بسم الله الرحمن الرحيم

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

Machine Learning and Data Science (ENCS5341)

Assignment #1 Report

Prepared by : Momen Salem ID : 1200034

Instructor : Dr. Yazan Abu Farha

Section : 2

Date : 28-11-23

Question #1: Read the dataset and examine how many features and examples does it have?

```
      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]
```

After read cars.csv file using pandas, I print the data and I conclude that there are 398 examples with 8 features each.

```
RangeIndex: 398 entries, 0 to 397
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64
 7   origin        396 non-null    object
```

Also using .info() function I now know the data type of each feature.

Question #2: Are there features with missing values? How many missing values are there in each one?

Yes, from above figure .info() function give me a summary of data and I can see that there is missing values. To know these values I use the .isnull() function in pandas which give my a data frame of True and False values then loop for each value and check it is True then this value is missing and I got the following result:

```
The feature horsepower has = 6 missing values
The feature origin has = 2 missing values
```

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

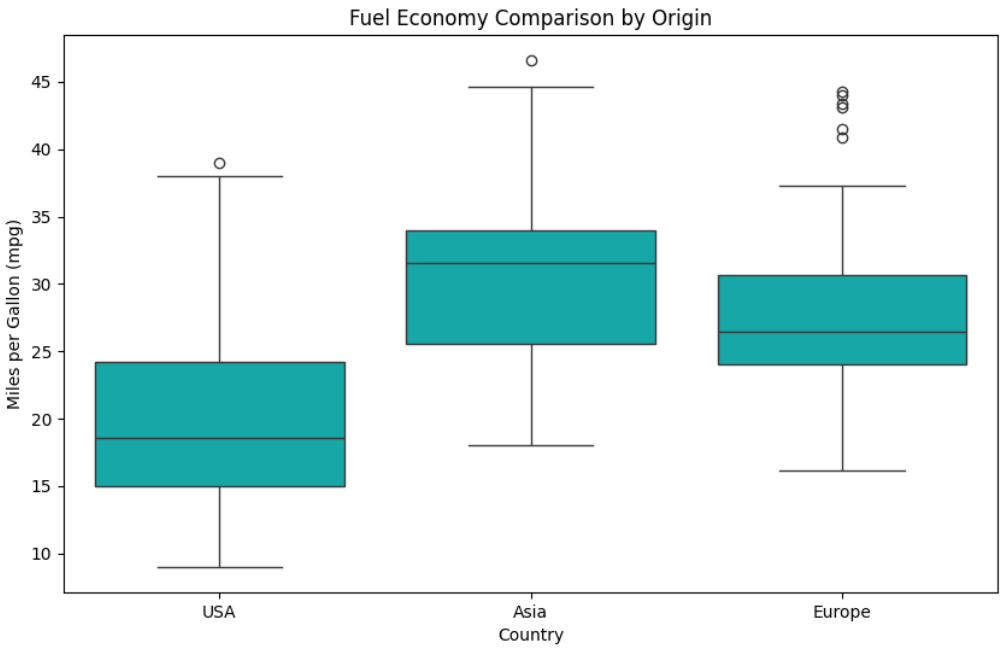
I see that I have two types with missing values one is numeric (horsepower) and the other is not numeric (origin) so I use the median to fill missing values in horsepower feature and the mode to fill the origin missing values. And to check if my work is correct I use .info() again to see if there is a missing value and I got the following result:

```
The feature horsepower has = 6 missing values
The feature origin has = 2 missing values
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    float64
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64
 7   origin        398 non-null    object
dtypes: float64(4), int64(3), object(1)
memory usage: 25.0+ KB
None
```

there is no missing value after I fill the missed values and the data is cleaned properly.

