جامعة بيرزيت

BIRZEIT UNIVERSITY

# *Electrical and Computer Engineering Department*

# *Machine Learning and Data Science - ENCS5341*

# *Assignment #1*

***Prepared by:*** Lana Batnij __ 1200308

***Instructor***: Dr.Yazan Abu Farha

***Section:*** 2

## *Table of Contents:*

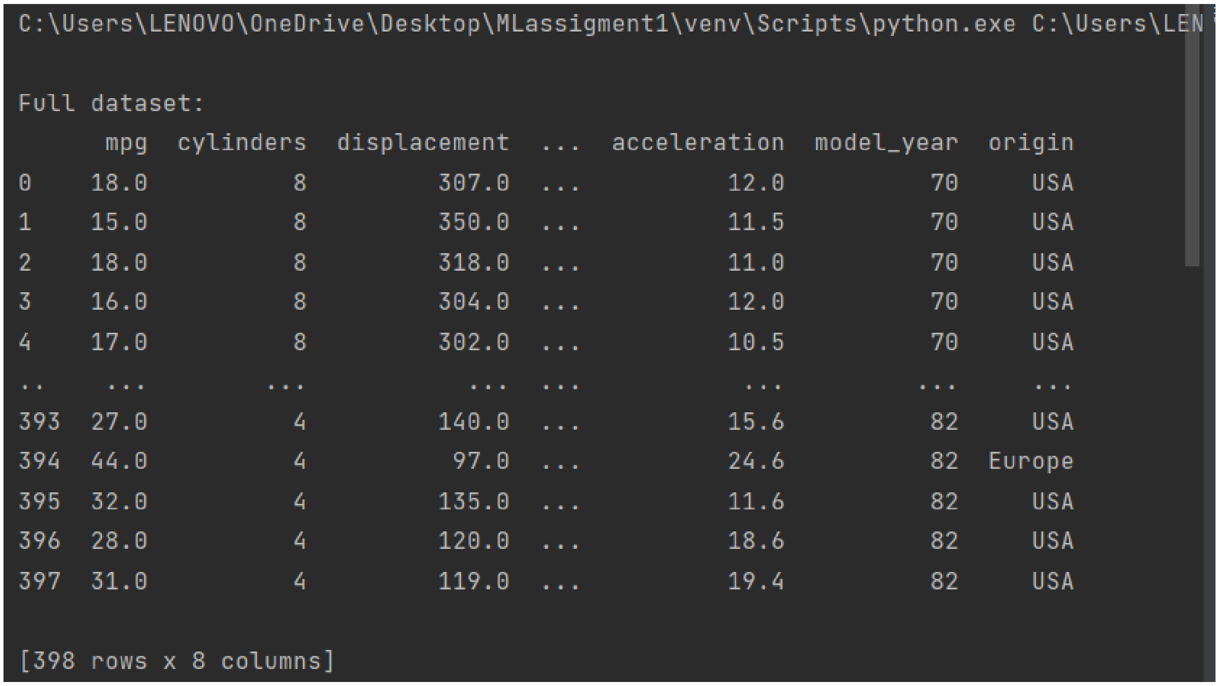
## *Table of Figures:*

## *List of Tables:*

- ***Assignment Progress and Results Discussion:***

○ ***Part 1:***

Initially, the tool used was PYCHARM, and the first step was to read the dataset and see how many features and examples it had. This was accomplished by using Pandas to convert the CSV file into a Data Frame, which is a two-dimensional labeled data structure. As Figure 1 shows the data set was read and it has 399(From 0 to 398) rows which are the examples and 8 columns that represent the features.

