
Faculty of Engineering and Technology

Electrical and Computer Engineering Department

ENCS5341

Machine Learning and Data Science

---

# Assignment No.1

Studying the dataset and analyzing aspects

---

**Student's Name:** Lojain Abdalrazaq. **ID:** 1190707.

**Instructor's Name: Dr.** Yazan Abu Farha.

Section: 2.

November 29, 2023

## Summarization the Results

*Question 1:*

1- Read the dataset and examine how many features and examples does it have?
*(Hint: you can use `Pandas` to load the dataset into a `dataframe`)*

In this part the Panda was used , and the shape[0], and shape[1] were used to find the number of lines (records), and the number of features (columns) in the dataset.

```
Number of lines in the CSV file:  398
Number of features in the CSV file:  8
```

```
In [5]: import pandas as pd
        # reading the file in the Panda framework object
        df = pd.read_csv('cars.csv')
        # using the shape attribute to find the number of lines and features in the cvs file
        num_lines = df.shape[0]
        num_features = df.shape[1]

        print("Number of lines in the CSV file: ", num_lines)
        print("Number of features in the CSV file: ",num_features )

        Number of lines in the CSV file:  398
        Number of features in the CSV file:  8
```

And here is the loaded dataset (398 records, and 8 features).

```
In [10]: # loaded data in the data frame
         df

Out[10]:
```

| | mpg | cylinders | displacement | horsepower | weight | acceleration | model_year | origin |
|---|---|---|---|---|---|---|---|---|
| 0 | 18.0 | 8 | 307.0 | 130.0 | 3504 | 12.0 | 70 | USA |
| 1 | 15.0 | 8 | 350.0 | 165.0 | 3693 | 11.5 | 70 | USA |
| 2 | 18.0 | 8 | 318.0 | 150.0 | 3436 | 11.0 | 70 | USA |
| 3 | 16.0 | 8 | 304.0 | 150.0 | 3433 | 12.0 | 70 | USA |
| 4 | 17.0 | 8 | 302.0 | 140.0 | 3449 | 10.5 | 70 | USA |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 393 | 27.0 | 4 | 140.0 | 86.0 | 2790 | 15.6 | 82 | USA |
| 394 | 44.0 | 4 | 97.0 | 52.0 | 2130 | 24.6 | 82 | Europe |
| 395 | 32.0 | 4 | 135.0 | 84.0 | 2295 | 11.6 | 82 | USA |
| 396 | 28.0 | 4 | 120.0 | 79.0 | 2625 | 18.6 | 82 | USA |
| 397 | 31.0 | 4 | 119.0 | 82.0 | 2720 | 19.4 | 82 | USA |

398 rows × 8 columns

*Question 2:*

2- Are there features with missing values? How many missing values are there in each one?
*(Hint: you can use `isnull()` from `Pandas`)*

Now, in this part we want to know the total number of missing values in the dataset, this done by using the *isnull() function.*

The following table represents the Boolean table such that the *false* value means the index is not null or missed, while the *true* value means the index has a missing value.

```
In [12]: missing_values = df.isnull()
         missing_values
```

Out[12]:

| | mpg | cylinders | displacement | horsepower | weight | acceleration | model_year | origin |
|---|---|---|---|---|---|---|---|---|
| 0 | False | False | False | False | False | False | False | False |
| 1 | False | False | False | False | False | False | False | False |
| 2 | False | False | False | False | False | False | False | False |
| 3 | False | False | False | False | False | False | False | False |
| 4 | False | False | False | False | False | False | False | False |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 393 | False | False | False | False | False | False | False | False |
| 394 | False | False | False | False | False | False | False | False |
| 395 | False | False | False | False | False | False | False | False |
| 396 | False | False | False | False | False | False | False | False |
| 397 | False | False | False | False | False | False | False | False |

398 rows × 8 columns

- And the number of missing values in 8.
- Horsepower: 6 missing values.
- Origin: 2 missing values.

```
[398 rows x 8 columns]
mpg             0
cylinders       0
displacement    0
horsepower      6
weight          0
acceleration    0
model_year      0
origin          2
dtype: int64
```

*Question 3:*

3- Fill the missing values in each feature using a proper imputation method (for example: fill with mean, median, or mode)

For the first feature (*Horsepower*) I used the mean to fill the missing values. While the (*origin*) feature, the mode was used which represents the most frequent value of the feature.

In addition, it was checked for a second time, and the number of missed values were 0 instead of 8 values.

