Assignment-1 Report

INTRODUCTION

The report summarizes the data preparation and data exploration tasks which are carried out on the data file automobile.csv provided in this assignment. With the knowledge learnt from the previous classes and using the below provided references, the appropriate steps needed for the above two tasks have developed and implemented in the IPython Jupyter Notebook. Please refer to the following sections to find more details.

1. DATA PREPARATION

The Data received in the Data Retrieval phase is likely to be “*A Diamond in The Rough*”. Data preparation is tremendously important because our models will perform better and we’ll lose less time trying to fix strange output.

This task includes the following steps used for preparing the data from automobile.csv and the associated details are given below.

1.1. Loading Data File

In this step, the **pandas** library is imported into **IPython** and the given data file automobile.csv is loaded into a variable called **automobile** using the pandas function **read.csv()**.

After loading the file, the following pandas functions are used to validate the structure, data types and underlying data.

- **dtypes**: to find datatypes of data given in dataset
- **columns**: to find column names in given dataset
- **shape**: to find number of rows and columns presented in dataset
- **head()** : to view data present in dataset

From my observation on the data file and variable "automobile", it is noted that the given data file comprises of 238 rows and 26 columns.

1.2. Removing duplicate rows

Duplicate rows are resolved by using the function **drop_duplicates()** by keeping first occurrence [4]. After removing duplicates there are 215 rows presented in the variable **automobile**.

1.3. Dealing with Impossible Values

In the given dataset, the column **symboling** contains an impossible value “4” which is outside of the give range, it is dealt by using condition **automobile.loc[(automobile['symboling'] > 3),'symboling'] = 3** which replaces the value greater than 3 with 3.

1.4. Data entry errors

Data Entry errors (like typos) in the given dataset are examined for every column using the pandas function **value_counts**() and replaced any typos with the appropriate values using the **replace**() function.

The errors in the columns **make, aspiration** and **num-of-doors** are resolved as follows.

Column Name	Incorrect value	Updated value
make	vol00112ov	volvo
aspiration	turrrrbo	turbo
num-of-doors	fourR	four

1.5. Dealing with extra white spaces

Extra white spaces in given dataset are identified using the function **value_counts**() for all columns containing the datatype **object**, and are resolved by using the **strip**() function.

1.6. Dealing with capital letter mismatch

Capital letter mismatch in given dataset are identified using the function **value_counts**() for all columns containing datatype **object**, and are resolved by using the **lower**() function.

1.7. Dealing with missing values

Missing values (**NaN**) in given dataset across all columns are identified/counted using the following command automobile.**isnull().sum(axis = 0)**.

For the numeric columns (**normalized-losses, bore, stroke, horsepower, peak-rpm** and **price**), the missing values are resolved by replacing them with the **mean** value of that column using the **mean()** function.

For the object datatype columns (**num-of doors**), the missing values are resolved by using the **mode** value of that column.

1.8. Dealing with zero values

The zero values in the **price** column are identified using the following command **automobile.loc[(automobile['price'] == 0)].** To resolve this issue for those automobile rows, the **mean** value is calculated from the related automobiles which have the proper prices and replaced the zero values accordingly.

2. DATA EXPLORATION

2.1. Exploring single columns

2.1.1 *Nominal values:*

The column **body-style** with nominal values has chosen from the given dataset. For nominal values, we can obtain frequency or percentage which can be represented in pie chart.

Observation: I have represented body style in Pie-chart to explore different body styles presented in data. Figure1 gives pie chart for body style which makes clear that most automobiles' **body-style** is **sedan** and very few are with **body-style convertible** in given dataset.

