# Department of Computing

**CS 471: Machine Learning**

**Submitted By: Zainab Anwaar**

**BESE: 11A**

**Lab 10: Feature Engineering**

**Date: 14 April 2023**

**Time: 02:00 PM-05:00 PM**

**Instructor: Dr. Seemab Latif**

**Lab 10: Feature Engineering**

**Tasks:**

In this lab, we will be using a real-world dataset (Automobile Data Set <https://archive.ics.uci.edu/ml/datasets/automobile>) to apply feature engineering techniques. The tasks involved are:

1. **Import the necessary libraries and load the dataset**

Libraries such as pandas, numpy, seaborn, sklearn.linear\_model, metrics, preprocessing, matplotlib.pyplot were imported first. After that a downloaded csv file from the given url was read through the pd.read\_csv() method.

1. **Perform data cleaning and handle missing values, duplicates, and outliers**

By using the df.info() method, the datatype and number of null values was first checked. The dataset contained “?” in place of null values. First those were corrected by replacing it with null. Each multiclass field was given numerical type through the method:

dataset[column]=dataset[column].astype('category').cat.codes

Other methods with object datatype were converted through:

dataset[col]=dataset[col].astype(float)

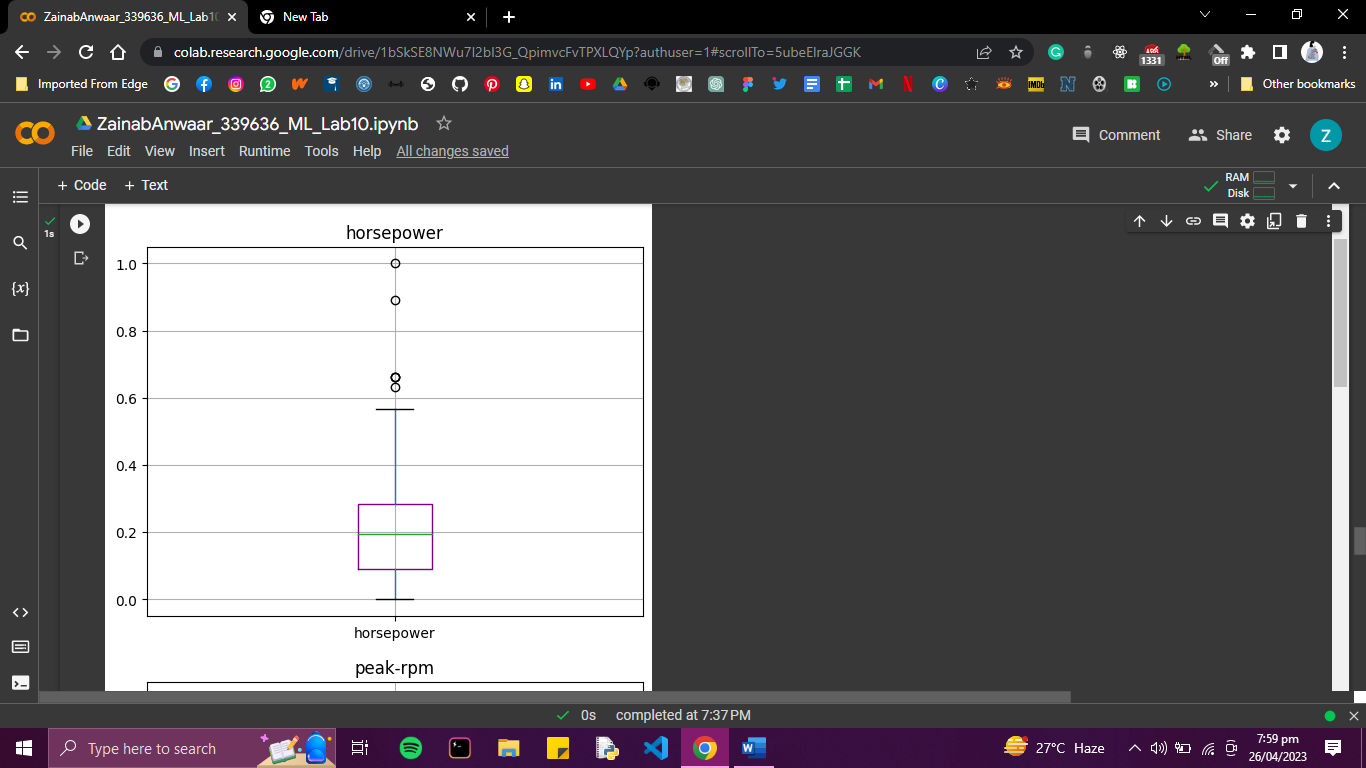
Imputed the missing values with the mean of each column using

mean=dataset[col].mean()

  dataset[col].fillna(mean, inplace=True)