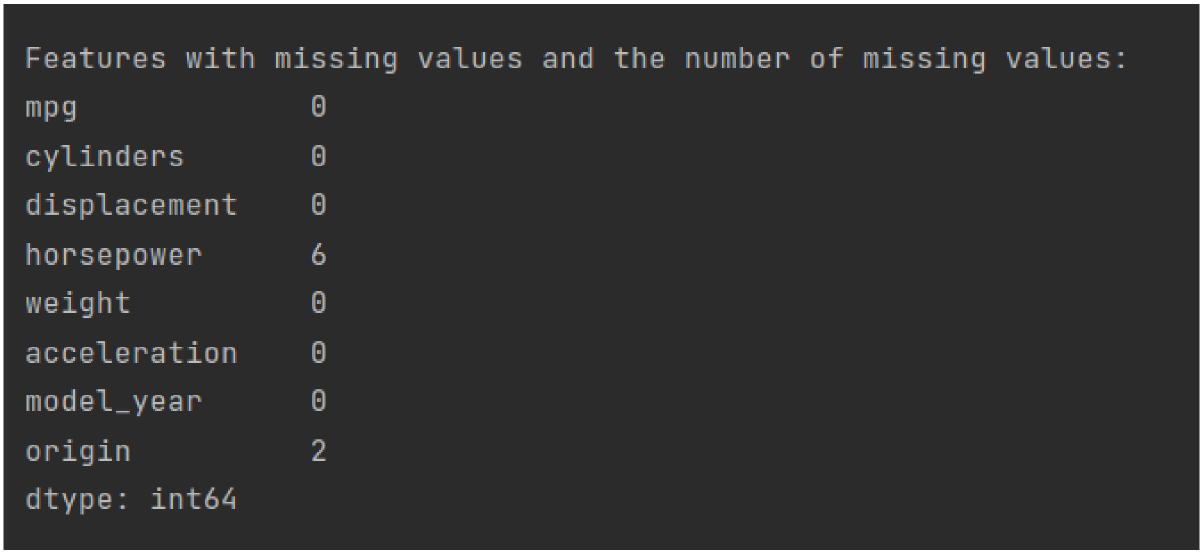


```
C:\Users\LENOVO\OneDrive\Desktop\MLassigment1\venv\Scripts\python.exe C:\Users\LEN

Full dataset:
      mpg  cylinders  displacement  ...  acceleration  model_year  origin
0    18.0          8         307.0  ...          12.0          70     USA
1    15.0          8         350.0  ...          11.5          70     USA
2    18.0          8         318.0  ...          11.0          70     USA
3    16.0          8         304.0  ...          12.0          70     USA
4    17.0          8         302.0  ...          10.5          70     USA
..    ...        ...           ...  ...           ...         ...     ...
393  27.0          4         140.0  ...          15.6          82     USA
394  44.0          4          97.0  ...          24.6          82  Europe
395  32.0          4         135.0  ...          11.6          82     USA
396  28.0          4         120.0  ...          18.6          82     USA
397  31.0          4         119.0  ...          19.4          82     USA

[398 rows x 8 columns]
```

*Figure 1: Reading the Data Set Result*

○ ***Part 2:***

Detecting Is there a feature with missing values, and if yes, how many missing values are there in each one? This was done with isnull (), which returns True where the corresponding element in df is missing, and False. Pandas provides this method of operation, and the sum () function is then used to tell the number of True values along each column, generating the number of missing data in each feature.

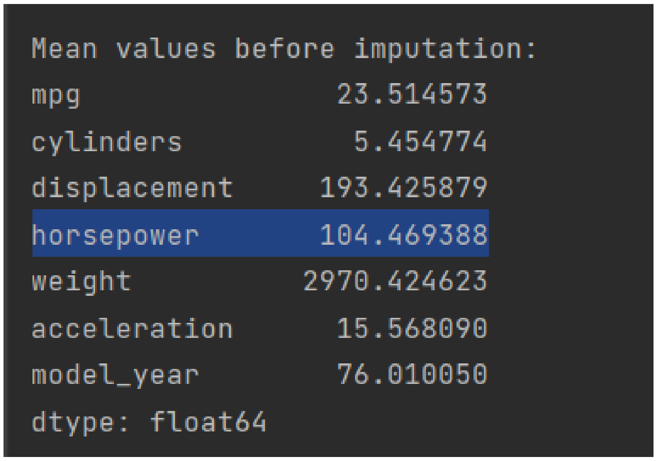


```
Features with missing values and the number of missing values:
mpg             0
cylinders       0
displacement    0
horsepower      6
weight          0
acceleration    0
model_year      0
origin          2
dtype: int64
```

