In [3]:

**import** pandas **as** pd

**import** matplotlib.pylab **as** plt

**import** numpy **as** np

In [5]:

*# Load the dataset using pandas library*

df **=** pd**.**read\_csv("autodata.csv")

In [6]:

*# Check the content of dataset using df.head() and df.tail() function*

df**.**head(5)

Out[6]:

**Unnamed: symboling**

**0**

**normalized- make aspiration losses**

alfa-

**num-**

**of- doors**

**body- style**

**drive- wheels**

**engine- location**

**wheel- ... base**

**comp**

**0** 0 3 122

romero std two convertible rwd front 88.6 ...

**1**

1

3

122

alfa- romero

std two convertible

rwd

front

88.6 ...

**2** 2 1 122 alfa- romero

std two hatchback rwd front 94.5 ...

**3**

3

2

164

audi

std four

sedan

fwd

front

99.8 ...

**4** 4 2 164 audi std four sedan 4wd front 99.4 ...

# 5 rows × 30 columns

In [7]:

df**.**tail(5)

Out[7]:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Unnamed:**  **0** | **symboling** | **normalized-**  **losses** | **make** | **aspiration** | **num-**  **of- doors** | **body- style** | **drive- wheels** | **engine- location** | **wheel- base** | **... compress**  **r** |
| **196** | 196 | -1 | 95 | volvo | std | four | sedan | rwd | front | 109.1 | ... |
| **197** | 197 | -1 | 95 | volvo | turbo | four | sedan | rwd | front | 109.1 | ... |
| **198** | 198 | -1 | 95 | volvo | std | four | sedan | rwd | front | 109.1 | ... |
| **199** | 199 | -1 | 95 | volvo | turbo | four | sedan | rwd | front | 109.1 | ... |
| **200** | 200 | -1 | 95 | volvo | turbo | four | sedan | rwd | front | 109.1 | ... |

# 5 rows × 30 columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 |  | Unnamed: 0 | 201 | non-null |  | int64 |
| 1 |  | symboling | 201 | non-null |  | int64 |
| 2 |  | normalized-losses | 201 | non-null |  | int64 |
| 3 |  | make | 201 | non-null |  | object |
| 4 |  | aspiration | 201 | non-null |  | object |
| 5 |  | num-of-doors | 201 | non-null |  | object |

|  |  |  |
| --- | --- | --- |
| In [8]: | df**.**info() |  |
|  | <class 'pandas.core.frame.DataFrame'> RangeIndex: 201 entries, 0 to 200 |  |
|  | Data columns (total 30 columns):  # Column Non-Null Count | Dtype |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | body-style | 201 | non-null | object |
| 7 | drive-wheels | 201 | non-null | object |
| 8 | engine-location | 201 | non-null | object |
| 9 | wheel-base | 201 | non-null | float64 |
| 10 | length | 201 | non-null | float64 |
| 11 | width | 201 | non-null | float64 |
| 12 | height | 201 | non-null | float64 |
| 13 | curb-weight | 201 | non-null | int64 |
| 14 | engine-type | 201 | non-null | object |
| 15 | num-of-cylinders | 201 | non-null | object |
| 16 | engine-size | 201 | non-null | int64 |
| 17 | fuel-system | 201 | non-null | object |
| 18 | bore | 201 | non-null | float64 |
| 19 | stroke | 197 | non-null | float64 |
| 20 | compression-ratio | 201 | non-null | float64 |
| 21 | horsepower | 199 | non-null | float64 |
| 22 | peak-rpm | 199 | non-null | float64 |
| 23 | city-mpg | 201 | non-null | int64 |
| 24 | highway-mpg | 201 | non-null | int64 |
| 25 | price | 201 | non-null | float64 |
| 26 | city-L/100km | 201 | non-null | float64 |
| 27 | horsepower-binned | 199 | non-null | object |
| 28 | diesel | 201 | non-null | int64 |
| 29 | gas | 201 | non-null | int64 |

dtypes: float64(11), int64(9), object(10) memory usage: 47.2+ KB

In [9]:

df**.**describe()

