****

**BIRZIET UNIVERSITY**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**Machine Learning and Data Science**

**Exploratory Data Analysis and Regression Analysis on Cars Dataset**

**Prepared by**

**Rama Abdlrahman 1191344**

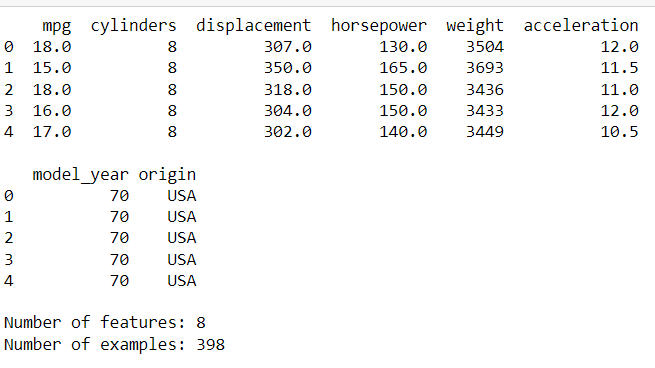
**Supervised by**

**Dr. Yazan Abu Farha**

**BIRZEIT**

**November – 2023**

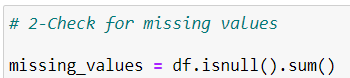
# Dataset Overview

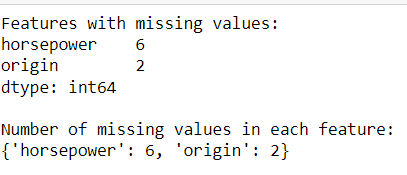


The dataset includes information on different automotive features, such as mpg, number of cylinders, engine displacement, horsepower, vehicle weight, acceleration, model year, and origin. There are a total of 398 entries in the dataset, each representing a unique car model with 8 features. All values in the dataset are of numeric type, except for the 'origin' feature which is categorical and represents the origin of the car (e.g., USA, Asia, Europe).

# Handling Missing Values

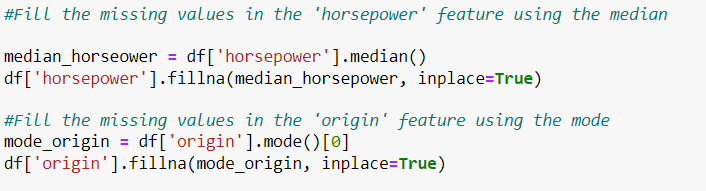
## **Identification**:

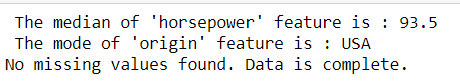




## Imputation:

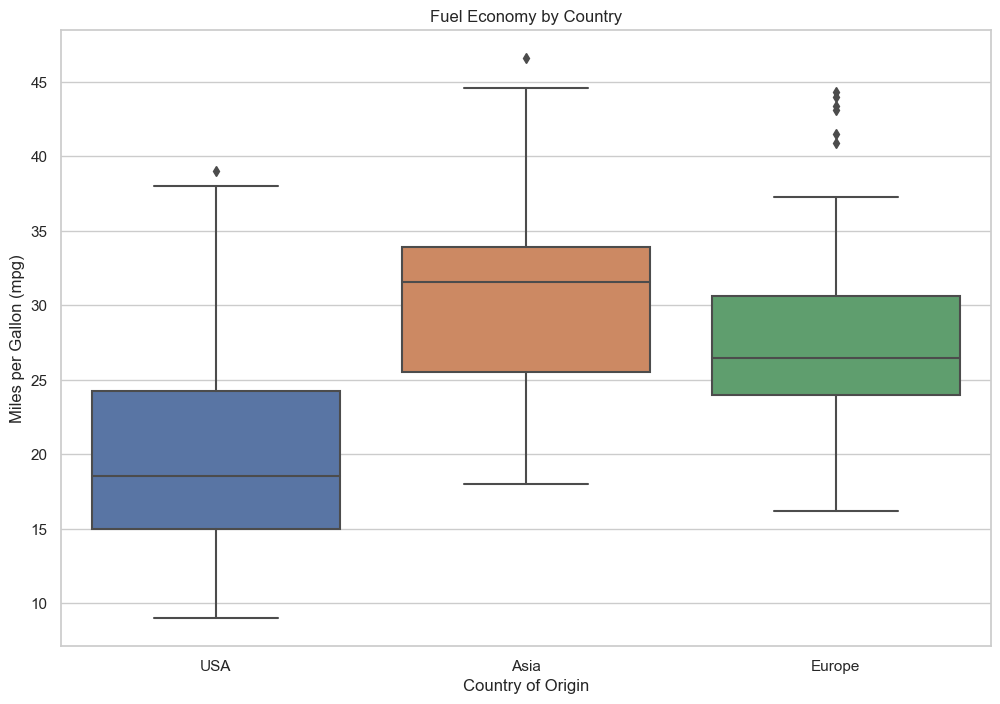
The missing values in the 'horsepower' feature will be filled using the median, as it is a numeric data type, and median is better for numeric data. Similarly, the missing values in the 'origin' feature will be filled using the mode, as it is a categorical data type. After filling the missing values, we check if there are any remaining missing values in the dataset.