*Figure 2: Features' Missing Values Counting Results*

- ***Part 3:***

After finding the missing values in which Features exactly, The Data Type of each feature was checked, and then the Mean value of Each Feature was used to fill in the Missing Values. The mean value of the horsepower Feature is 104.469 and the origin feature is NAN which means no known value. The values were calculated by the mean () function. By the The missing values in the horsepower feature were replaced by 104.469, and the missing values in the origin feature were replaced by NAN which means not a number and the value will remain not Known but not null anymore.



```
Mean values before imputation:
mpg               23.514573
cylinders          5.454774
displacement     193.425879
horsepower       104.469388
weight          2970.424623
acceleration      15.568090
model_year        76.010050
dtype: float64
```

*Figure 3: Mean Values of each feature*

- ***Part 4:***

Box Plots were used to decide which country produces better fuel, this was indicated by the interquartile range (IQR), median, and the number of Outliers.

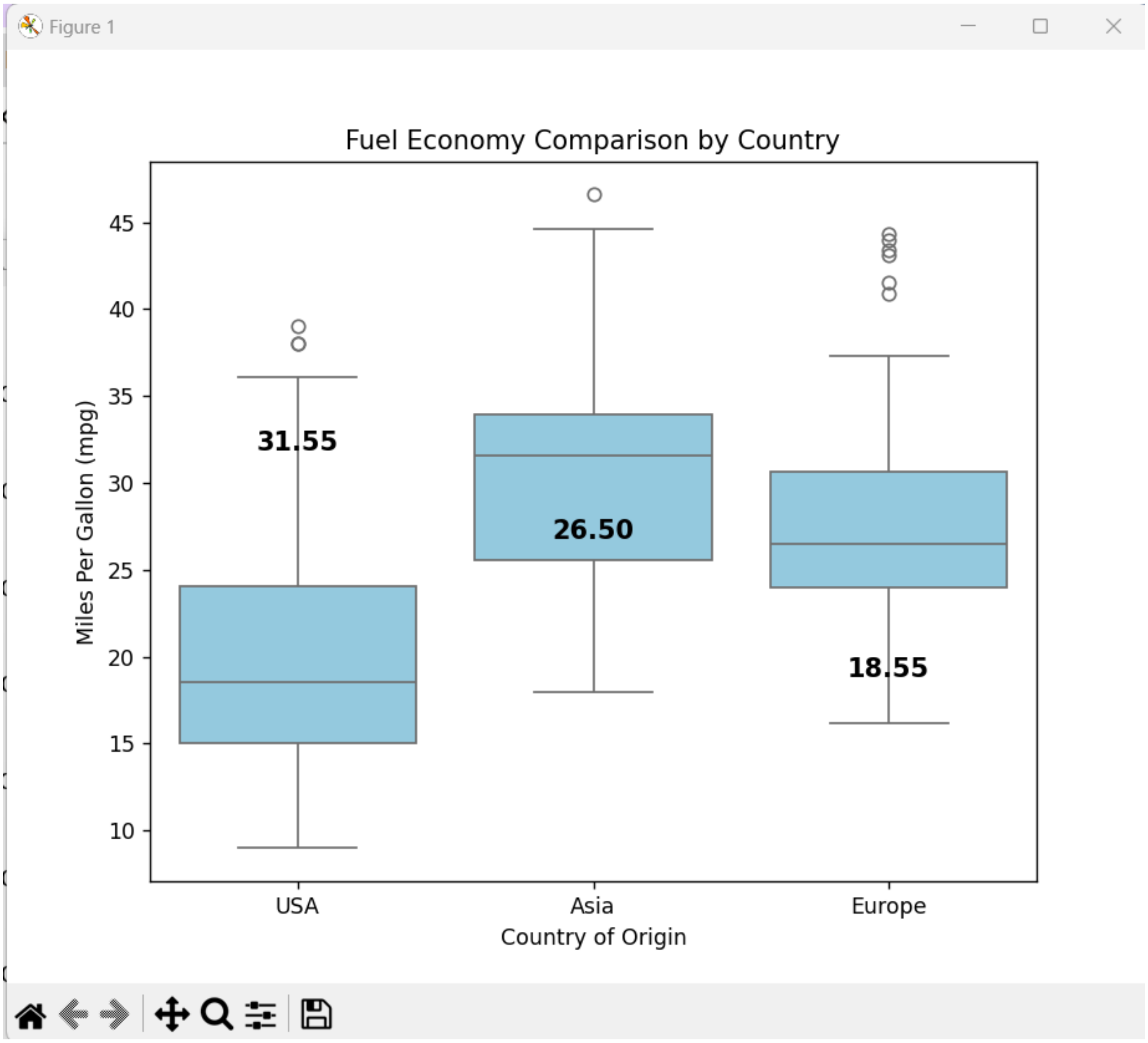
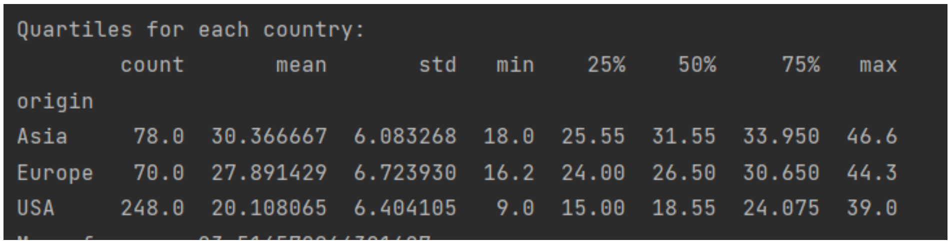