Out[9]:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Unnamed:**  **0** | **symboling** | **normalized-**  **losses** | **wheel- base** | **length** | **width** | **height** | **curb- weight** | **engin**  **si** |
| **count** | 201.000000 | 201.000000 | 201.00000 | 201.000000 | 201.000000 | 201.000000 | 201.000000 | 201.000000 | 201.0000 |
| **mean** | 100.000000 | 0.840796 | 122.00000 | 98.797015 | 0.837102 | 0.915126 | 53.766667 | 2555.666667 | 126.8756 |
| **std** | 58.167861 | 1.254802 | 31.99625 | 6.066366 | 0.059213 | 0.029187 | 2.447822 | 517.296727 | 41.5468 |
| **min** | 0.000000 | -2.000000 | 65.00000 | 86.600000 | 0.678039 | 0.837500 | 47.800000 | 1488.000000 | 61.0000 |
| **25%** | 50.000000 | 0.000000 | 101.00000 | 94.500000 | 0.801538 | 0.890278 | 52.000000 | 2169.000000 | 98.0000 |
| **50%** | 100.000000 | 1.000000 | 122.00000 | 97.000000 | 0.832292 | 0.909722 | 54.100000 | 2414.000000 | 120.0000 |
| **75%** | 150.000000 | 2.000000 | 137.00000 | 102.400000 | 0.881788 | 0.925000 | 55.500000 | 2926.000000 | 141.0000 |
| **max** | 200.000000 | 3.000000 | 256.00000 | 120.900000 | 1.000000 | 1.000000 | 59.800000 | 4066.000000 | 326.0000 |

In [10]:

*# Evaluating for missing data*

*#isnull() #.notnull()*

*#True : missing value #False : not missing value*

df**.**isnull()

Out[10]:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Unnamed:**  **0** | **symboling** | **normalized-**  **losses** | **make** | **aspiration** | **num-**  **of- doors** | **body- style** | **drive- wheels** | **engine- location** | **wheel- compress base ... r** |
| **0** | False | False | False | False | False | False | False | False | False | False ... F |
| **1** | False | False | False | False | False | False | False | False | False | False ... F |
| **2** | False | False | False | False | False | False | False | False | False | False ... F |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3** | False | False | False | False | False | False | False | False | False | False ... F |
| **4** | False | False | False | False | False | False | False | False | False | False ... F |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... ... |
| **196** | False | False | False | False | False | False | False | False | False | False ... F |
| **197** | False | False | False | False | False | False | False | False | False | False ... F |
| **198** | False | False | False | False | False | False | False | False | False | False ... F |
| **199** | False | False | False | False | False | False | False | False | False | False ... F |
| **200** | False | False | False | False | False | False | False | False | False | False ... F |

# 201 rows × 30 columns

In [11]:

|  |  |
| --- | --- |
| df**.**isnull()**.**sum() |  |
| Unnamed: 0 | 0 |
| symboling | 0 |
| normalized-losses | 0 |
| make | 0 |
| aspiration | 0 |
| num-of-doors | 0 |
| body-style | 0 |
| drive-wheels | 0 |
| engine-location | 0 |
| wheel-base | 0 |
| length | 0 |
| width | 0 |
| height | 0 |
| curb-weight | 0 |
| engine-type | 0 |
| num-of-cylinders | 0 |
| engine-size | 0 |
| fuel-system | 0 |
| bore | 0 |
| stroke | 4 |
| compression-ratio | 0 |
| horsepower | 2 |
| peak-rpm | 2 |
| city-mpg | 0 |
| highway-mpg | 0 |
| price | 0 |
| city-L/100km | 0 |
| horsepower-binned | 2 |
| diesel | 0 |
| gas  dtype: int64 | 0 |

Out[11]:

In [12]:

df**.**notnull()

Out[12]:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Unnamed:**  **0** | **symboling** | **normalized-**  **losses** | **make** | **aspiration** | **num-**  **of- doors** | **body- style** | **drive- wheels** | **engine- location** | **wheel- base** | **... compress**  **r** |
| **0** | True | True | True | True | True | True | True | True | True | True | ... |
| **1** | True | True | True | True | True | True | True | True | True | True | ... |
| **2** | True | True | True | True | True | True | True | True | True | True | ... |
| **3** | True | True | True | True | True | True | True | True | True | True | ... |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **4** | True | True | True | True | True | True | True | True | True | True | ... |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **196** | True | True | True | True | True | True | True | True | True | True | ... |
| **197** | True | True | True | True | True | True | True | True | True | True | ... |
| **198** | True | True | True | True | True | True | True | True | True | True | ... |
| **199** | True | True | True | True | True | True | True | True | True | True | ... |
| **200** | True | True | True | True | True | True | True | True | True | True | ... |

