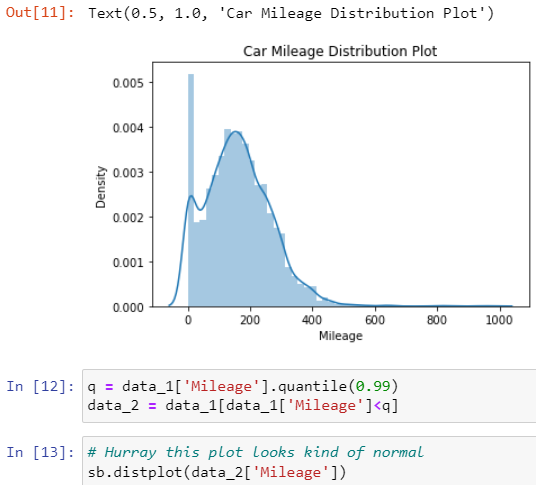
****

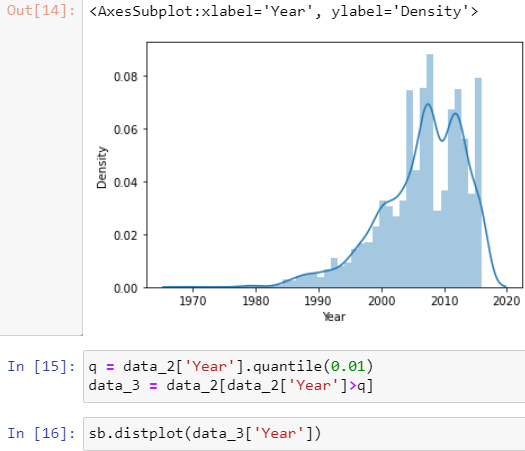
****

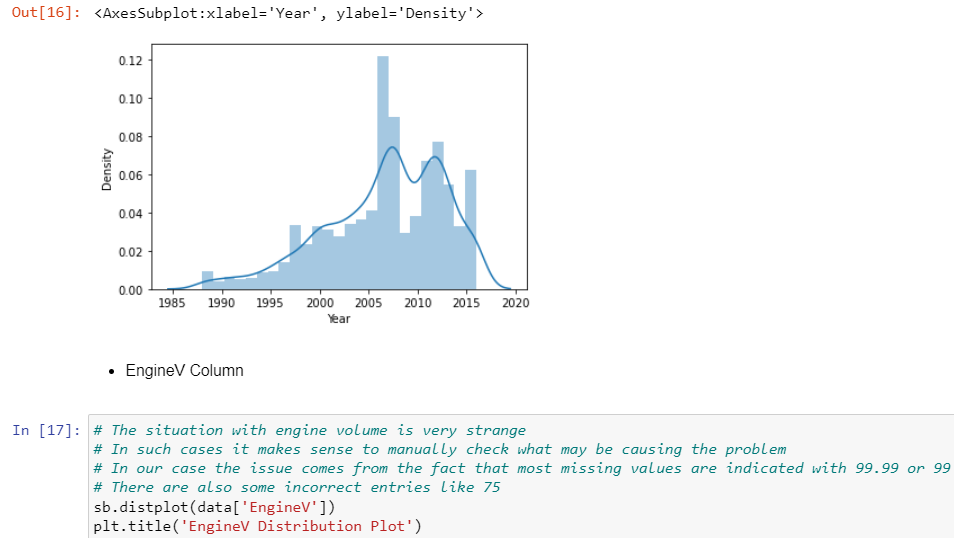
****

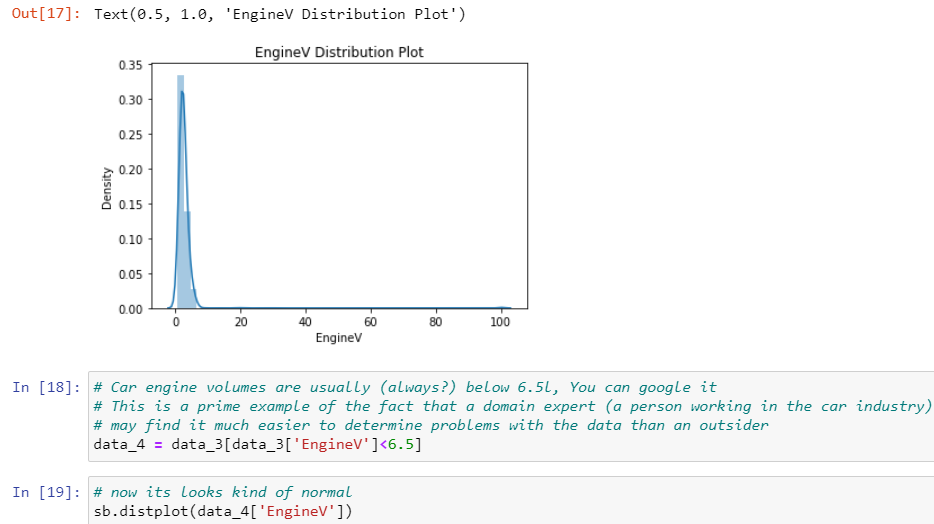
****

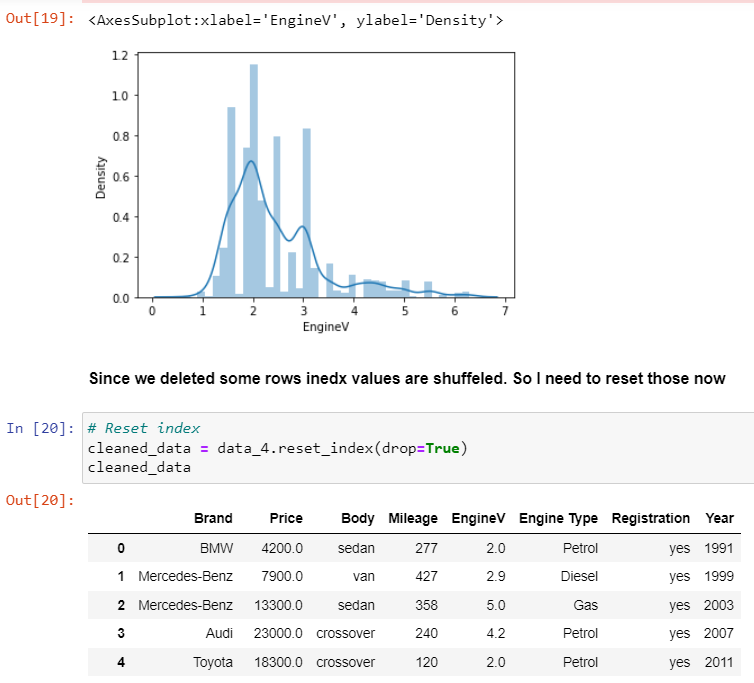
****

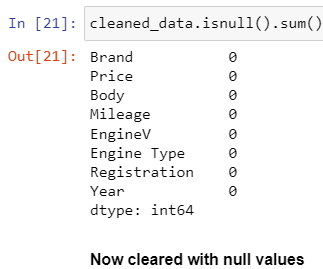
****

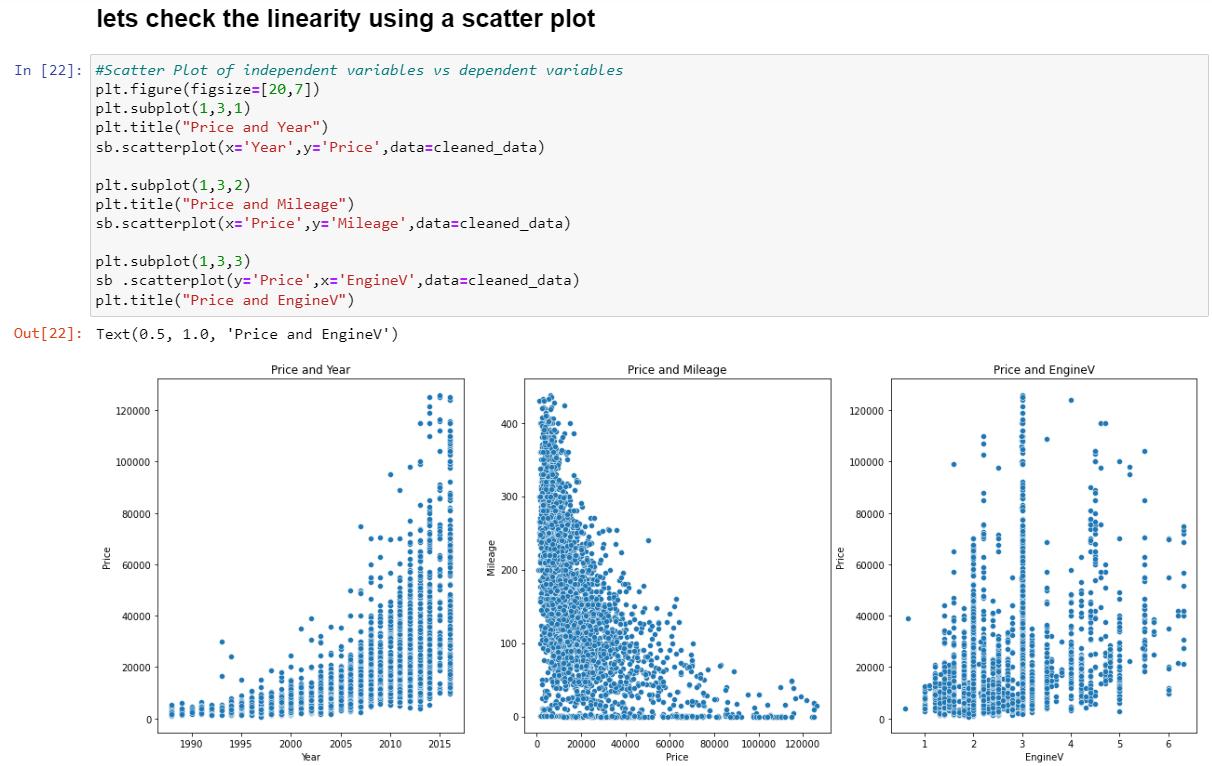
****

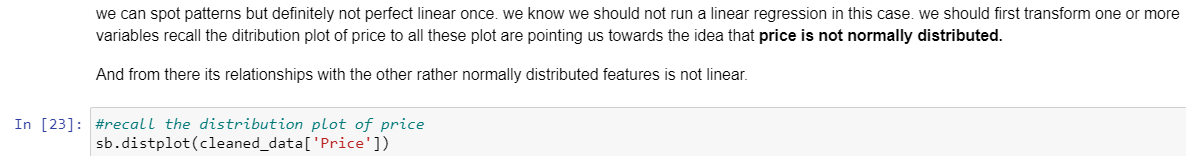
****

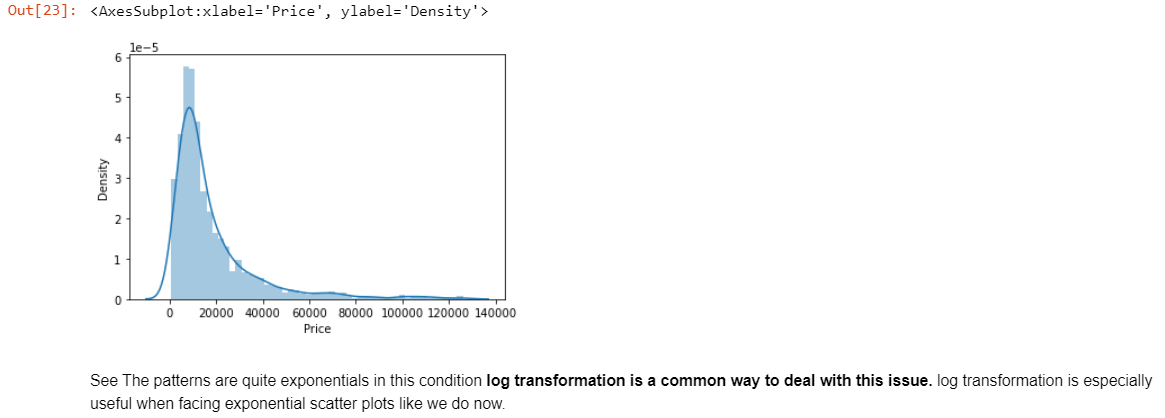
****

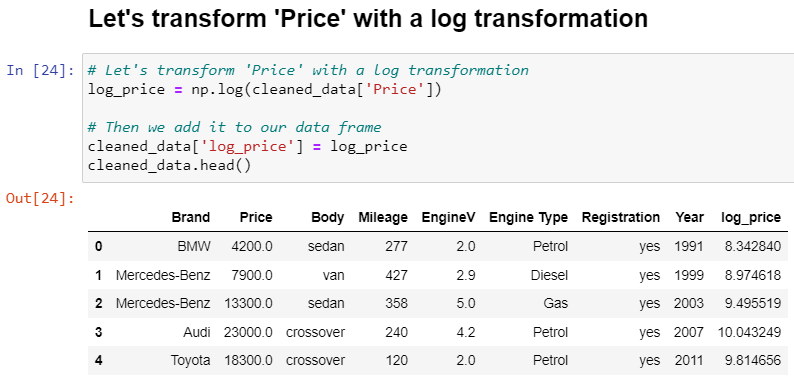
****

****

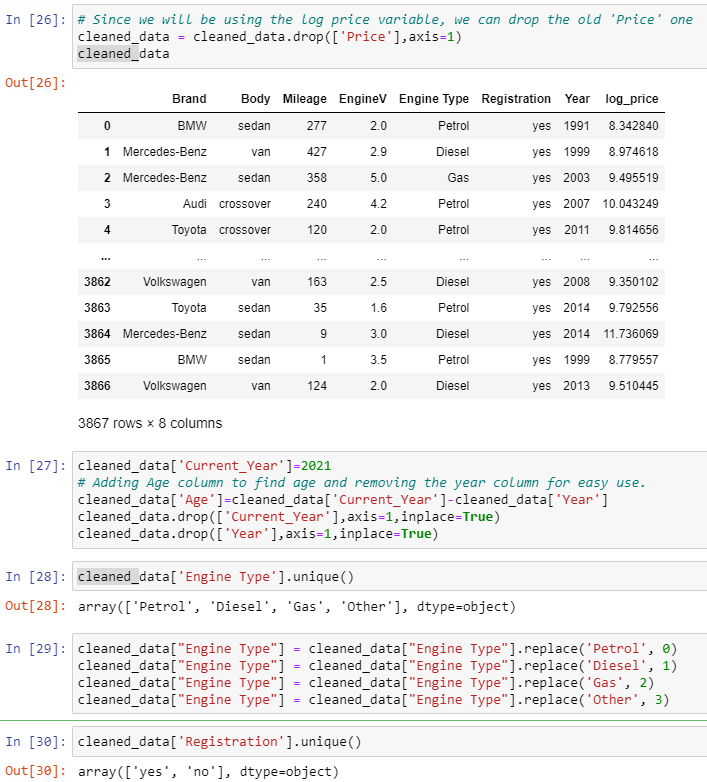
****

****

****

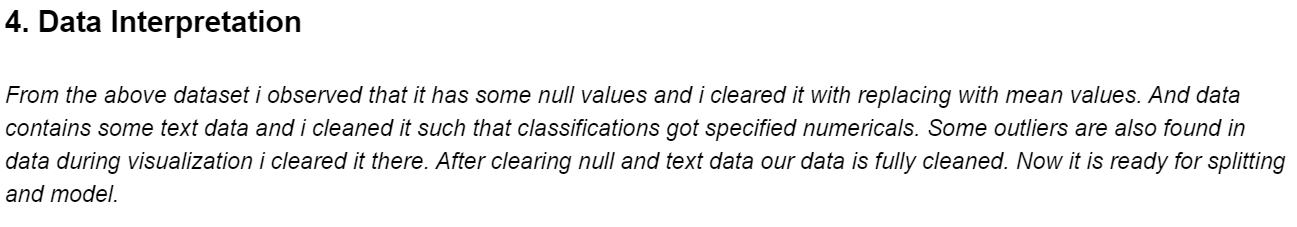
****

****

****

****

**Data Interpretation**

****

**Python Packages**

**Numpy**: NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

**Pandas**: pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. 8

**Seaborn**: Seaborn is a Python data visualisation library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

**matplotlib.pyplot**: It is a collection of functions that make matplotlib work like MATLAB. Each py plot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

**Nltk**: NLTK, or Natural Language Toolkit, is a Python package that you can use for NLP. used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

**Sklearn.feature\_extraction**: The sklearn.feature\_extraction module can be used to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image

**Learning Algorithms**

**DECISION TREE**

**Decision Tree** is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**KNN**

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as [GMM](https://en.wikipedia.org/wiki/Mixture_model), which assume a Gaussian distribution of the given data).

We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

**LINEAR REGRESSION**

**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used. There are many names for a regression’s dependent variable.  It may be called an outcome variable, criterion variable, endogenous variable, or regressand.  The independent variables can be called exogenous variables, predictor variables, or regressors.