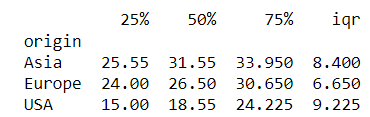


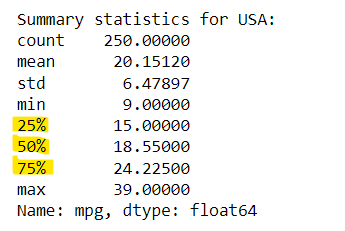


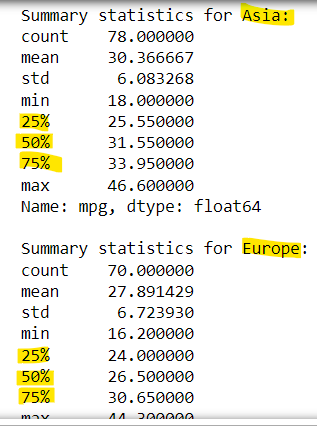
## Fuel Economy by Country

**Box Plot:**









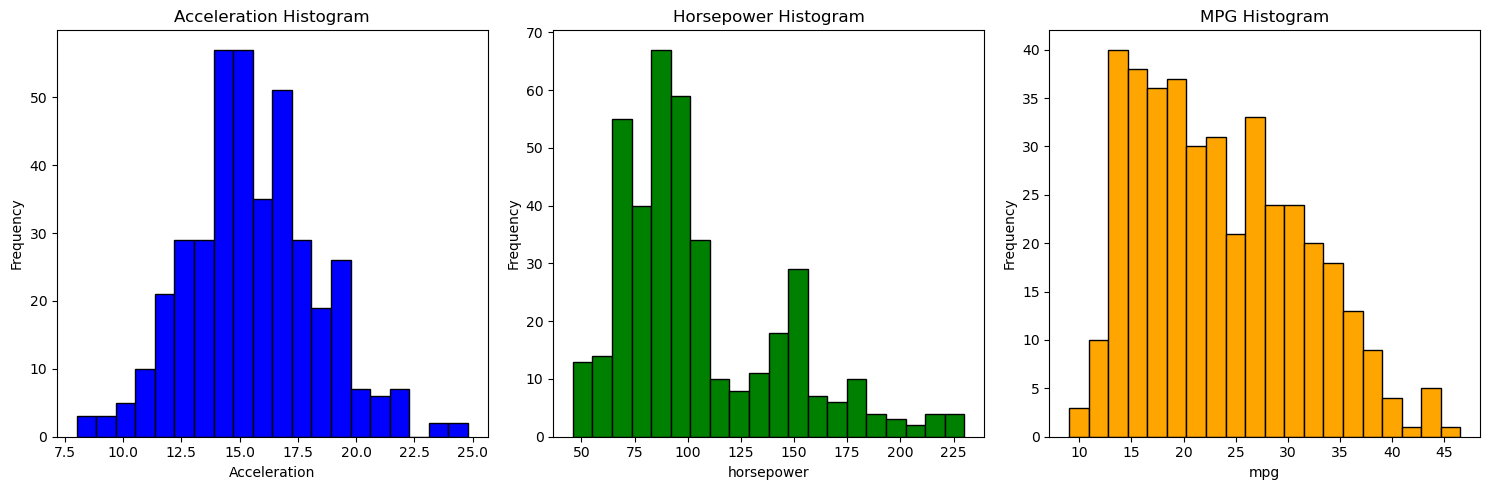
The box plot shows the distribution of fuel economy by country. The center line of each box represents the median fuel economy, while the edges of the box represent the 25th and 75th percentiles. The whiskers extend to the most extreme data points that are not outliers.