*Figure 4:Box Plots Graph*



```
Quartiles for each country:
        count       mean       std   min    25%    50%     75%   max
origin
Asia     78.0  30.366667  6.083268  18.0  25.55  31.55  33.950  46.6
Europe   70.0  27.891429  6.723930  16.2  24.00  26.50  30.650  44.3
USA     248.0  20.108065  6.404105   9.0  15.00  18.55  24.075  39.0
```

*Figure 5: Quartiles For each country*

Figure 4 reveals that the USA and Asia produce much better fuel than Europe since there are fewer Outliers and the Midian is higher than Europe. Then after seeing the calculations of the Q1 Q2 and Q3. The IQR was calculated (Q3-Q1). Asia appears to generate cars that get better average miles per gallon (MPG) since it has the highest mean MPG. The interquartile range (IQR) measures variability, and Asia has a slightly greater IQR than Europe and the United States, indicating that its MPG values are more spread out.

| | USA | ASIA | EUROPE |
|---|---|---|---|
| **IQR** | 9.075 | 8.4 | 6.65 |
| **Mean** | 20.11 | 30.37 | 27.89 |

*Table 1: Countries fuel mpg's IQR and mean*

- ***Parts 5 and 6:***

If any of the Features have a Distribution that is most similar to a Gaussian, examine for symmetry, a bell-shaped curve, and closeness between mean and median to select the characteristic with the closest Gaussian distribution. Gaussian distributions feature smooth tapering tails and a consistent spread as represented by the standard deviation. By Figure 5, the acceleration distribution is the closest to the Gaussian distribution.