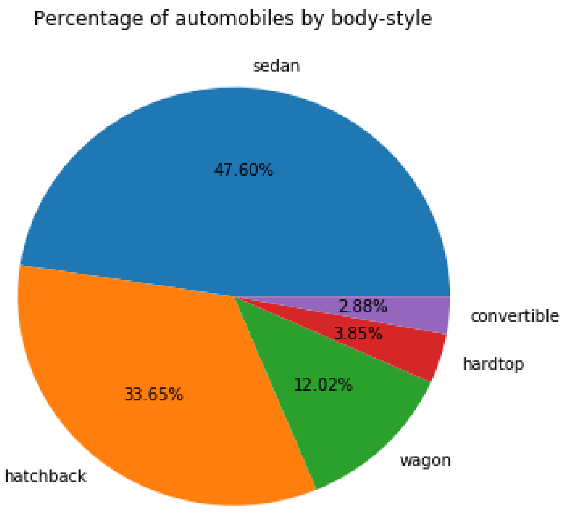

Figure 1

2.1.2 *Ordinal values:*

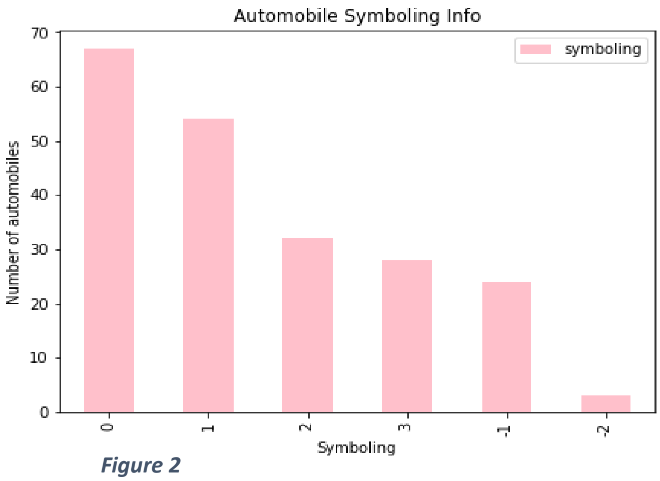

Figure 2

The column **symboling** with ordinal values is chosen from the given dataset and are visualized using the bar chart [1] as shown in figure 2. The pandas function **value_counts** is used to group the automobiles by **symboling** values and followed by bar method is used to produce the bar chat.

Observation: From the figure 2, we can say that most of the automobiles have medium (i.e., **symobling** = zero) insurance risk rating and very few got the lower risk rating.

2.1.3 *Numeric values:*

The column **length** with numerical values is chosen from the given dataset. This is to illustrate the automobile length distribution using histogram as the quantitative data is best explained using histogram, as it is used to plot the frequency of score occurrences in a continuous data set. [1]

Observation: From the figure 3, we can say that most of the automobiles are produced with the length ranging from 175 to 180. However, a very few automobiles produced with length ranging between 140 and 150, and between 200 and 210.

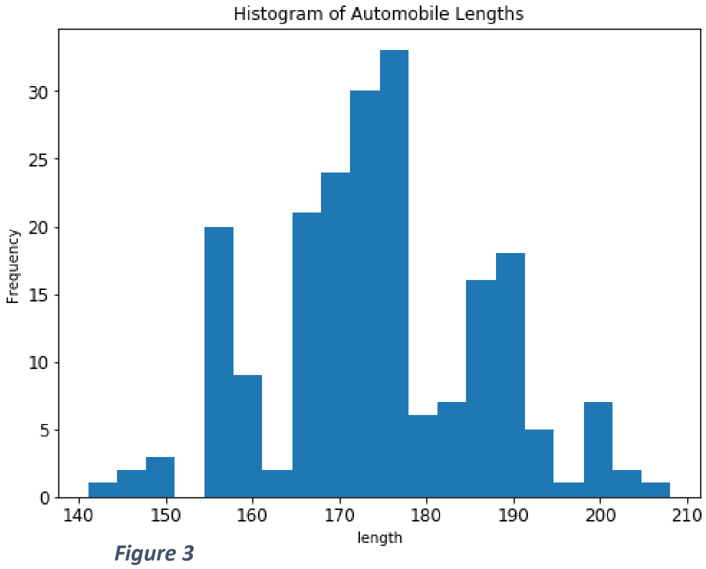

Figure 3

2.2. Relationships between columns

2.2.1 Relationship between city-mpg and fuel type:

I have chosen **boxplot** to explain relationship between **city-mpg** and **fuel-type** as it is often used in explanatory data analysis and is used to show the shape of the distribution, its central value, and its variability. Moreover, fuel-type plays a significant role in the city-mpg.

Observation: From this graph figure 4, we can say that diesel cars have high city-mpg than the gas cars. However, there are two outliers in gas cars which are giving high city-mpg than the diesel cars.

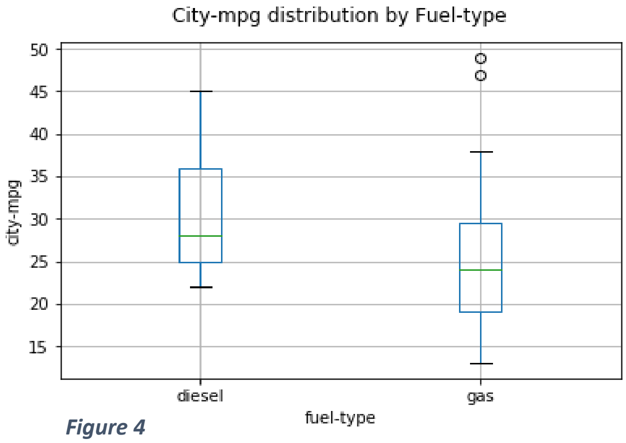

Figure 4

2.2.2 Relationship between engine-size and price:

I have chosen **scatter plot** to visualize relation between **engine-size** and **price** and to locate the higher population as well as the correlation. This clearly shows how much one variable is affected by another.