The box plot shows that Asia has the highest median fuel economy(produces cars with better fuel economy), followed by Europe and the USA. The USA also has the widest IQR, indicating that there is more variability in fuel economy among vehicles in the USA than in Asia or Europe.

The summary statistics show that the mean fuel economy is slightly higher than the median fuel economy in all three countries. This is because the mean is influenced by the presence of outliers, while the median is not. The standard deviation is a measure of the variability in fuel economy. The standard deviation is highest in the USA, followed by Europe and Asia.

## Feature Distributions

**Histograms:**



Based on the histograms, the distribution of acceleration is most similar to a Gaussian. It is bell-shaped and symmetrical, with a single peak in the center. The histograms of horsepower and mpg are both skewed, with longer tails on one side.

Acceleration is affected by a number of factors, including engine power, weight, and aerodynamics. These factors are all distributed in a normal way, which would result in a normal distribution of acceleration.

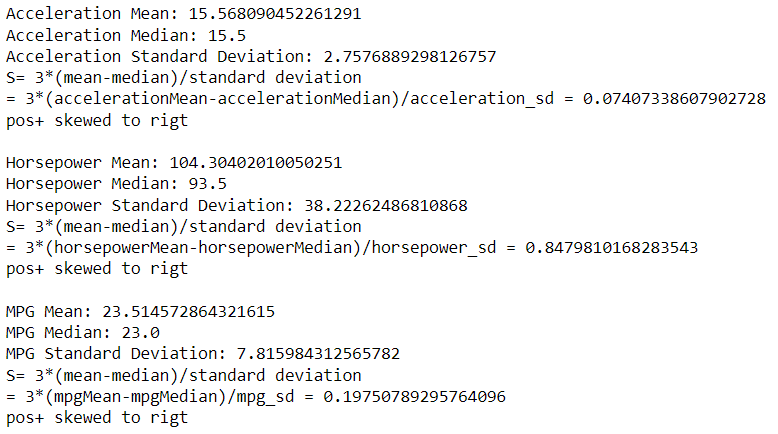
## Gaussian Distribution Analysis: Quantitative Measure:

**(Using the first and Second Coefficient of Skewness)**

1. **Pearson's Second Coefficient of Skewness**

Skewness is a measurement that indicates the distortion or asymmetry in a data set's distribution. When skewness is close to 0, it suggests that the distribution is close to normal. A negative skewness indicates that the distribution is skewed to the left, while a positive skewness indicates that the distribution is skewed to the right.

To calculate Pearson's second skewness coefficient, we need to find the mean, median, and standard deviation for the three features: acceleration, horsepower, and mpg. The coefficient is calculated by multiplying the difference between the mean and median by three, and then dividing it by the standard deviation. Applying this equation will provide us with the skewness coefficient for each feature.



**Explaining based on results:**

The acceleration feature has a skewness coefficient of 0.0741, which is close to zero (close to normal distribution) and indicates a slightly right-skewed distribution. The horsepower feature has a skewness coefficient of 0.8480, indicating a positive skew and a right-skewed distribution. Lastly, the mpg feature has a skewness coefficient of 0.1975, indicating a mildly positive skew and a right-skewed distribution.