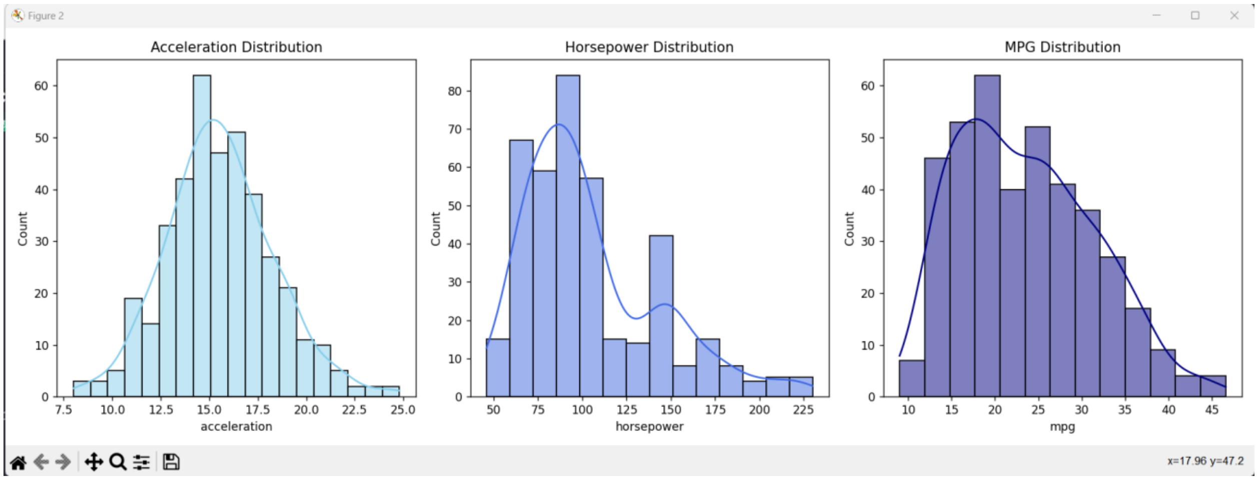
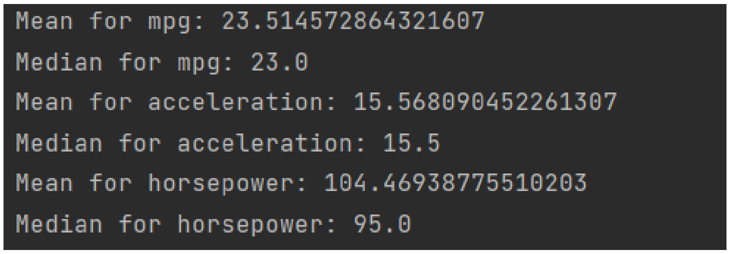


*Figure 6: Histogram graphs*

"Acceleration" is the feature that is most similar to a Gaussian distribution. The mean and median of a Gaussian distribution are very close to one other. In the case of "acceleration," the mean (15.57) and median (15.5) are reasonably close, showing symmetrical.



```
Mean for mpg: 23.514572864321607
Median for mpg: 23.0
Mean for acceleration: 15.568090452261307
Median for acceleration: 15.5
Mean for horsepower: 104.46938775510203
Median for horsepower: 95.0
```