**RANDOM FOREST REGRESSION MODEL**

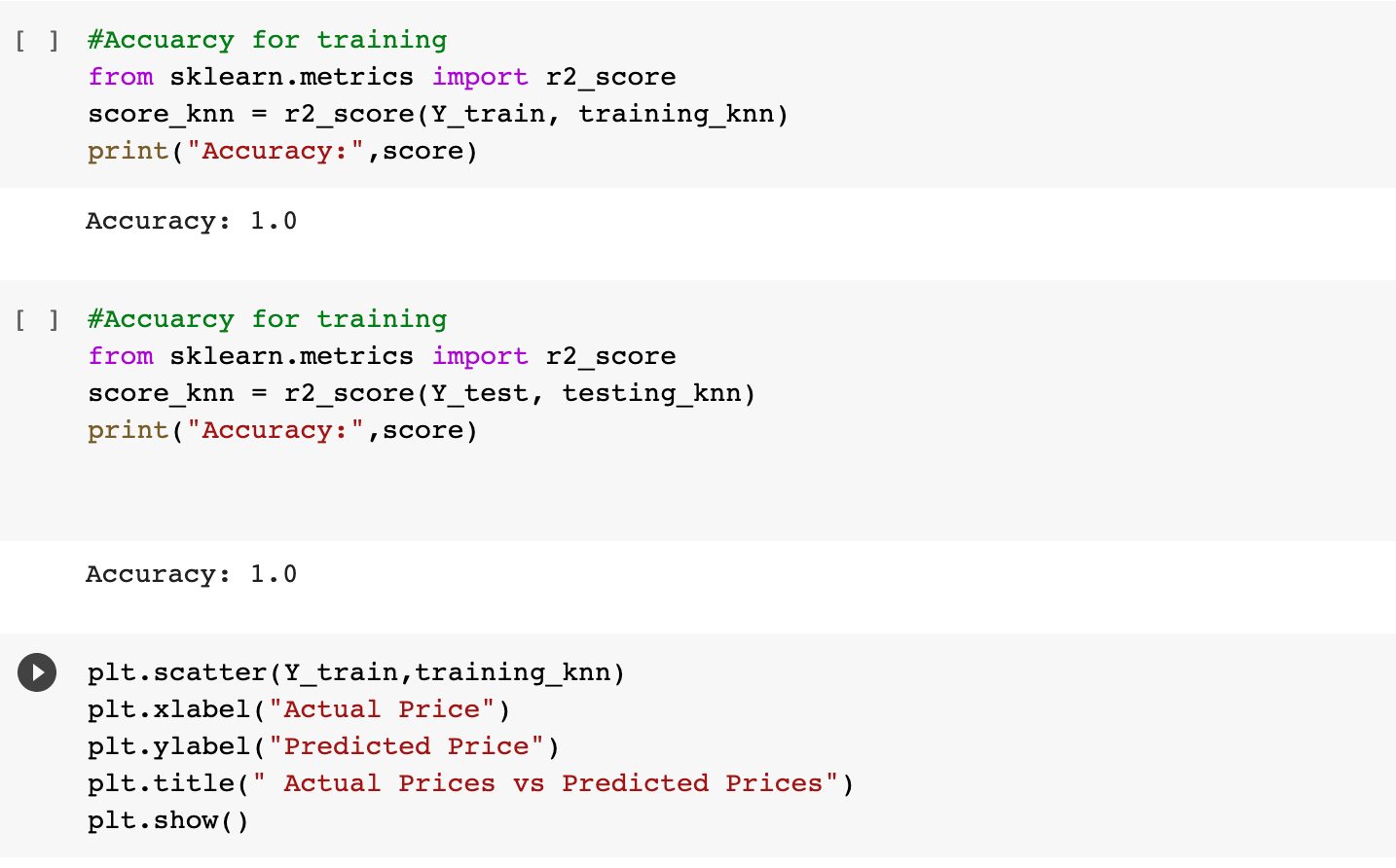
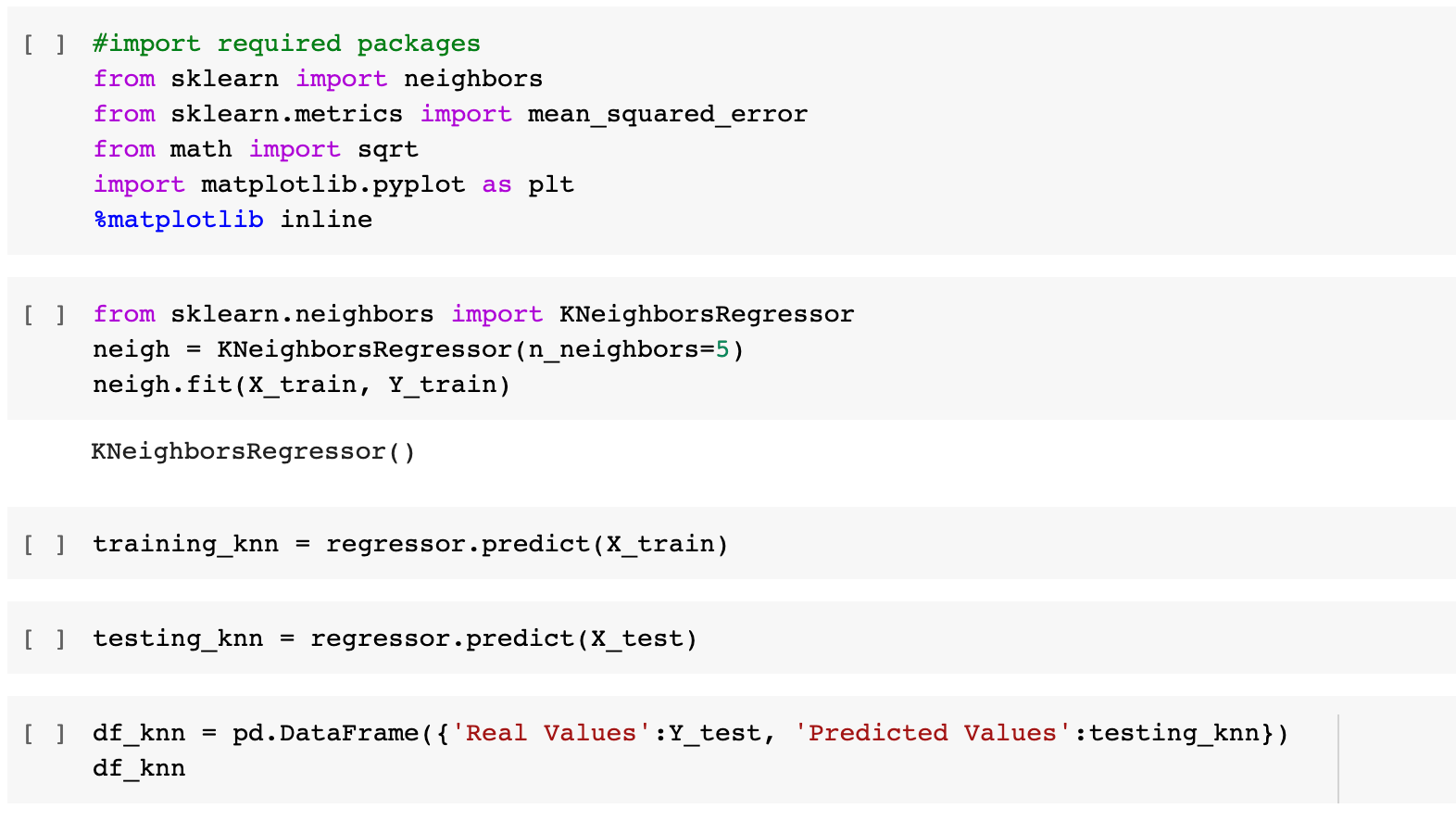
**Random Forest Regression** is a supervised learning algorithm that uses **ensemble learning** method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