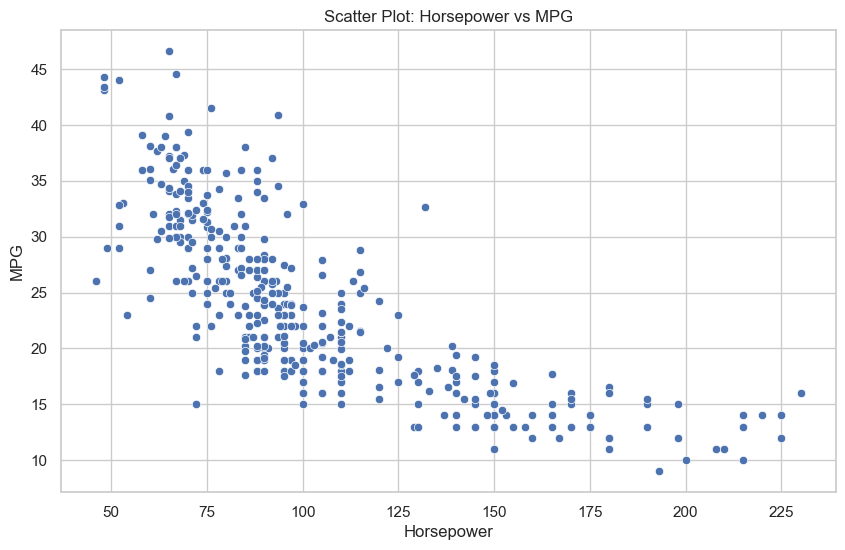
1. **Using ready method .skew() (Which use the first coefficient of skewness)**

Among the three variables, acceleration has a skewness coefficient of 0.28, indicating a slight rightward skewness. This suggests that the distribution of acceleration is the closest to a normal distribution compared to the other variables.

On the other hand, the horsepower variable has the highest positive skewness coefficient of 1.11, indicating a significant rightward skewness. This means that the distribution of horsepower is heavily skewed to the right.

Lastly, the mpg variable has a skewness coefficient of 0.46, indicating a moderate rightward skewness. This suggests that the distribution of mpg is moderately skewed to the right.

## Horsepower vs MPG

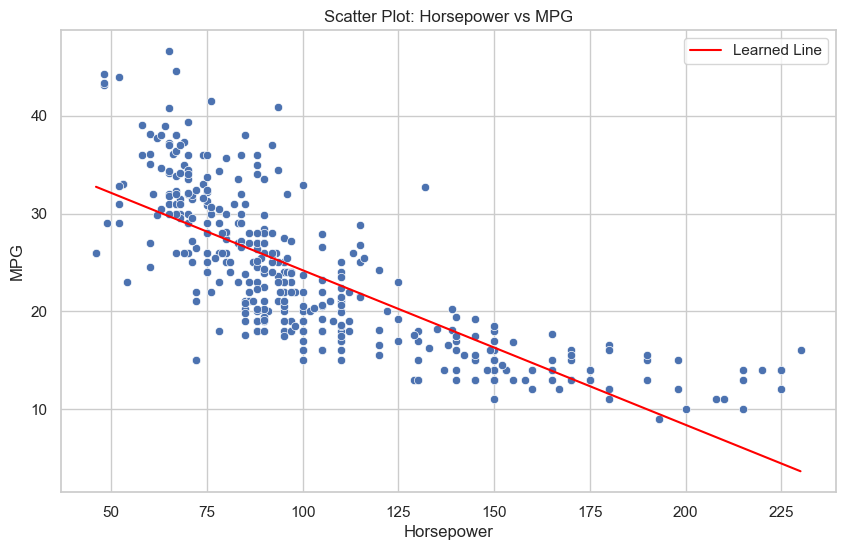


The correlation coefficient is a measure that ranges from -1 to 1. A value close to 1 indicates a strong positive correlation, while a value close to -1 indicates a strong negative correlation. If the value is close to 0, it means there may not be a significant linear correlation.

In the given scenario, the correlation coefficient between horsepower and MPG is -0.7734532045742155, which is close to -1, indicating a strong negative correlation. However, the scatter plot also shows that there is a lot of variation in MPG for cars with the same horsepower, which suggests that there are other factors besides horsepower that affect MPG.

## Linear Regression

**Visualization:**



The code calculates the closed-form solution using the normal equation (theta = (X^T X)^(-1) X^T y) and then extracts the slope and intercept from the resulting theta vector.

The learned line can be used to predict the MPG of a car based on its horsepower. We can notice that Linear regression may not be efficient for this scatter plot because the data points may not follow a linear trend, indicating that a linear model may not accurately represent the relationship between the variables. And that’s because linear regression is a simple model.