*Figure 7: mean and median for some features*

- ***Part 7:***

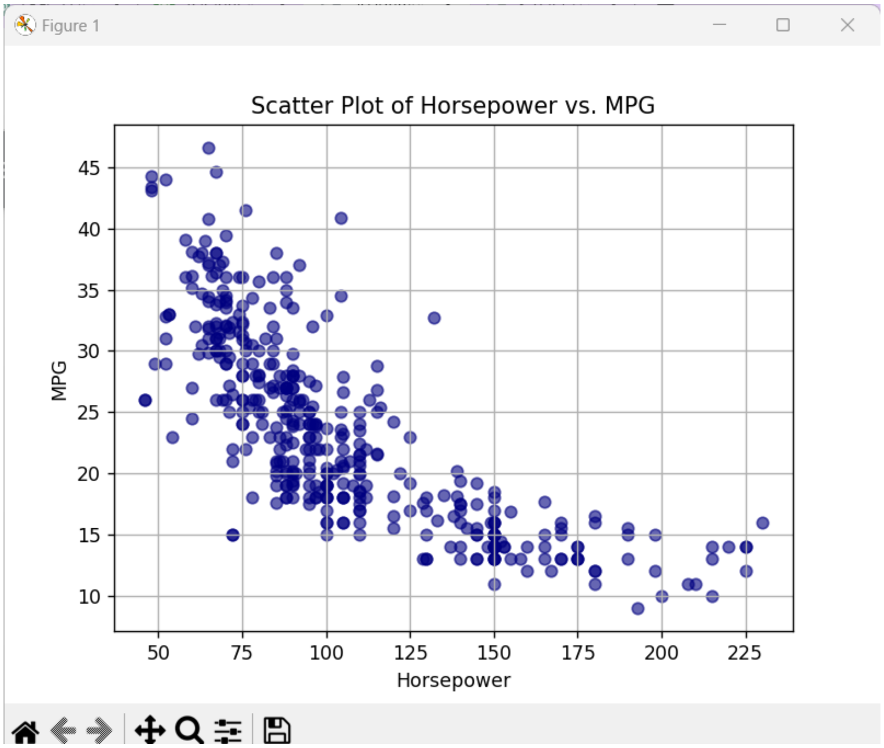
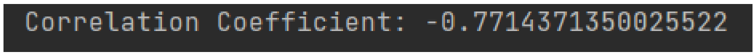


*Figure 8: Scatter plot of Horsepower vs. MPG*

The scatter plot confirms this view, that it increases from right to left, showing that there is a negative correlation between The Horsepower and the MPG, and to support this the correlation coefficient was calculated:



```
Correlation Coefficient: -0.7714371350025522
```

*Figure 9: Correlation coefficient*

The correlation coefficient is -0.7714, and because it is negative, it implies that there is a negative correlation between them.

- ***Part 8:***

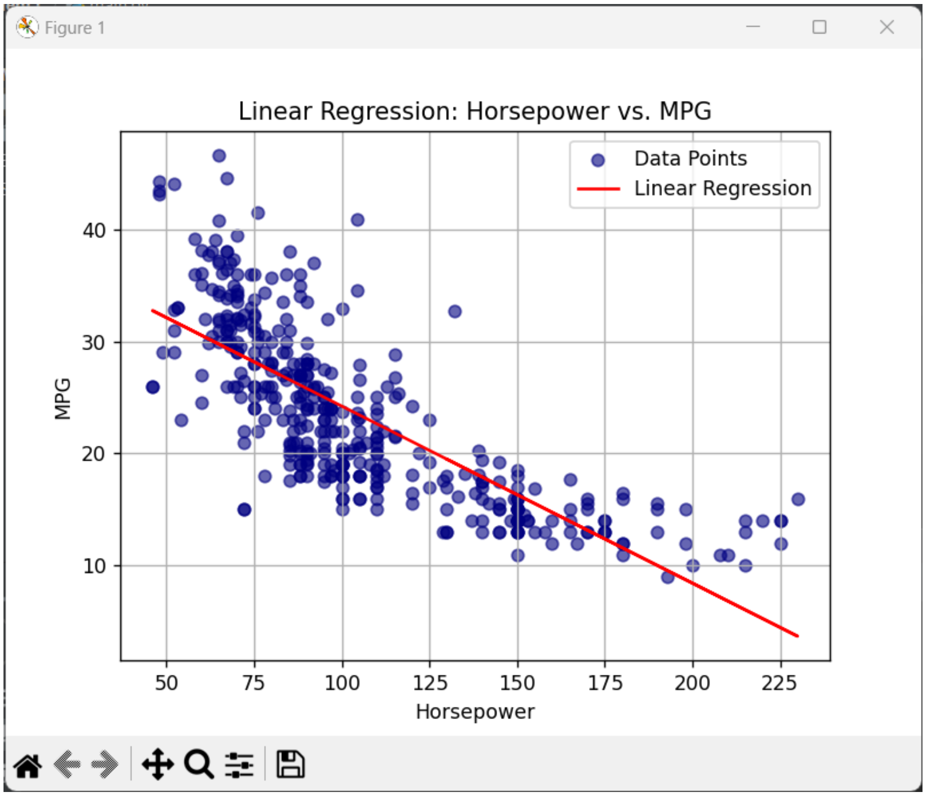


*Figure 10: Simple linear regression*

The closed-form solution for simple linear regression is given by the equation:

$$\mathbf{w} = (\mathbf{X}^{T}\,\mathbf{X})^{-1}\,\mathbf{X}^{T}\,\mathbf{y}$$

*Figure 11: linear regression solution*

w is the parameter vector, X is the matrix with the first column set to 1 (for X0=1), and y is the target variable.