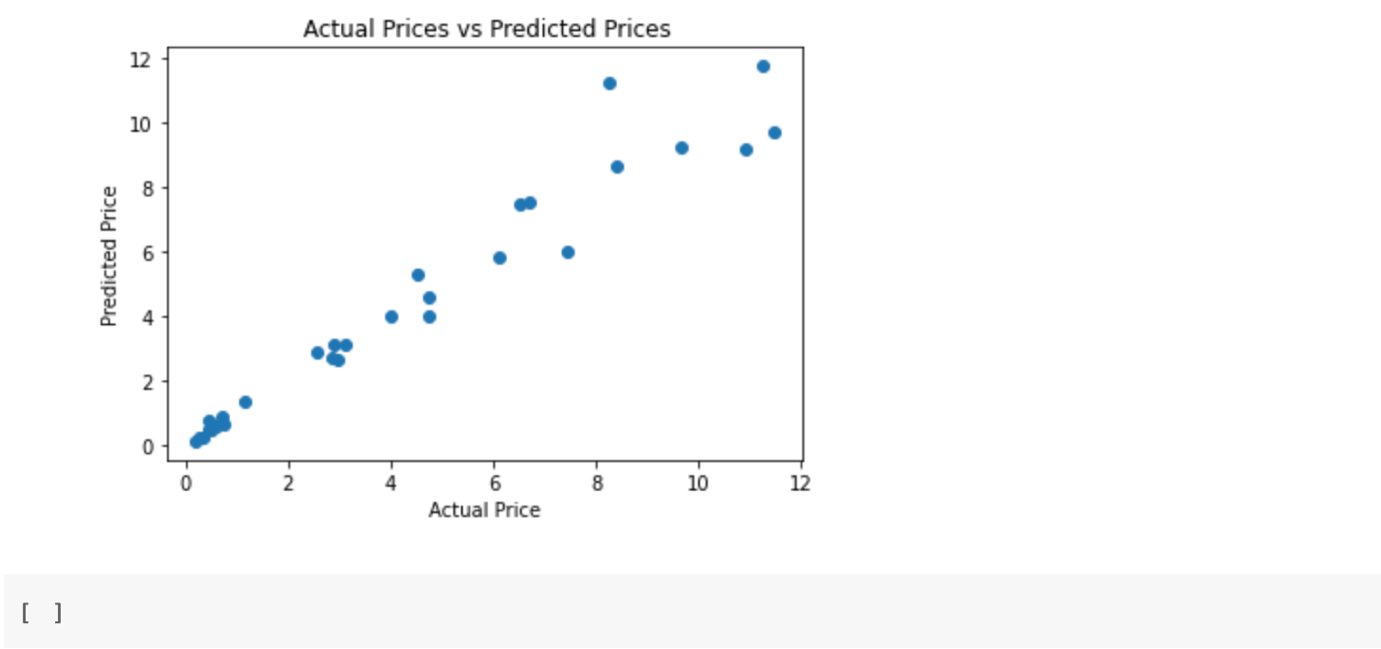
**GBT REGRESSION**

Gradient-boosted decision trees are **a popular method for solving prediction problems in both classification and regression domains**. The approach improves the learning process by simplifying the objective and reducing the number of iterations to get to a sufficiently optimal solution.