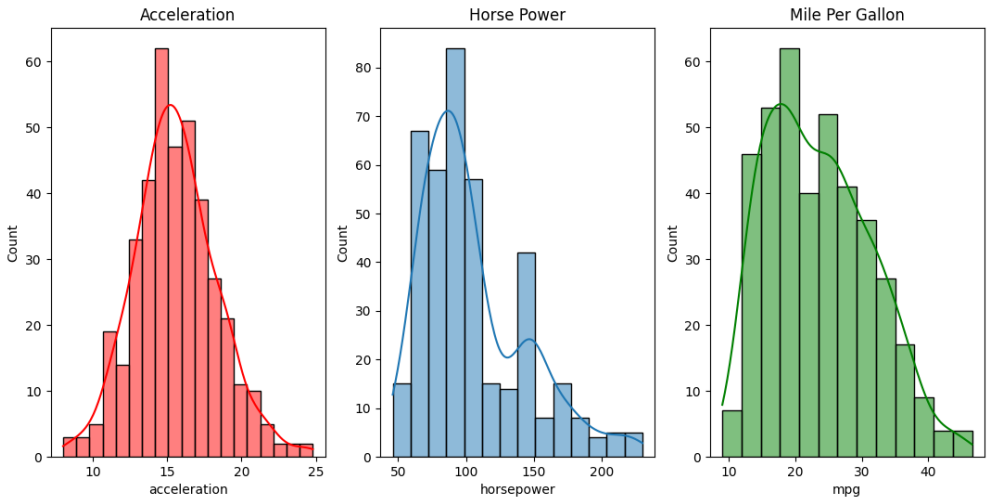
Question #4: Which country produces cars with better fuel economy?

using the seaborn library which have boxplot function that can plot what we need (the x-axis for each country and the y-axis for the mpg)



As we can see from the median of the boxes plot (the middle line of each box) that the country Asia is the highest among other boxes so the cars produced in Asia are traveling more miles per gallon (the same as we say that Asia cars are consuming less gallons per mile which means better fuel economy). Also, the upper quartile of Asia is the largest and the lower quartile is also the largest. This indicates that the country Asia is the best in fuel economy compared with USA and Europe.

Question #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.



We know that gaussian distribution is like a bell curve so the feature acceleration with red color is most likely to be a gaussian distribution (compared to the gaussian distribution plot).

Question #6: Support your answer for part 5 by using a quantitative measure ?

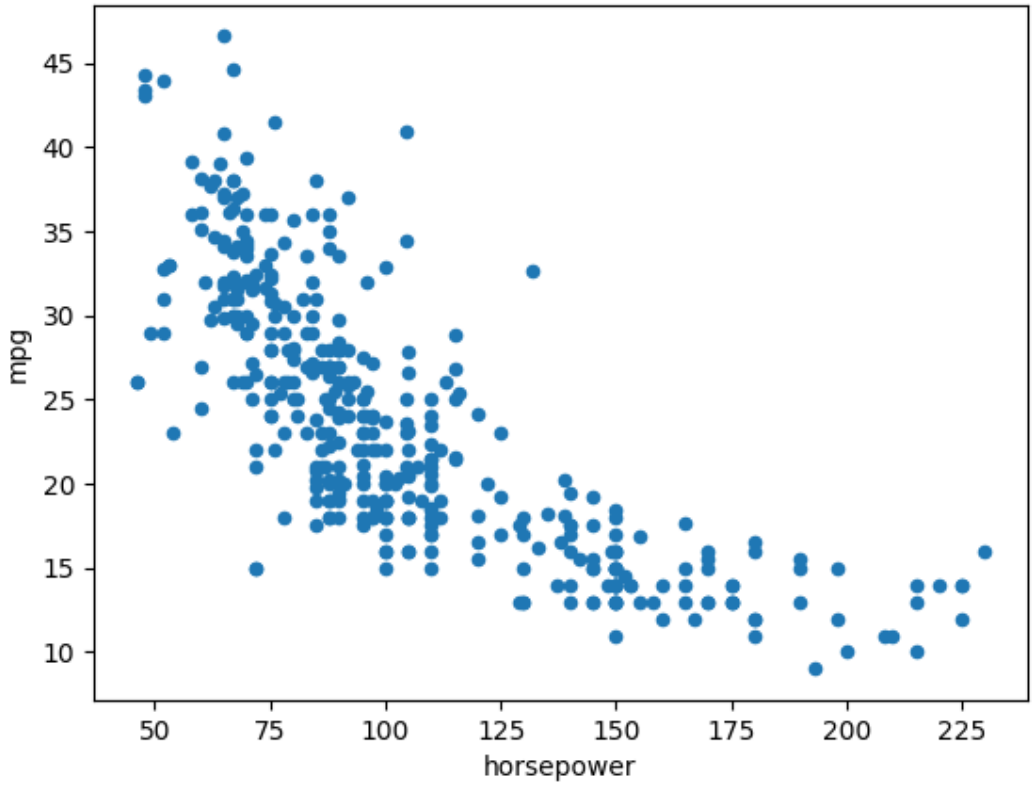
I calculate the skewness using the Pearsons second skewness coefficient (median skewness) for each feature and see which feature is near to zero and consider its distribution as gaussian distribution the result is :

```
----------------------------Question #6-------------------------------
The skewness of the feature ( acceleration ) is =  0.07407338607904465
The skewness of the feature ( horsepower ) is =  0.7436850157955579
The skewness of the feature ( mpg ) is =  0.19750789295763824
```