## Quadratic Regression (polynomial regression model with a degree of 2)

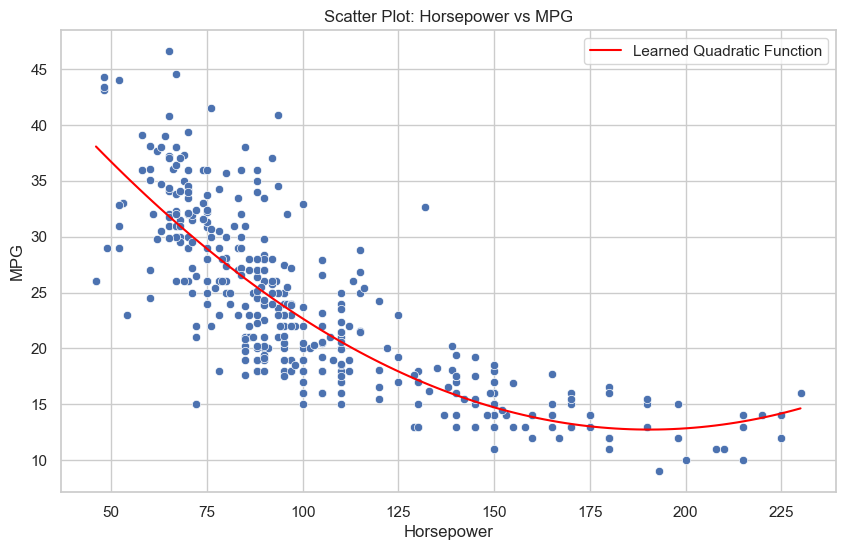
**METHOD 1:**

Automated approach is taken using the Polynomial Features class from scikit-learn. This class generates polynomial features up to a specified degree, in this case, degree 2. It automatically adds the quadratic term, and then a linear regression model is trained on the extended feature matrix.

**METHOD 2:**

Quadratic term is manually added to the feature matrix. This method explicitly calculates the coefficients using the normal equation.

**Both methods gave same results:**



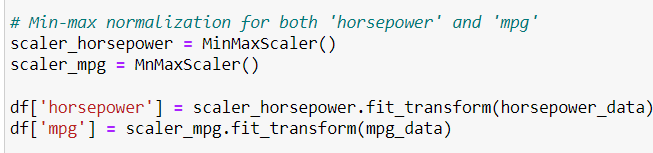
The learning line of the quadratic plot is the best-fit quadratic function for the data. It is the function that minimizes the sum of the squared errors between the actual MPG values and the predicted MPG values.

The quadratic plot shows that the relationship between horsepower and MPG is quadratic, meaning that it is curved. The learning line captures this curvature and provides a more accurate prediction of MPG than a linear regression model.

**Gradient Descent**

1. **min-max normalization**

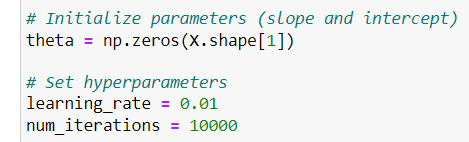
For both 'horsepower' and 'mpg' I used the MinMaxScaler class from scikit-learn. This normalization scales the data to a range between 0 and 1, making the features more comparable and improving the convergence of the optimization algorithm. Normalization helps the optimization algorithm converge faster, especially when features are on different scales.



1. **Linear Regression with Gradient Descent**

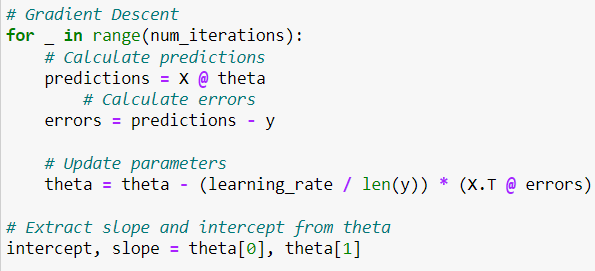
Implements linear regression using the gradient descent algorithm to estimate the model parameters (slope and intercept). Gradient descent aims to minimize a loss function by adjusting the model parameters in the direction of the negative gradient.