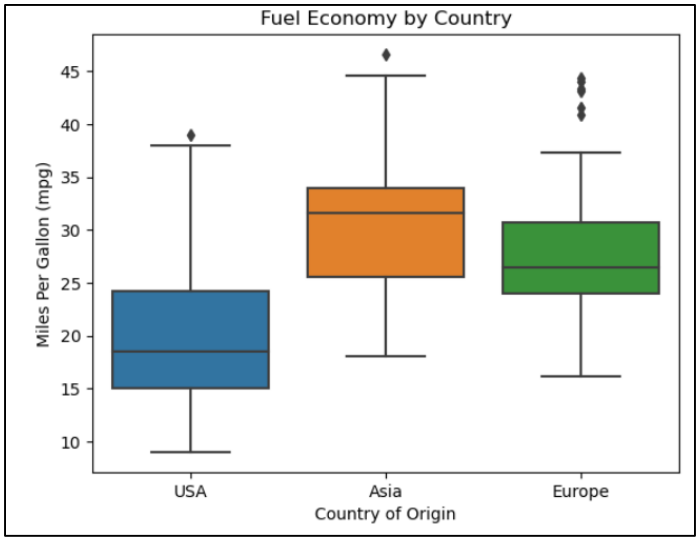
```
In [34]: # filling the missing values of the features
         df['horsepower'].fillna(value = df.horsepower.mean(), inplace = True)
         df['origin'].fillna(value = df.origin.mode().iloc[0], inplace=True)
         # testing
         count_missing_values = df.isnull().sum()
         count_missing_values

Out[34]: mpg             0
         cylinders       0
         displacement    0
         horsepower      0
         weight          0
         acceleration    0
         model_year      0
         origin          0
         dtype: int64
```

*Question 4:*

4- Which country produces cars with better fuel economy?
*(Hint: use box plot that shows the mpg for each country (all countries in one plot))*

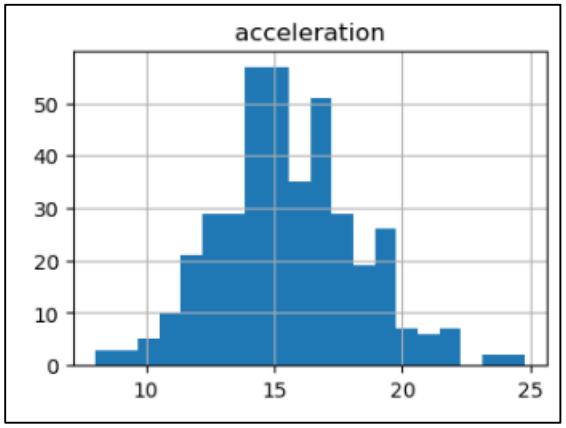
In this part, the following figure is the result from plotting the 'country' vs 'mpg' features in X and Y axis. And from the plot, it is notices that the Asia has the most economic cars, then Europe, and finally America.

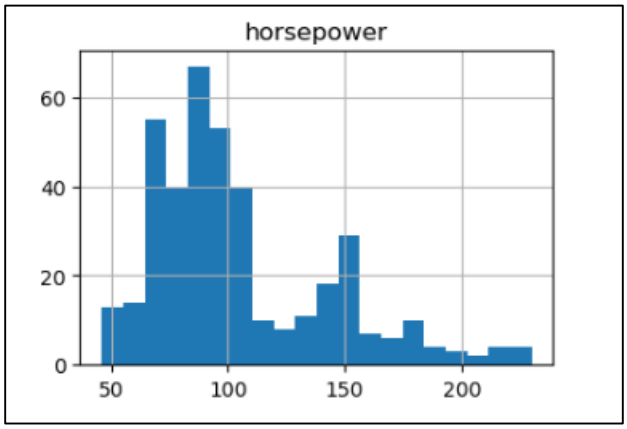


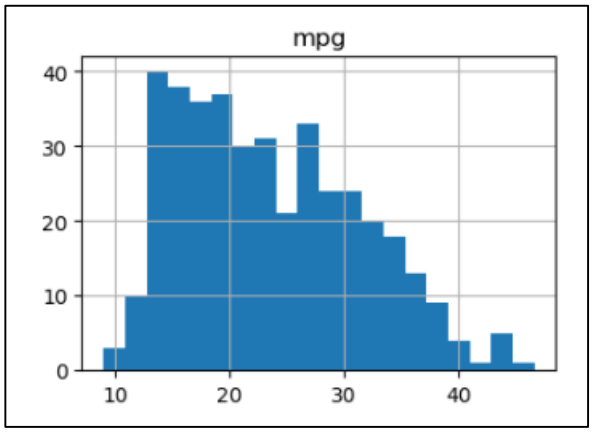

*Question 5:*

5- Which of the following features has a distribution that is most similar to a Gaussian: 'acceleration', 'horsepower', or 'mpg'? Answer this part by showing the histogram of each feature.