The feature acceleration has very small skewness so it is similar to be gaussian distribution.

The two other features have positively skewed more than the acceleration.

Question #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 scatter plot I see that there is a correlation between two features → if the horsepower increase that’s lead to decrease for mpg (negative correlation).

To verify my result I use the corr() function to check the correlation and I obtain the following result :

```
----------------------------Question #7-----------------------------
            horsepower       mpg
horsepower    1.000000 -0.771437
mpg          -0.771437  1.000000
```

My result is accepted and the correlation is negative and good one.

Question #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 ?

First I add the bias of the data (first column with ones value) and then implement the closed form solution using the matrix concept and then find the parameters as following figure :

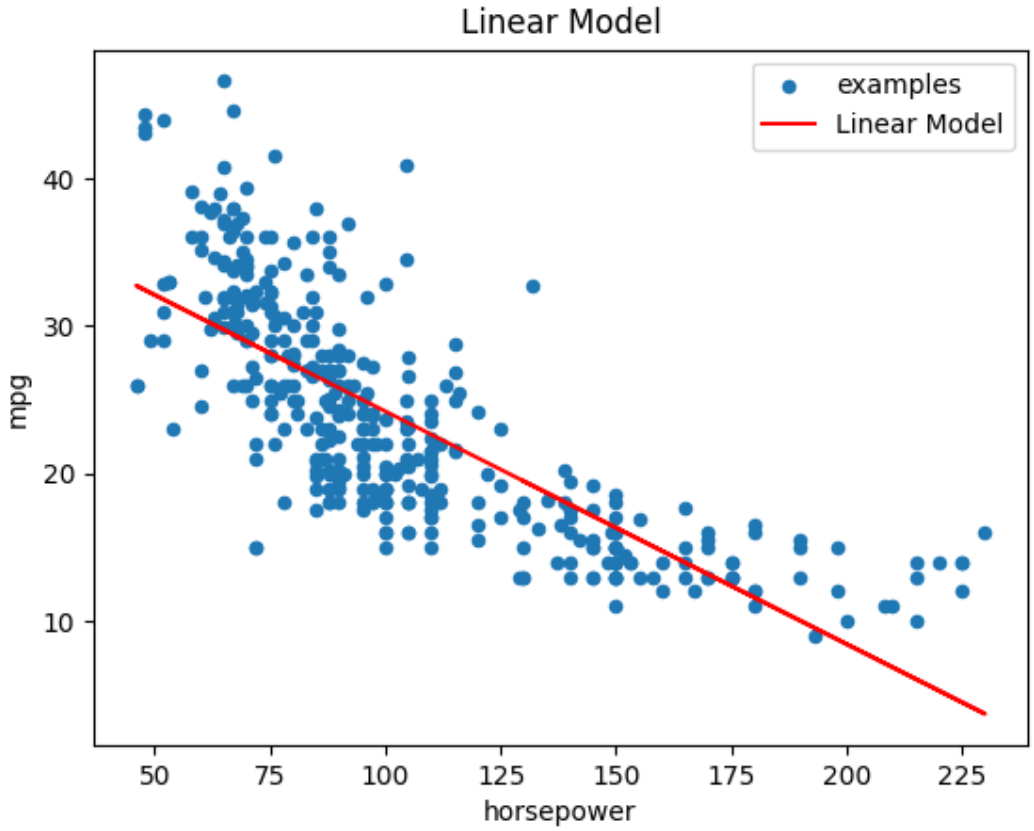
```
----------------------------Question #8-----------------------------
     x0  horsepower   mpg
0     1       130.0  18.0
1     1       165.0  15.0
2     1       150.0  18.0
3     1       150.0  16.0
4     1       140.0  17.0
..   ..         ...   ...
393   1        86.0  27.0
394   1        52.0  44.0
395   1        84.0  32.0
396   1        79.0  28.0
397   1        82.0  31.0

[398 rows x 3 columns]
w0 = 40.004515518145176  & w1 = -0.15784473335365387
```