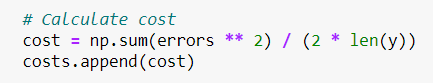
The code first initializes the parameters (theta) to zeros and sets the hyperparameters (learning\_rate and num\_iterations).



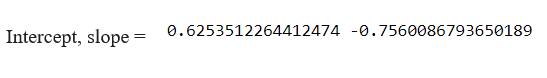
**steps:**

1. The code predicts the MPG values for each horsepower value using the current parameter estimates.
2. The code calculates the errors between the predicted MPG values and the actual MPG values.
3. The code updates the parameter estimates (theta) by moving in the opposite direction of the gradient of the loss function. The gradient is calculated as the average of the error derivatives with respect to each parameter.



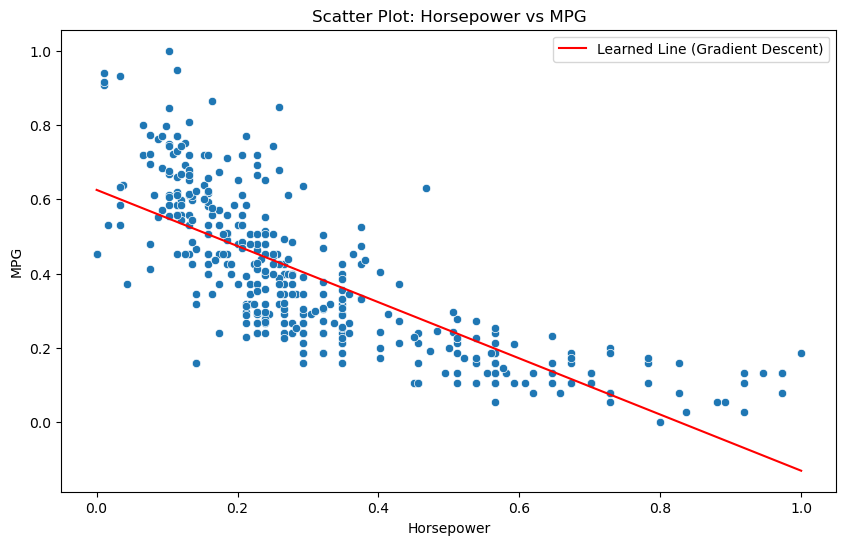


After the gradint descent loop completes, the code extracts the final slope and intercept from the updated theta vector:



**Visualizing the Results**

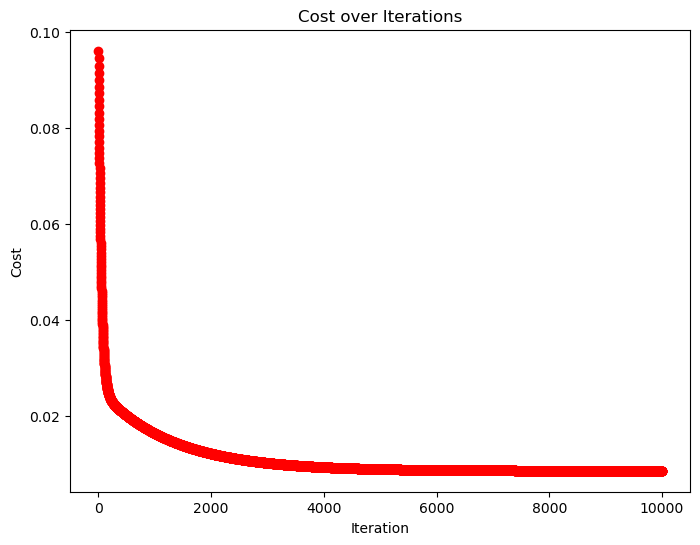
The code creates a scatter plot showing the relationship between horsepower and MPG. It also plots the learned regression line, which represents the predicted MPG values for each horsepower value. shows how the gradient descent algorithm works to find the best fit line for a scatter plot.