- ***Part 9:***

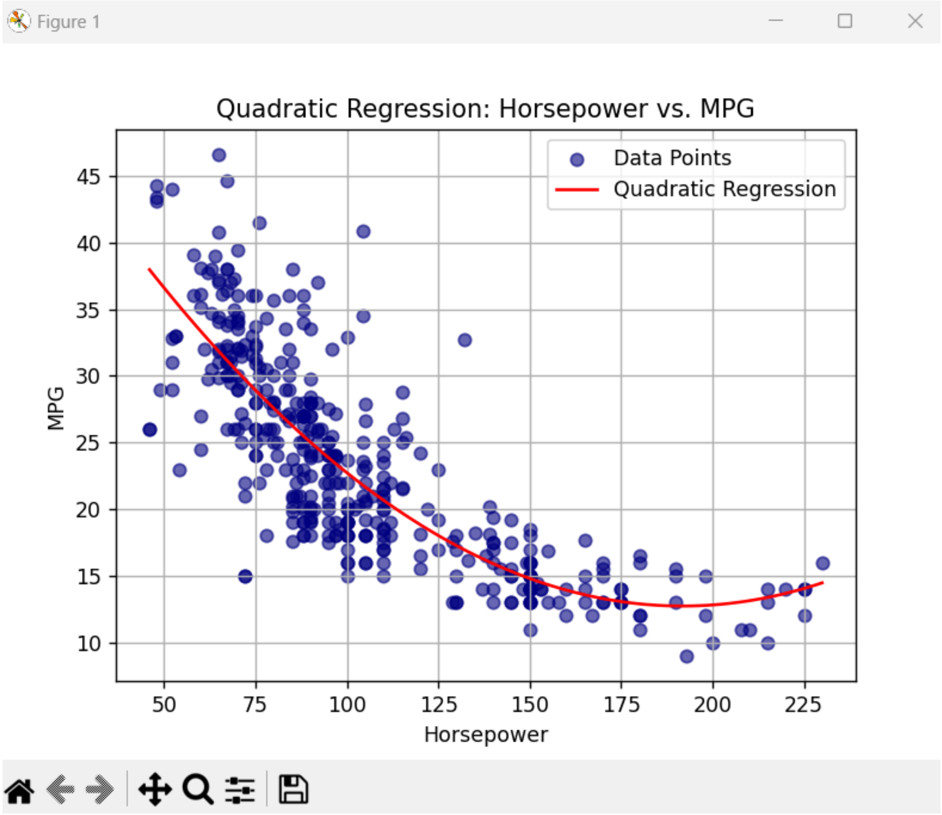


*Figure 12: Quadratic Regression*

Change the design matrix X to include both x and x^2 as features to learn a quadratic function of the form f= w0+ w1*x+ w2*x^2. The closed-form solution remains unchanged, but the design matrix has grown. Figure 12 indicates that the applied non-linear regression, which is quadratic, fits the presented data considerably better and is not underfitted like the linear regression, but is also not overfitted.