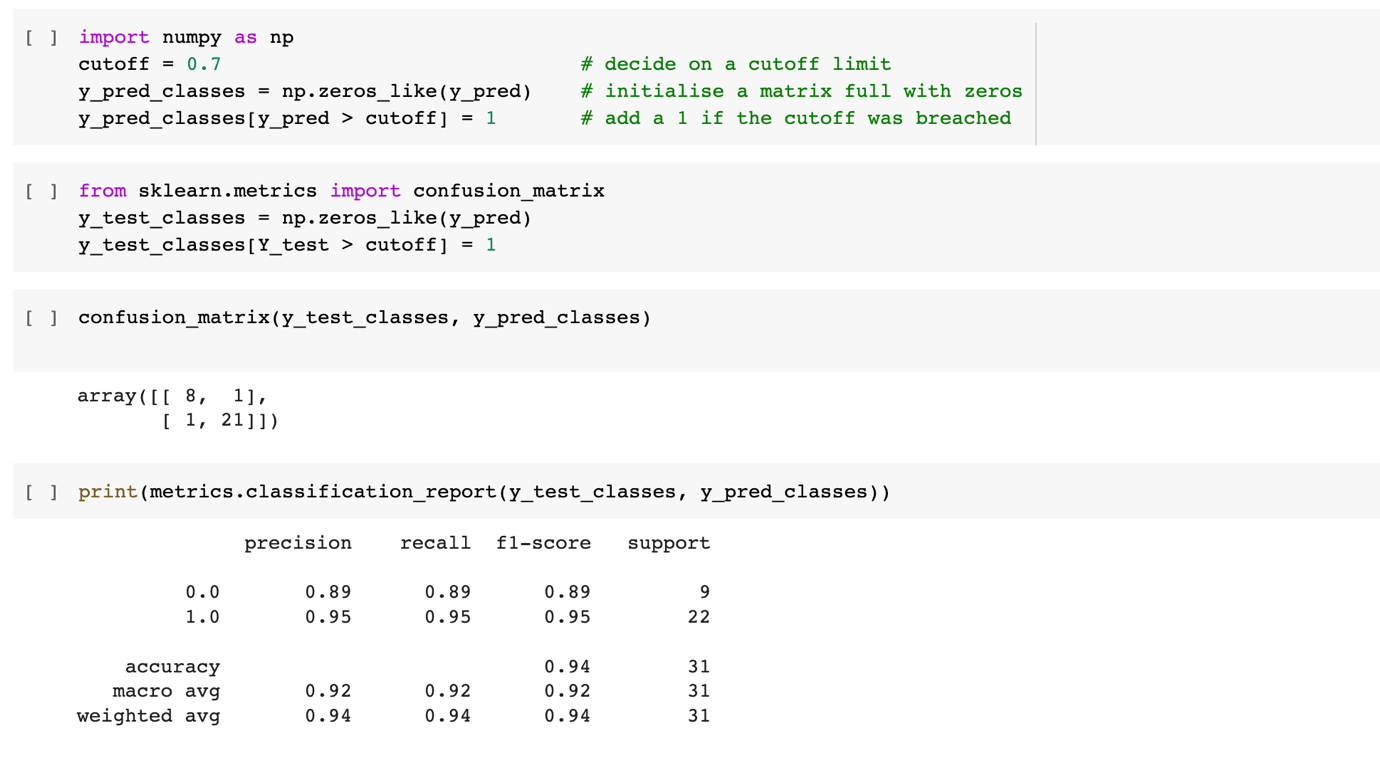
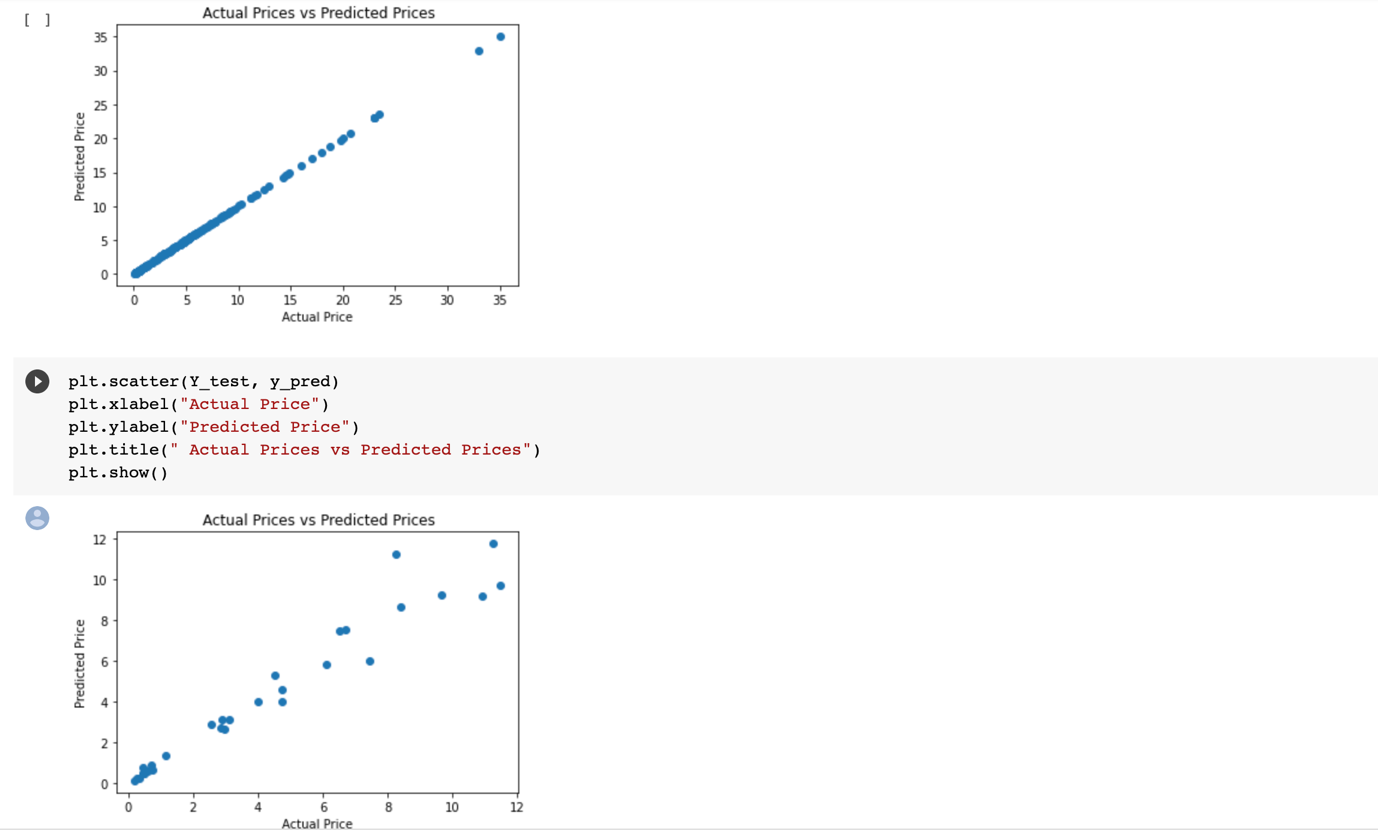