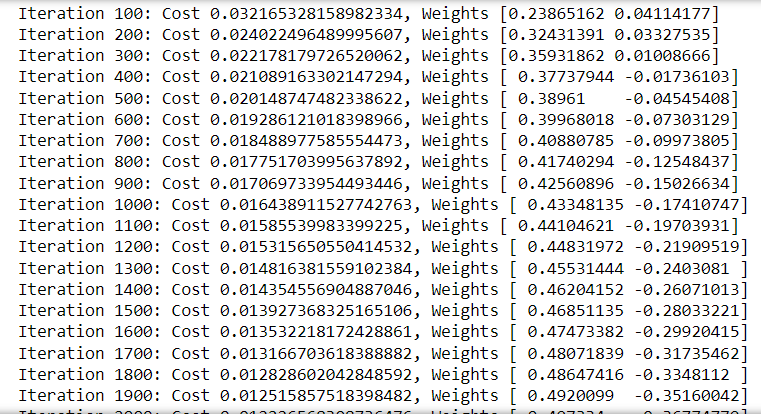


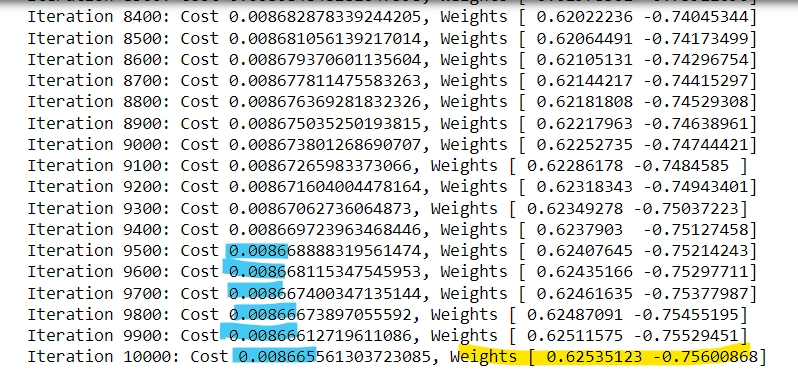
**Additional**

**Learning Curve, Cost vs Iterations**

Additional part that demonstrates the learning curve, which shows how the cost (mean squared error) decreases over iterations of the gradient descent algorithm.



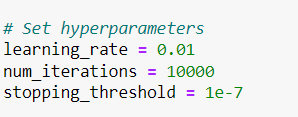




The initial cost is 0.03216, and it gradually decreases to around 0.00866 after 10,000 iterations. This decreasing trend is a good sign, as it suggests that the algorithm is moving towards a minimum.The weights are being updated at each iteration. Initially, the weights are [0.23865162 0.04114177], and after 10,000 iterations, they become [ 0.62535123 -0.75600868]. These changes in weights are the result of the optimization process, where the model adjusts its parameters to minimize the cost function.

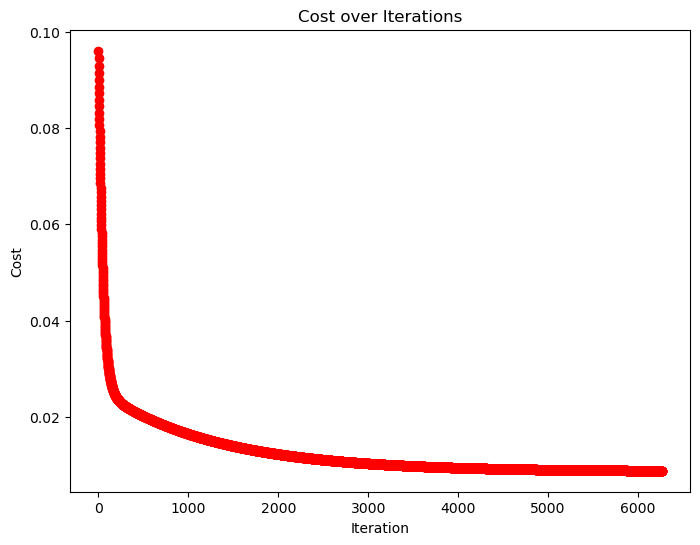
The fact that the cost is consistently decreasing and the weights are stabilizing indicates that the algorithm is converging. The final weights are the estimated optimal weights for the this problem. The success of gradient descent heavily depends on the learning rate. If the learning rate is too small, the algorithm may take a long time to converge, and if it's too large, it might overshoot the minimum. I chose it to be 0.01.