Observation: From the scatter plot shown in figure 5, we can observe positive correlation [2] between engine-size and price, as engine-size increases price of automobile also increases, and very few cars have engine-size 300 or above with higher price among all the automobiles.

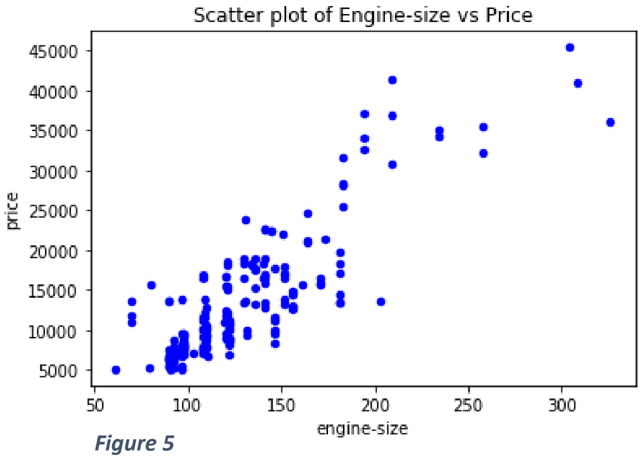

Figure 5

2.2.3 Relation between horsepower and highway-mpg:

I have chosen **Hexbin** to represent relation between **horsepower** and **highway-mpg**. This is because, using Hexbin, it is easy to locate the high-volume data points i.e., data aggregation used for grouping a dataset of N values into less than N discrete groups.

Observation: As shown in figure 6, by visualizing graph we can say that the high-volume of automobiles have the horsepower between 50 and 100 and when horsepower is lower then highway-mpg is higher. It means as power moves higher mileage tends to be lower.

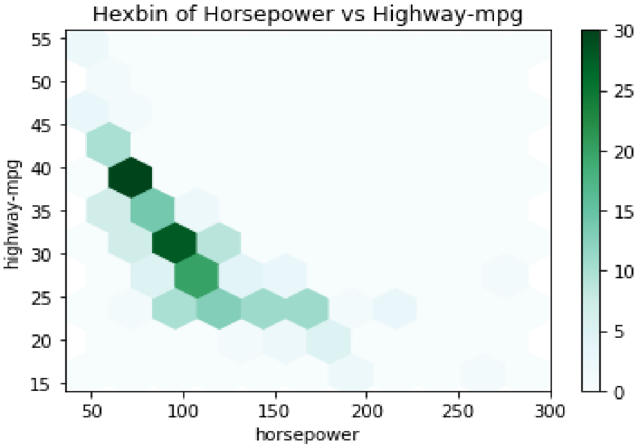

Figure 6

2.3. Scatter Matrix

Scatterplot matrices are a great way to roughly determine if you have a linear correlation between multiple variables. For a set of data variables (dimensions), the scatter plot matrix shows all the pairwise scatter plots of the variables on a single view with multiple scatterplots in a matrix format. [3]. A scatter plot matrix is table of scatter plots. Each plot is small so that many plots can be fit on a page.