The histogram of each feature is as the following:




**From the given histograms, the plot of the "acceleration" feature is the most similar to the Gaussian distribution.**

*Question 6:*

6- Support your answer for part 5 by using a quantitative measure.

To check the normalization of the required features in quantitative measure, I used the p-value measure, such that of the P>=0.05, then the feature is normally distributed, while P<0.05 has a less chance to be normally distributed.

The P-Value of the 3 features is as the following:

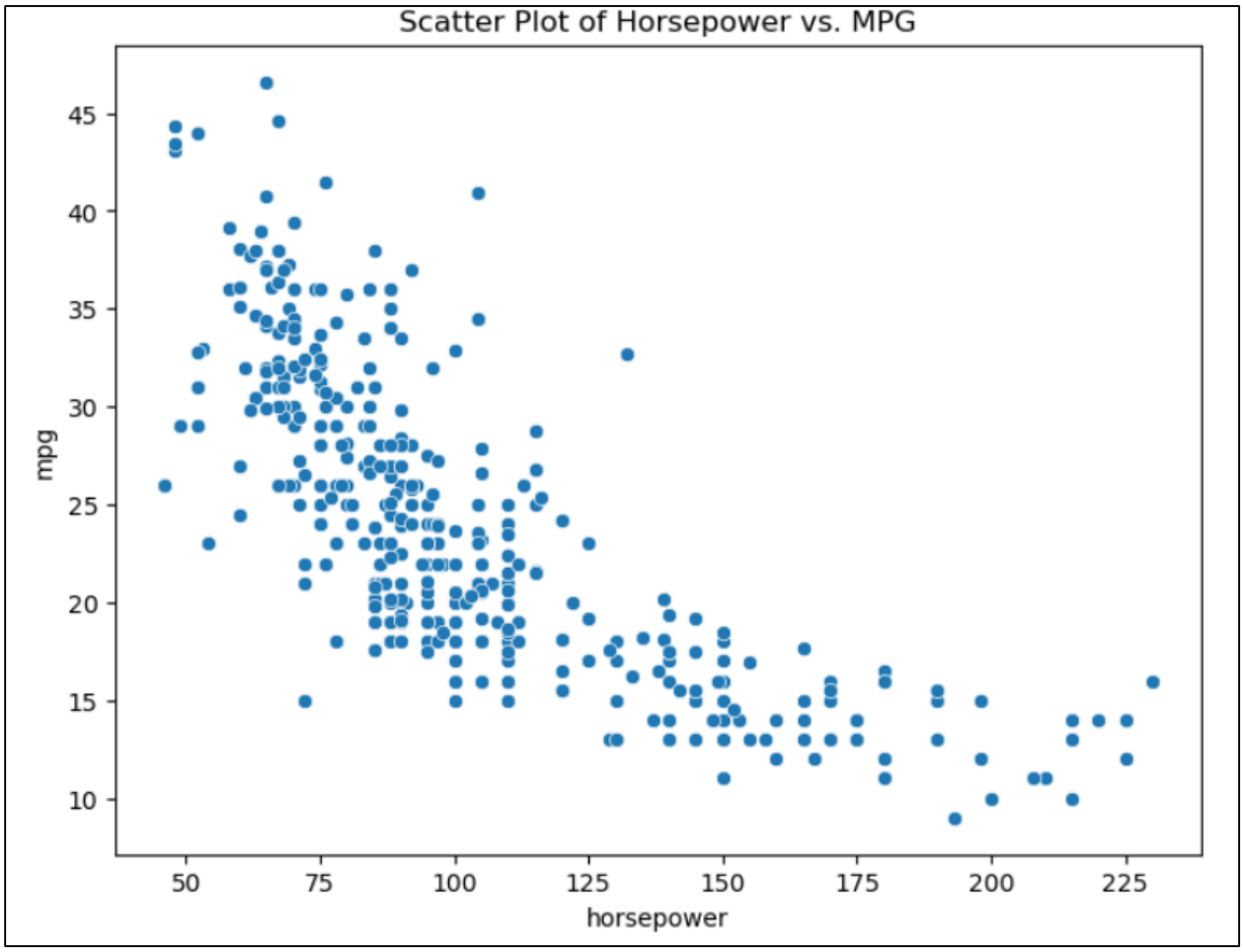
- Acceleration: pvalue=0.039872437715530396
- Horsepower: pvalue=4.5655601017434086e-15
- Mpg: pvalue=1.1833407853600875e-07.

And it is noticed that the last two values are too small, while the Acceleration is the most close to the 0.05 value, so it is the most likely to be similar to the Gaussian distribution.