After imputing the dataset, boxplots of the continuous data showed the outliers.

e.g. for horsepower:

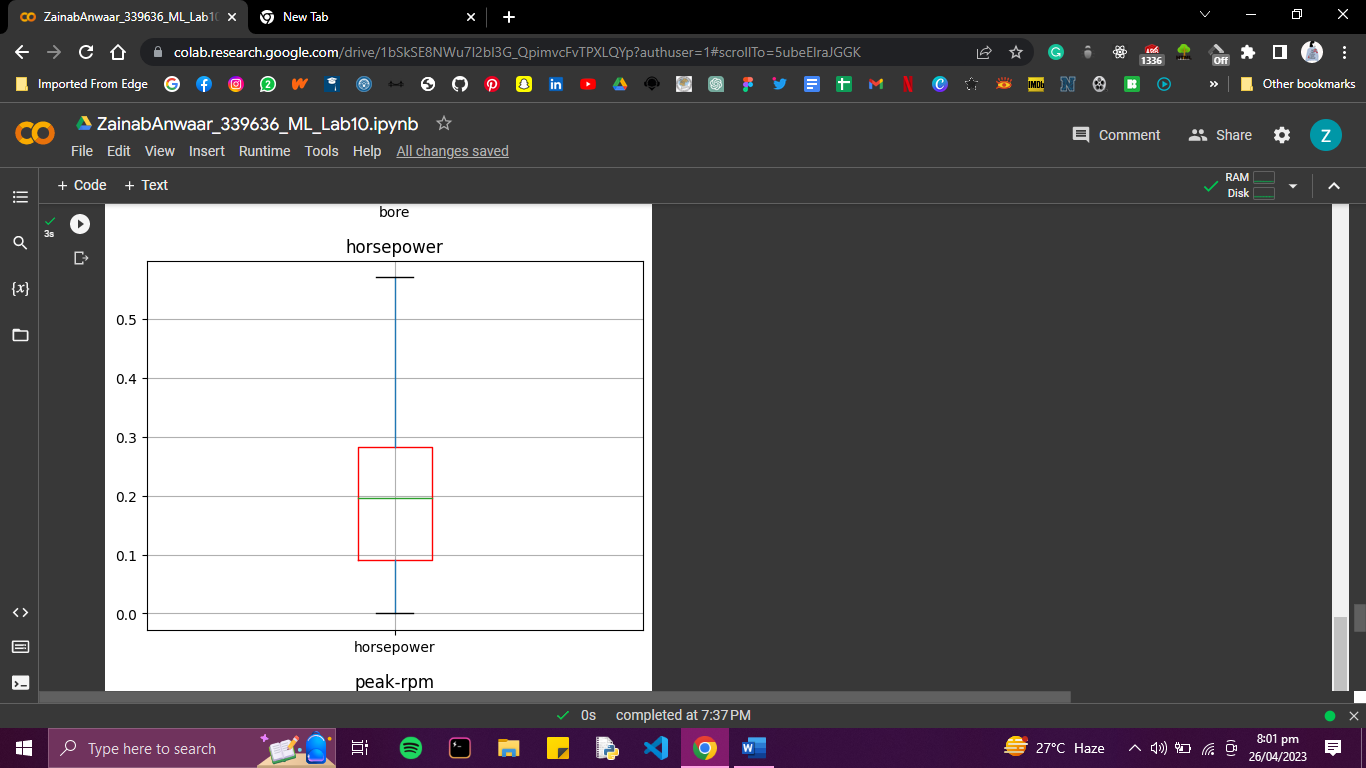


Moreover, skewness of each fields was checked

print(column,":",dataset[column].skew())

and corrected by using IQR method, which removed the outliers in the columns.