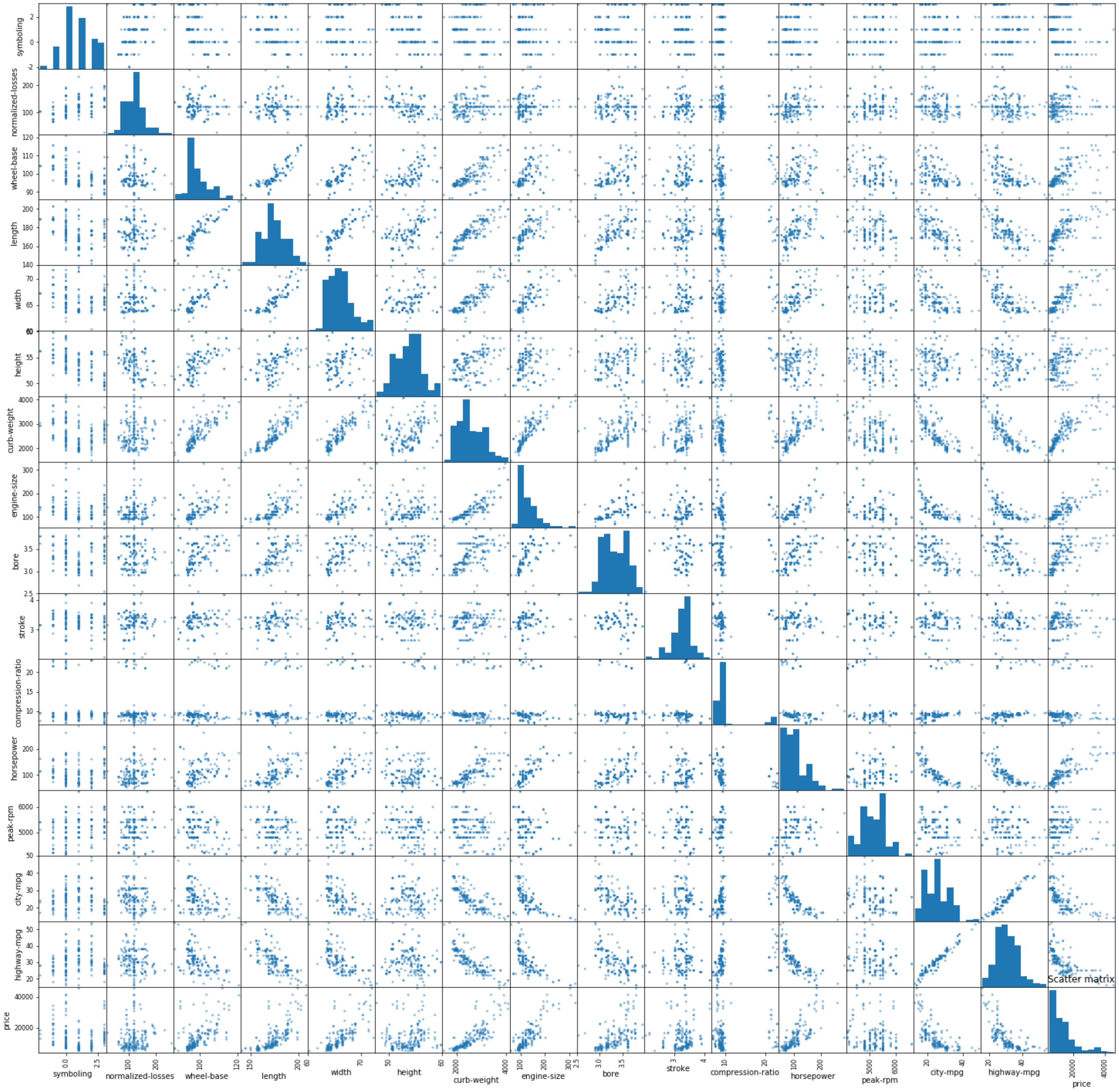

Figure 7 Scatter matrix of all numeric columns of automobile dataset

The above figure 7 is scatter matrix of all numeric columns. It can be used to determine whether the variables are correlated and whether the correlation is positive or negative. If the line goes from a high-value on the y-axis down to a high-value on the x-axis, the variables have a negative correlation. [5]

Observation: The below list includes the variable pairs (a variable on x-axis vs a variable on y-axis) which have positive, negative and no correlations.

Positive Correlation	Negative Correlation	No Correlation
City-mpg vs highway-mpg, Highway-mpg vs city-mpg, Wheel-base vs length, Curb-weight vs engine-size, Engine-size vs price, Curb-weight vs price	Horsepower vs city-mpg, Horsepower vs highway-mpg, City-mpg vs engine-size, Highway-mpg vs engine-size, Curb-weight vs city-mpg Curb-weight vs highway-mpg	Height vs highway-mpg, Height vs city-mpg, Wheel-base vs peak-rpm, Length vs peak-rpm, Width vs peak-rpm, Bore vs normalized-loss

Furthermore, histogram of wheel-base, price, horsepower and engine-size are **positively skewed**.

REFERENCES:

1. https://www.youtube.com/watch?v=hZxnzfnt5v8
2. https://www.shmoop.com/basic-statistics-probability/scatter-plots-correlation.html
3. https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Scatter_Plot_Matrix.pdf
4. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop_duplicates.html
5. https://support.minitab.com/en-us/minitab/18/help-and-how-to/graphs/how-to/matrix-plot/interpret-the-results/key-results/