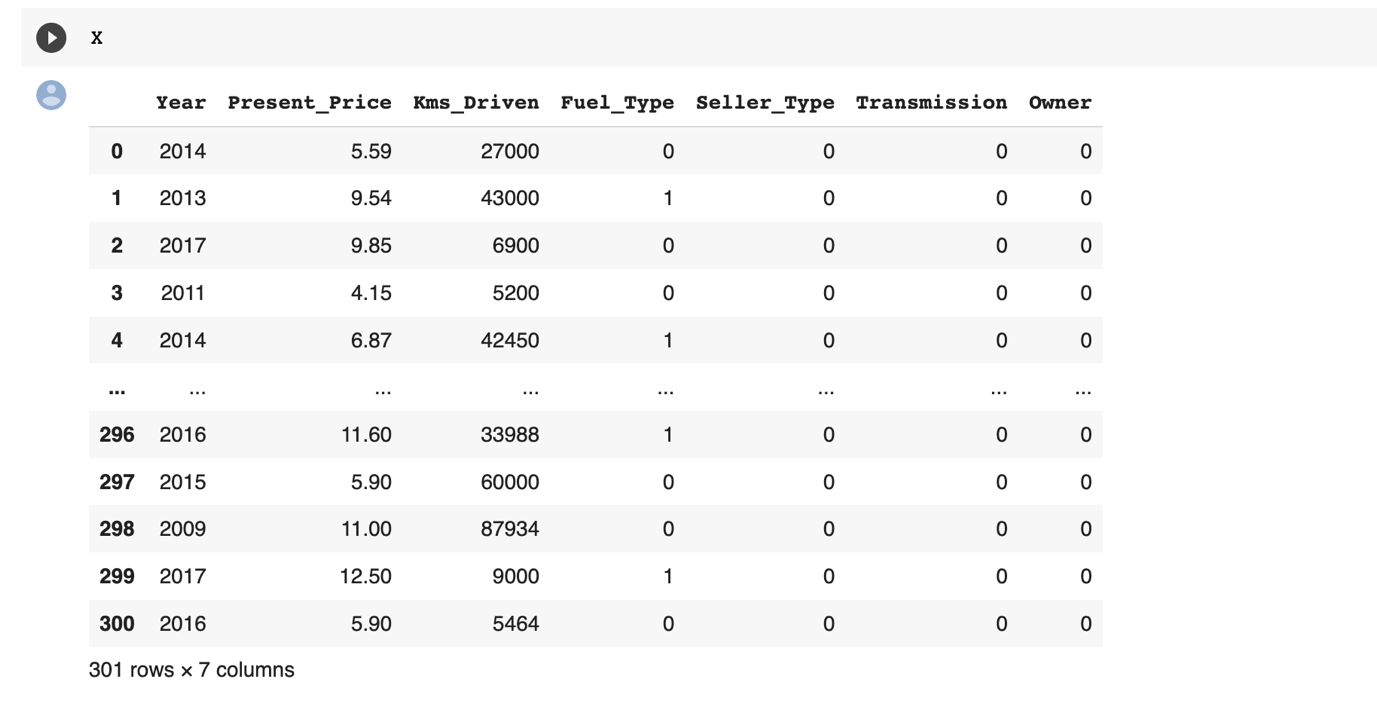
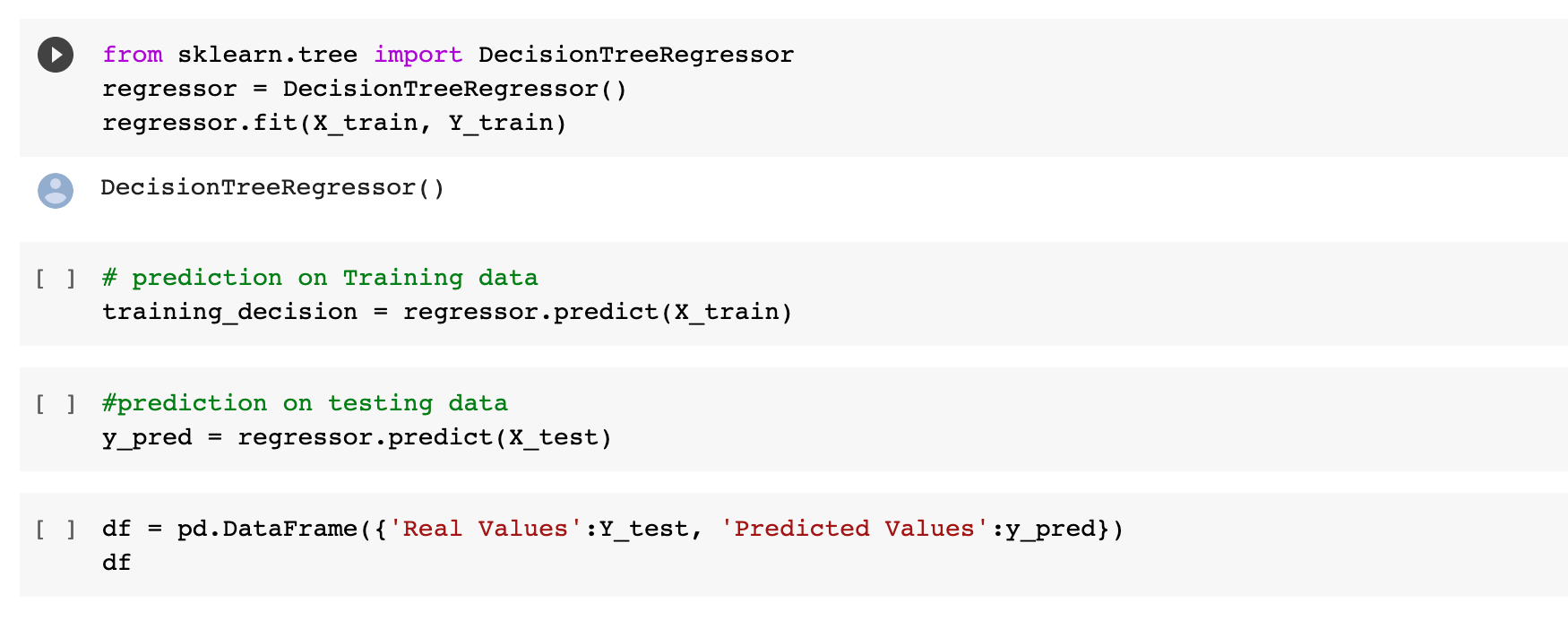
**DATASET 1**