After capping the outliers:



1. **Perform feature selection and remove the irrelevant features**

By using correlation matrix, features with low correlation could be seen. Best features were those with more than 0.1 absolute correlation.

best\_features=corr\_matrix.index[abs(corr\_matrix['price'])>0.1]

rest of the features were dropped

dataset = dataset[best\_features]

1. **Perform feature scaling and transform the data features to a similar scale**

Using MinMaxScalar(), the features were scaled

scaler = MinMaxScaler()

numeric\_cols = ['normalized-losses','wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-size', 'bore', 'stroke', 'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg', 'highway-mpg', 'price']

dataset[numeric\_cols] = scaler.fit\_transform(dataset[numeric\_cols])

1. **Perform feature transformation and create new features by combining, extracting, or transforming existing features**

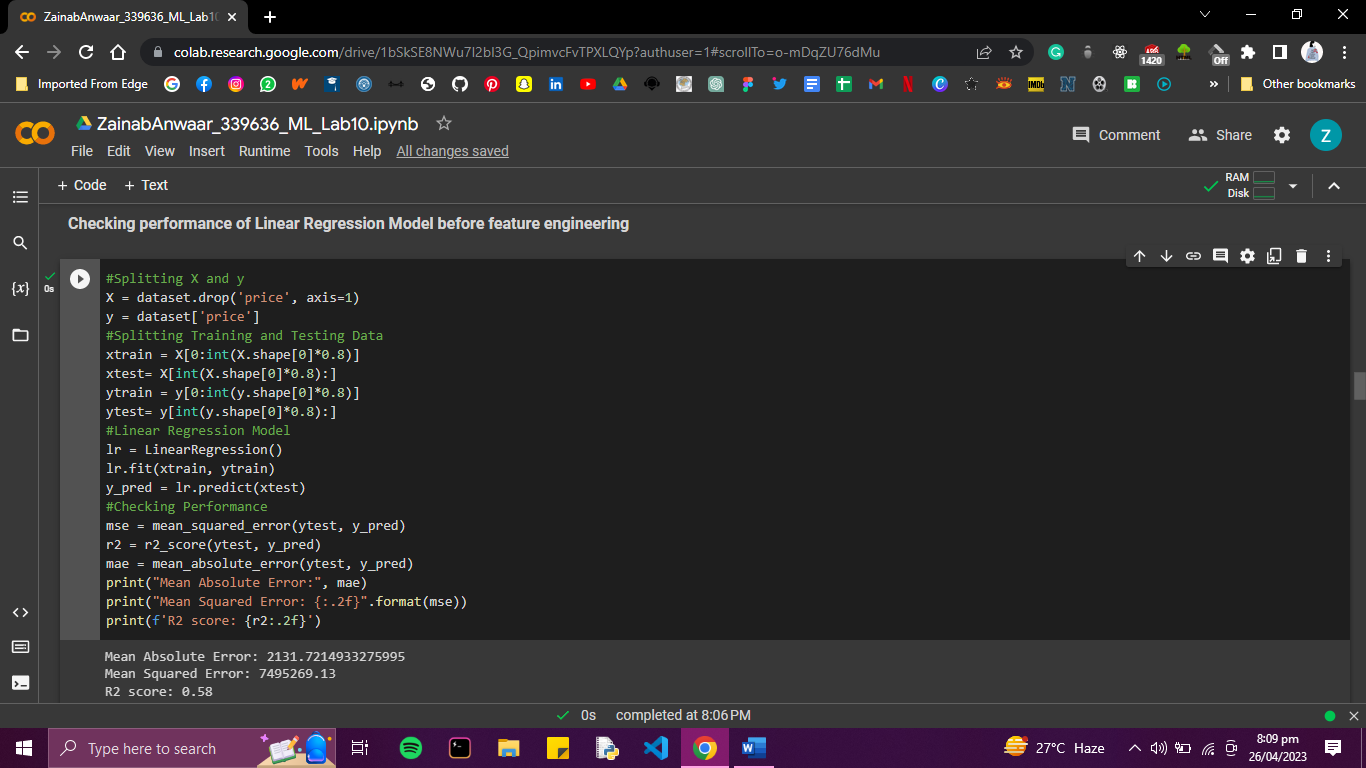
New feature power-to-weight is extracted from the horsepower and curb weight

dataset['power-to-weight'] = dataset['horsepower'] / dataset['curb-weight']