The parameters are good because first one used to determine the y-intercept which may be perfect to equal 40 and the second parameter indicate the slope of line which has negative sign and support the conclusion that two features are negatively correlated.

After that I substitute the parameters and find the desired output (y) for each input (x)

And plot the result which is :



The result is good because the desired output must be line (linear regression) and it is lie nearest many points (the best line) with less error. The equation of finding parameters is explained carefully in code.

Question #9: Repeat part 8 but now learn a quadratic function ?

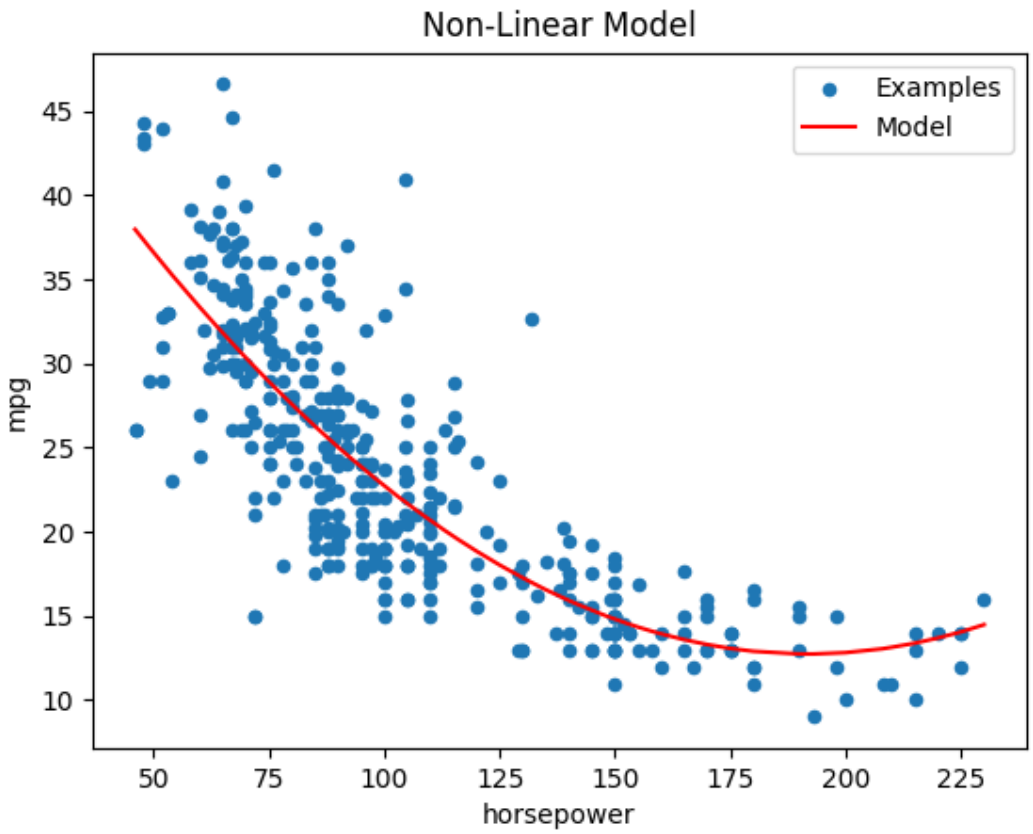
In this question I need to perform linear weighted sum of non-linear basis function (quadratic). So, I add feature to matrix which equal the squared value of horsepower and then do the same steps done in pervious question but the main difference in that the data must be sorted (on x-axis horsepower data) to have perfect curve.

```
----------------------------Question #9-----------------------------
     x0  horsepower      x^2   mpg
0     1       130.0  16900.0  18.0
1     1       165.0  27225.0  15.0
2     1       150.0  22500.0  18.0
3     1       150.0  22500.0  16.0
4     1       140.0  19600.0  17.0
..   ..         ...      ...   ...
393   1        86.0   7396.0  27.0
394   1        52.0   2704.0  44.0
395   1        84.0   7056.0  32.0
396   1        79.0   6241.0  28.0
397   1        82.0   6724.0  31.0

[398 rows x 4 columns]
[56.403522229122274, -0.45543497195874005, 0.0011876166457977936]
```