**KNN**





**DECISION TREE**



**LINEAR REGRESSION**

Graphical user interface, text, application

Description automatically generated

**DATASET 2**

**Splitting the Dataset into X and Y**

Table

Description automatically generated

**DATA PREPROCESSING**

Text

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

**LINEAR REGRESSION**

Graphical user interface, text, application, email

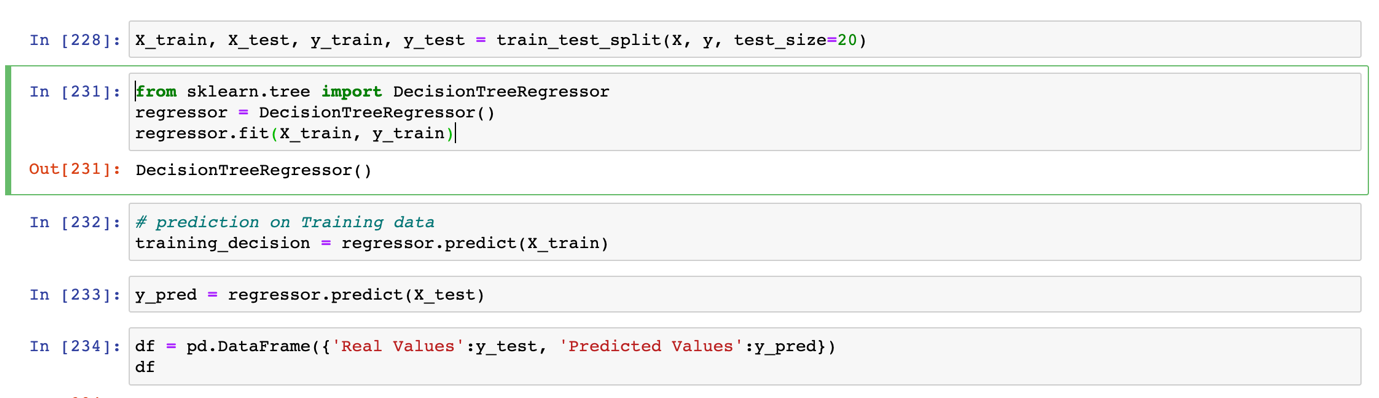
Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**DECISION TREE**