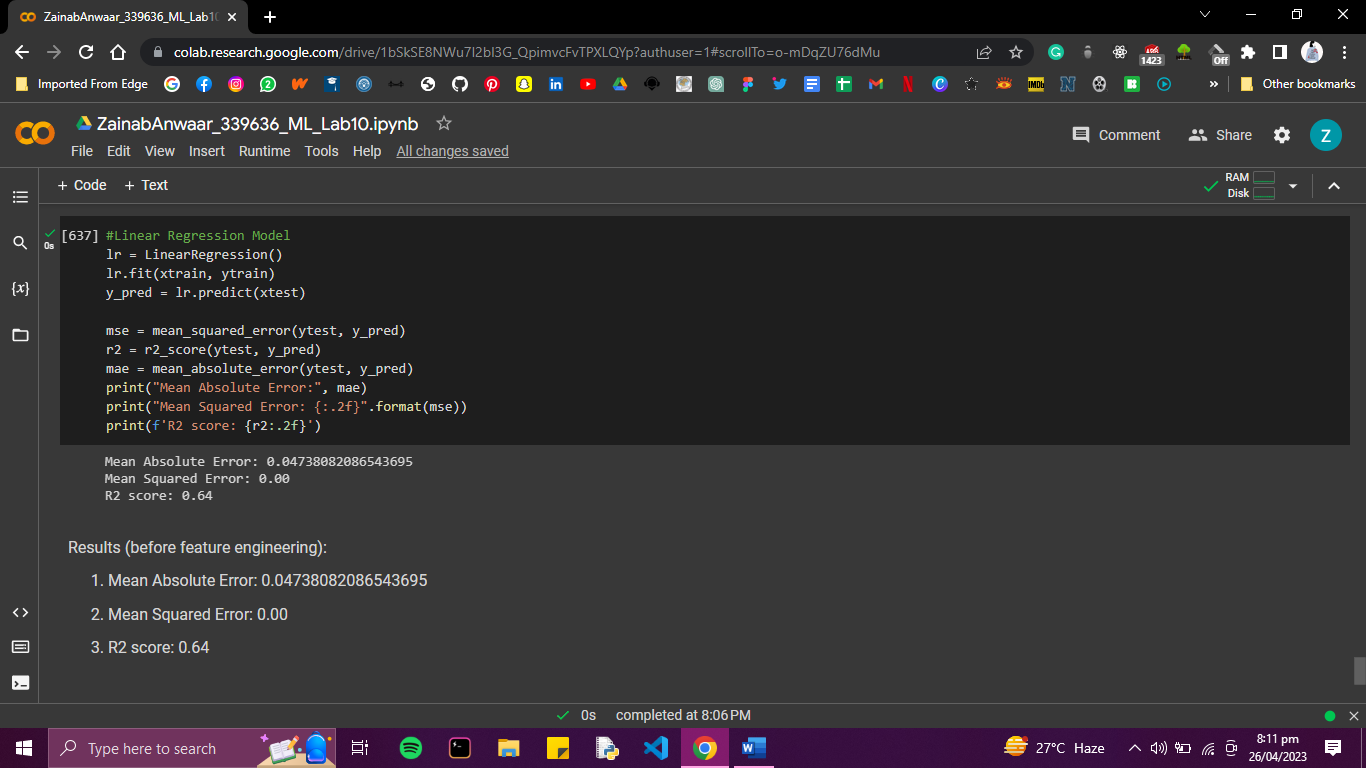
1. **Evaluate the performance of machine learning models before and after feature engineering**

Using Linear Regression:

Before the feature engineering:



After feature Engineering:



You are required to predict the price of a car, However, the choice of feature engineering techniques and machine learning models is left as your choice.

**Deliverables:**

The deliverables of this lab include:

1. A Jupyter Notebook containing the code used to apply feature engineering to the dataset
2. A report detailing the steps taken to apply feature engineering and the results obtained and summarize the findings of the analysis