*Question 7:*

7- Plot a scatter plot that shows the 'horsepower' on the x-axis and 'mpg' on the y-axis. Is there a correlation between them? Positive or negative?

From the figure, it is noticed that there is a negative correlation between the "mpg" and the "horsepower".

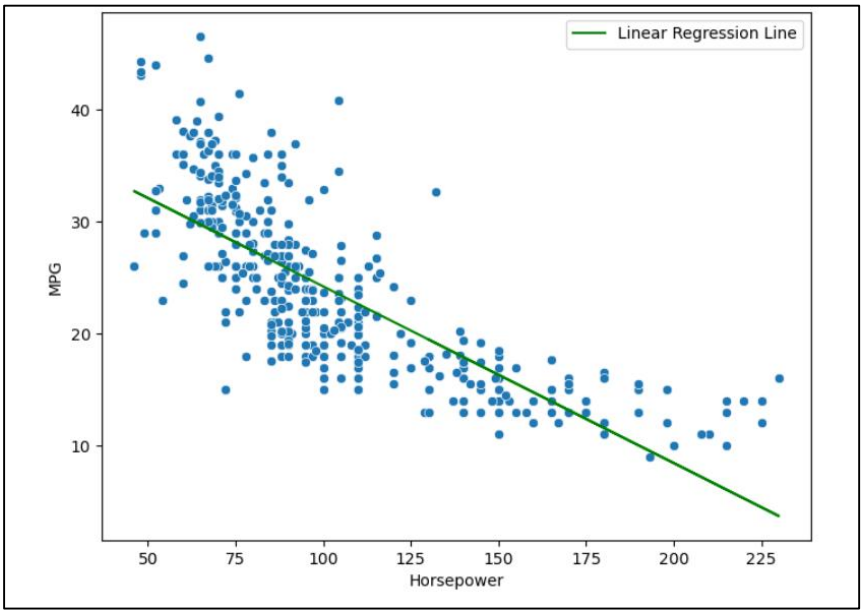



*Question 8:*

8- Implement the closed form solution of linear regression and use it to learn a linear model to predict the 'mpg' from the 'horsepower'. Plot the learned line on the same scatter plot you got in part 7.
*(Hint: This is a simple linear regression problem (one feature). Do not forget to add $x_0$=1 for the intercept. For inverting a matrix use np.linalg.inv from NumPy)*

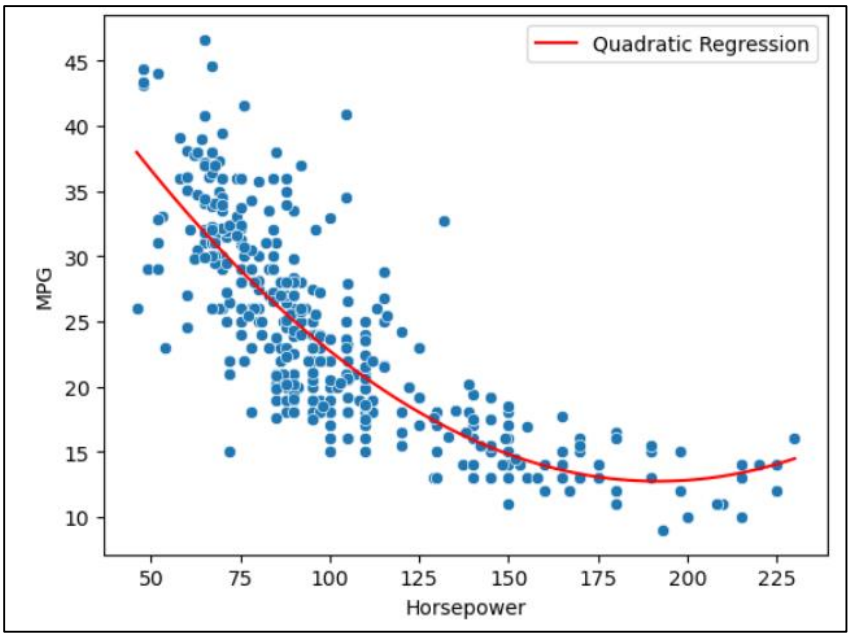
Using the linear system equation, the learned line of the simple linear regression is as the following:

e linear system of equations: $\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$




*Question 9:*

9- Repeat part 8 but now learn a quadratic function of the form $f = w_0 + w_1 x + w_2 x^2$.



*Question 10:*

10- Repeat part 8 (simple linear regression case) but now by implementing the gradient descent algorithm instead of the closed form solution.

It is noticed that the result is the same as the linear system equation as shown in the following result.