I see that there is a third parameter which means that the model in non-linear and like curve.

The parameters are order in my list (w_0, w_1, w_2)



The curve is non-linear and its result accepted more than linear line.

Question #10: Repeat part 8 (simple linear regression case) but now by implementing the gradient descent algorithm instead of the closed form solution ?

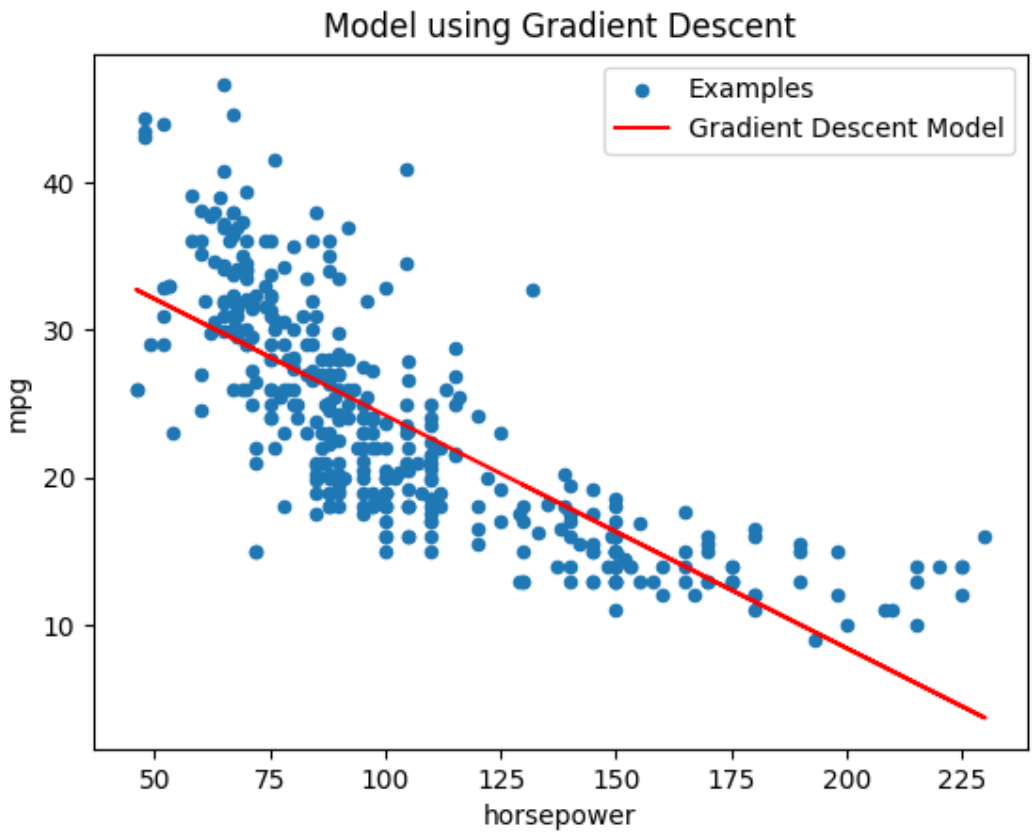
In this question I expect that the result must be the same as question 8 because the data seem to be as convex function (seen from non-linear model) which mean that if I use gradient descent I will have the optimal solution that implemented in closed form using matrices.