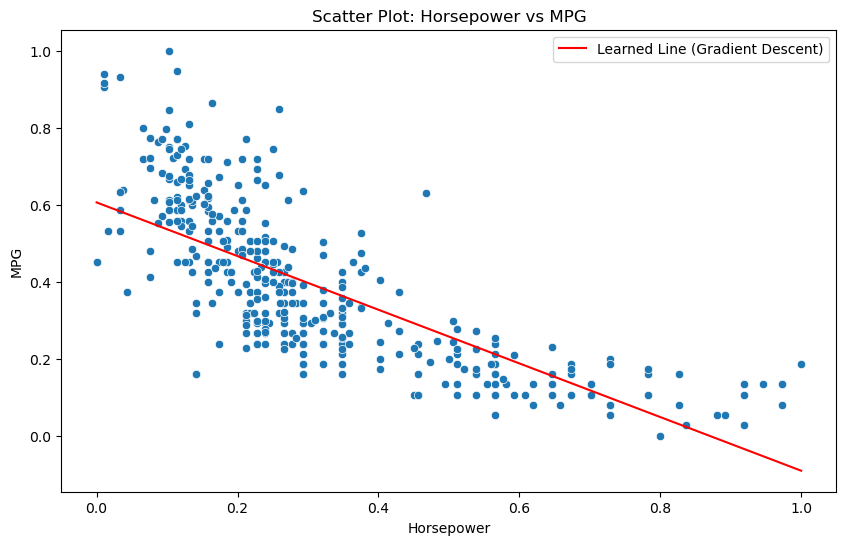
**The version of the code with the stopping threshold condition:**



Stopped at:

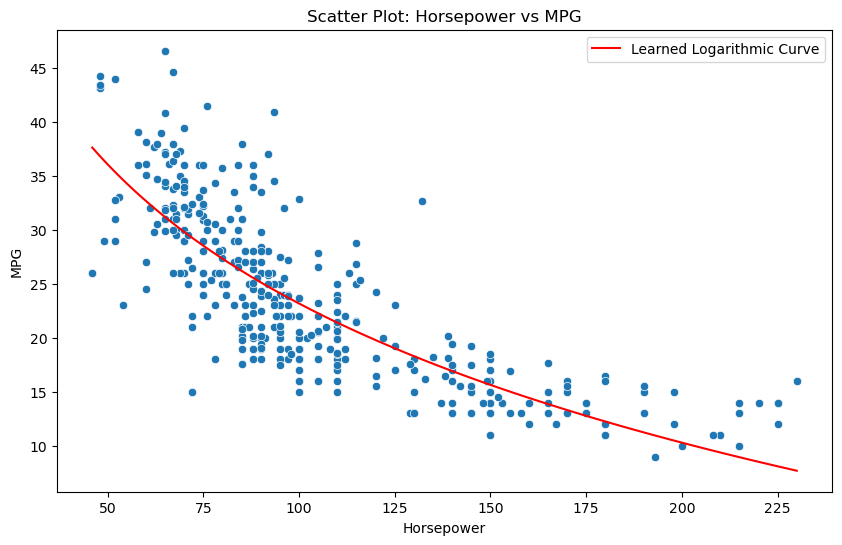






**Additional Part: Tried to apply logarithmic transformation**

Here, the 'horsepower' feature is transformed using the natural logarithm (np.log). The linear regression is then performed on the transformed feature. Applying a logarithmic transformation does not change the nature of the regression as a linear model. It's a common technique used to capture non-linear relationships within the framework of linear regression.



The logarithmic line in the scatter plot shows that the relationship between horsepower and MPG is logarithmic, meaning that MPG changes exponentially in response to changes in horsepower. This means that small changes in horsepower can lead to large changes in MPG. As seen the logarithmic line provides a good fit for the data and can be used to accurately predict the MPG of a car based on its horsepower.

## Code Availability

The complete Python code is available in the submitted Jupyter Notebook (.ipynb).

[http://localhost:8888/notebooks/OneDrive/Assignment1/Assignment1\_1191344\_Rama.ipynb#](http://localhost:8888/notebooks/OneDrive/Assignment1/Assignment1_1191344_Rama.ipynb)