Graphical user interface, application, table

Description automatically generated with medium confidence

**KNN**

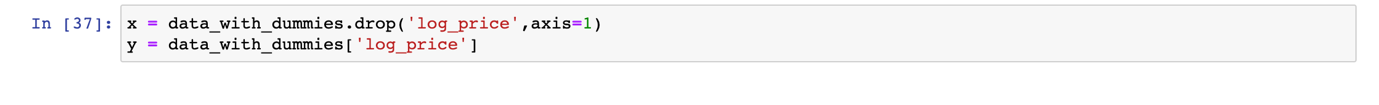
Text

Description automatically generated

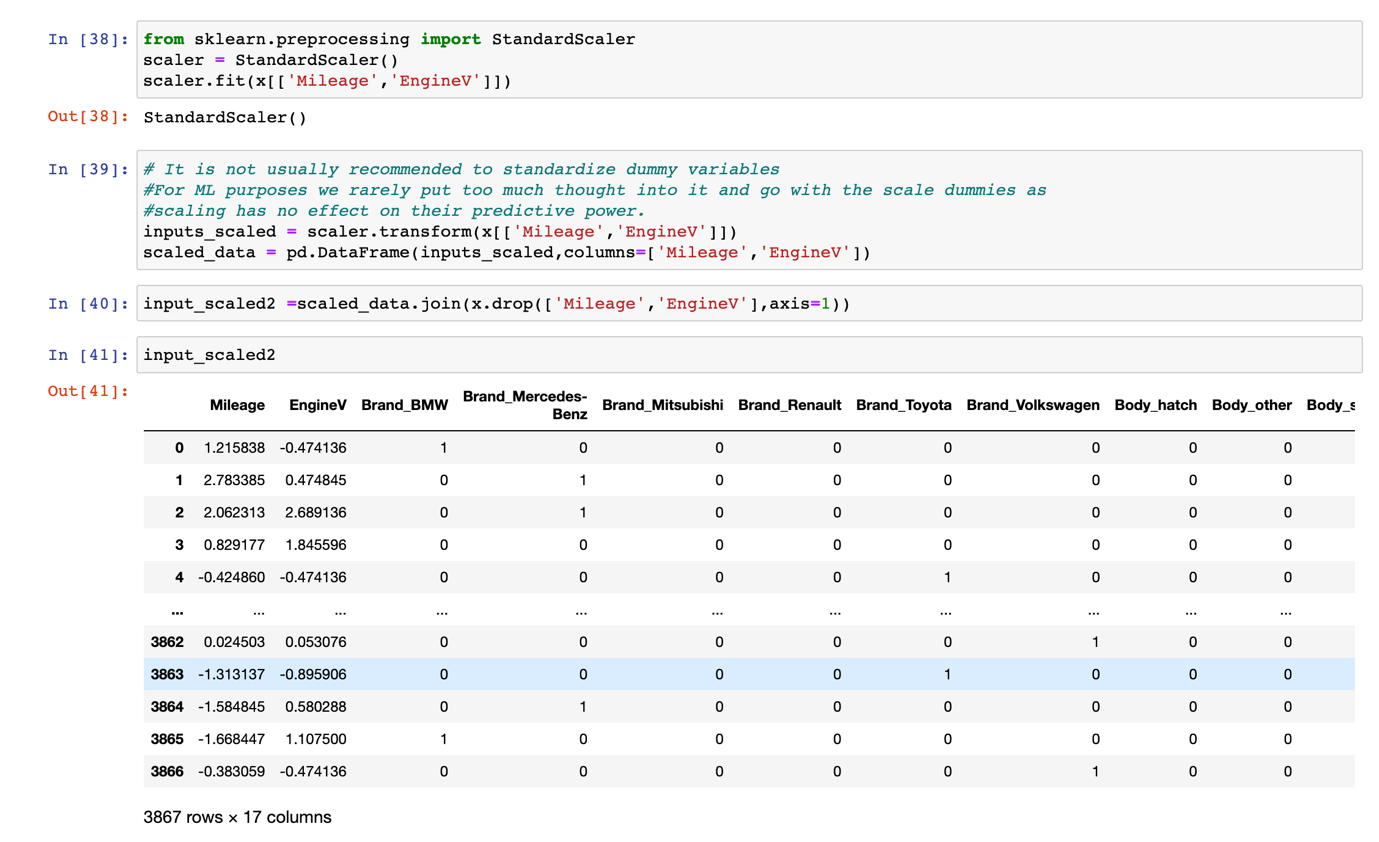
**DATASET 3**

**MODEL DEVELOPMENT**

Declaring dependent and independent variable



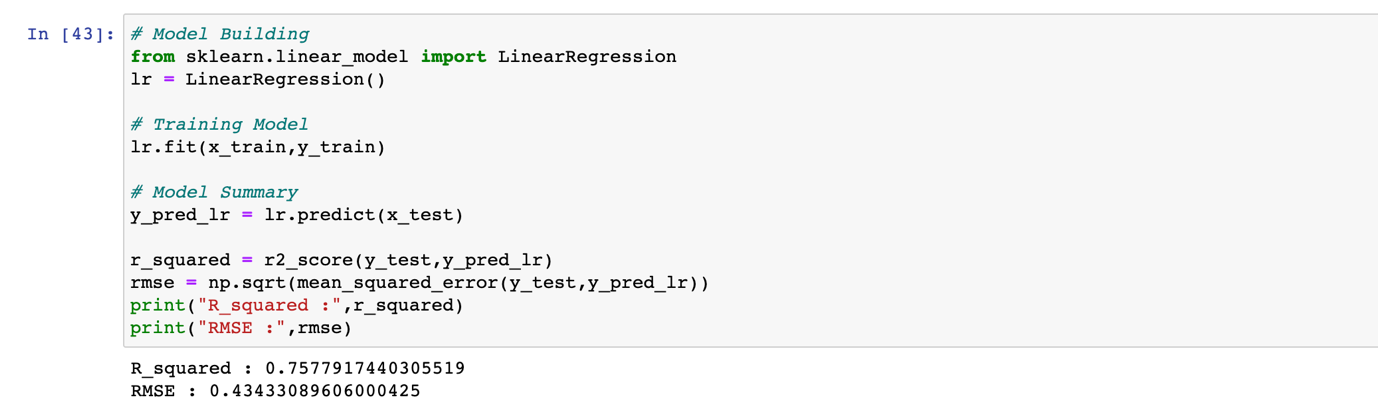
**FEATURE SCALING**



Splitting data into training, validation, and testing:

**LINEAR REGRESSION**

Brief on each ML algorithm chosen for creating model from your dataset with proper justification:



**RANDOM FOREST**

Graphical user interface, text, application, email

Description automatically generated

**GBT**

Graphical user interface, text, application, email

Description automatically generated

*After comparing all 3 model’s accuracies,*

*So, as we try different Regression Algorithms and found that "Random Forest Regressor Model" is giving better accuracy compare to others with 0.798 accuracy*

**Select The Best Model**

So as we try different Regression Algorithms and found that ***"GBT Regressor Model"*** is giving better accuracy compare to other.

*let's manually check these predictions*

Graphical user interface, text, application

Description automatically generated

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**