The parameters before and after applying gradient descent function are :

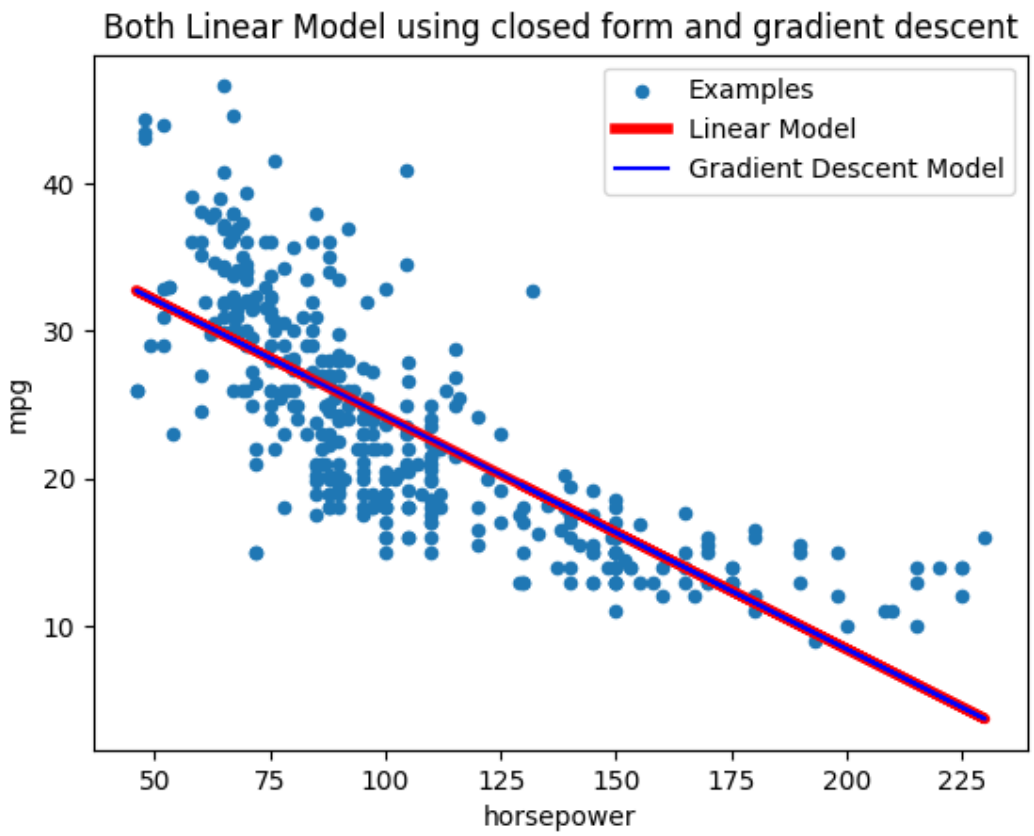
```
----------------------------Question #10-----------------------------
The initial parameters are =  0.947192279386562 0.8103217353585653
The updated parameters are =  [23.514572864321593, -6.029540545310636]
```

I see that updated parameters don't like the parameters obtained from closed form solution the reason of that is the data is normalized so the parameters will change as well as data changed.

The result after applying gradient descent function is :



To check if I have the same model from gradient descent and closed form solution I put the two lines on the same plot but with increase the width of one line to see if they are match.



The result is good and accepted because both models are the same and the gradient descent give me the minimum loss function (the line with minimum mean squared error).