- ***Part 10:***

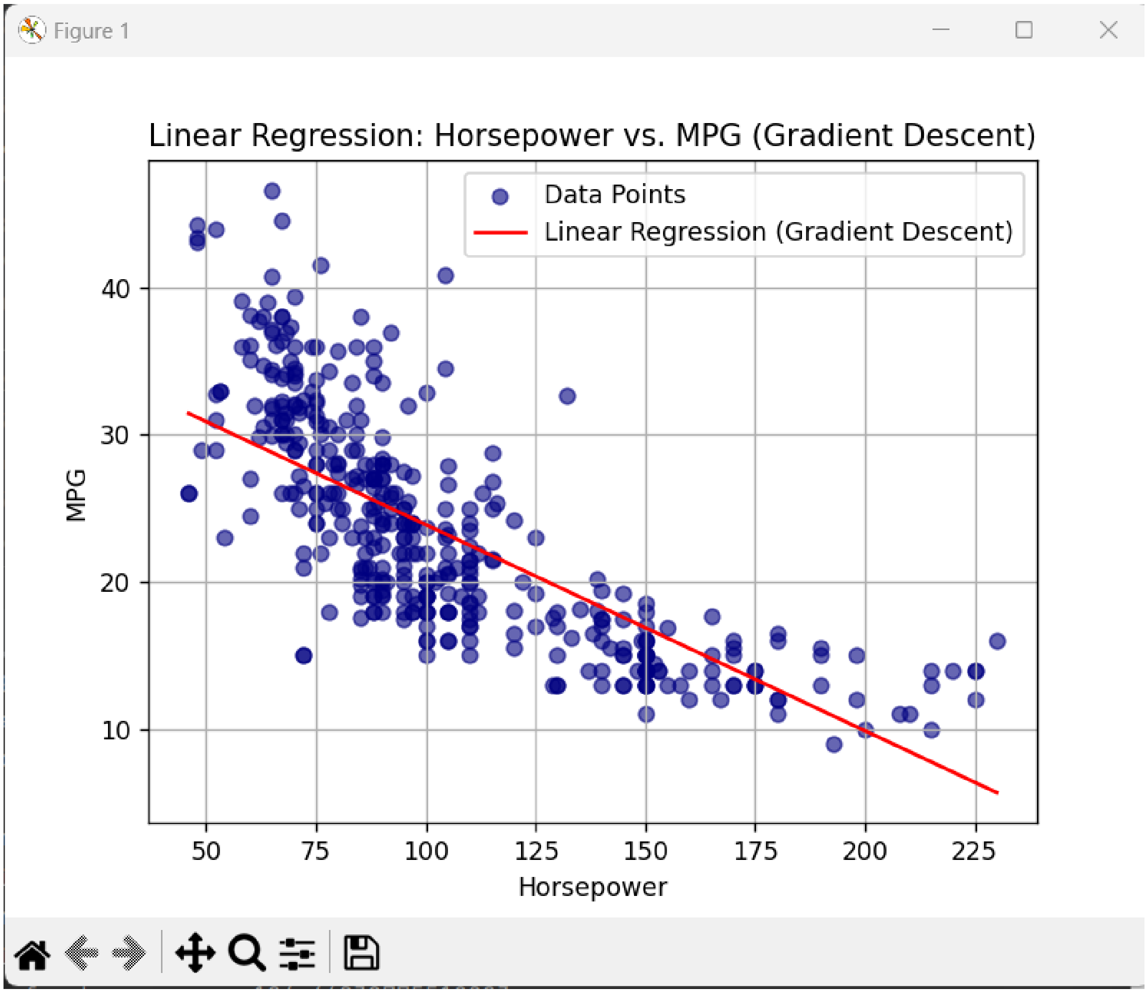


*Figure 13: linear regression by Gradient descent, not closed-form solution*

Linear Regression by the gradient Descent Algorithm depends on the learning rate, and it determines the size of the steps taken during each iteration of gradient descent. An alpha where set as 0.000025, which is quite small. This can be beneficial for convergence but may also require a larger number of iterations. Which is set to 1000000, and those two parameters affected the run time.

Closed-form linear regression gives an analytical solution by directly computing parameters, whereas gradient descent iteratively updates parameters based on the cost function slope. For smaller datasets, closed-form is more efficient, whereas gradient descent is more scalable and flexible for larger datasets and sophisticated models. The decision is influenced by processing resources, dataset size, and the requirement for real-time updates.