# 201 rows × 30 columns

In [13]:

|  |  |
| --- | --- |
| df**.**notnull()**.**sum() |  |
| Unnamed: 0 | 201 |
| symboling | 201 |
| normalized-losses | 201 |
| make | 201 |
| aspiration | 201 |
| num-of-doors | 201 |
| body-style | 201 |
| drive-wheels | 201 |
| engine-location | 201 |
| wheel-base | 201 |
| length | 201 |
| width | 201 |
| height | 201 |
| curb-weight | 201 |
| engine-type | 201 |
| num-of-cylinders | 201 |
| engine-size | 201 |
| fuel-system | 201 |
| bore | 201 |
| stroke | 197 |
| compression-ratio | 201 |
| horsepower | 199 |
| peak-rpm | 199 |
| city-mpg | 201 |
| highway-mpg | 201 |
| price | 201 |
| city-L/100km | 201 |
| horsepower-binned | 199 |
| diesel | 201 |
| gas  dtype: int64 | 201 |

Out[13]:

In [15]:

*# Calculate the mean value for stroke column*

avg\_stroke **=** df["stroke"]**.**astype("float")**.**mean(axis **=** 0) print("Average of stroke", avg\_stroke)

|  |  |  |
| --- | --- | --- |
|  | | Average of stroke 3.256903553299492 |
| In | [16]: | *# Calculate the mean value for the horsepower column* |
|  |  | avg\_hp **=** df["horsepower"]**.**astype("float")**.**mean(axis **=** 0) print("Average of stroke:",avg\_hp) |
|  |  | Average of stroke: 103.39698492462311 |
| In | [17]: | df["horsepower"]**.**replace(np**.**nan, avg\_hp, inplace **= True**) |

In [18]:

avg\_rpm **=** df["peak-rpm"]**.**astype("float")**.**mean(axis **=** 0) print("Average of stroke:",avg\_rpm)

Average of stroke: 5117.587939698493

In [21]:

df["peak-rpm"]**.**replace(np**.**nan, avg\_hp, inplace **= True**)

In [22]:

df['num-of-doors']**.**value\_counts()

Out[22]:

In [23]:

Out[23]:

In [25]:

*# replace the missing num-of-doors value by the most frequent* df["num-of-doors"]**.**replace(np**.**nan, "four", inplace**=True**) *#Simply drop whole row with NaN in horsepower-ninned column* df**.**dropna(subset**=**["horsepower-binned"], axis**=**0, inplace**=True**) *# reset index, because we drop the two roes*

df**.**reset\_index(drop**=True**, inplace**=True**)

four 115

two 86

Name: num-of-doors, dtype: int64

df['num-of-doors']**.**value\_counts()**.**idxmax()

'four'

In [26]:

|  |  |
| --- | --- |
| df**.**isnull()**.**sum() |  |
| Unnamed: 0 | 0 |
| symboling | 0 |
| normalized-losses | 0 |
| make | 0 |
| aspiration | 0 |
| num-of-doors | 0 |
| body-style | 0 |
| drive-wheels | 0 |
| engine-location | 0 |
| wheel-base | 0 |
| length | 0 |
| width | 0 |
| height | 0 |
| curb-weight | 0 |
| engine-type | 0 |
| num-of-cylinders | 0 |
| engine-size | 0 |
| fuel-system | 0 |
| bore | 0 |
| stroke | 4 |
| compression-ratio | 0 |
| horsepower | 0 |
| peak-rpm | 0 |
| city-mpg | 0 |
| highway-mpg | 0 |
| price | 0 |
| city-L/100km | 0 |
| horsepower-binned | 0 |
| diesel | 0 |
| gas | 0 |
| dtype: int64 |  |

Out[26]:

In [27]:

*# Data Standardization*

df['city-L/100km'] **=** 235**/**df["city-mpg"] df**.**head()

Out[27]:

**Unnamed: symboling**

**0**

**normalized- make aspiration losses**

alfa-

**num-**

**of- doors**

**body- style**

**drive- wheels**

**engine- location**

**wheel- ... base**

**comp**

**0** 0 3 122

romero std two convertible rwd front 88.6 ...

**1**

1

3

122

alfa- romero

std two convertible

rwd

front

88.6 ...

**2** 2 1 122 alfa- romero

std two hatchback rwd front 94.5 ...

**3**

3

2

164

audi

std four

sedan

fwd

front

99.8 ...

**4** 4 2 164 audi std four sedan 4wd front 99.4 ...

# 5 rows × 30 columns

In [28]:

df["highway-L/100km"] **=** 235**/**df["highway-mpg"] df**.**head()

Out[28]:

**Unnamed: symboling**

**0**

**normalized- make aspiration losses**

alfa-

**num-**

**of- doors**

**body- style**

**drive- wheels**

**engine- location**

**wheel- ... horse base**

**0** 0 3 122

romero std two convertible rwd front 88.6 ...

**1**

1

3

122

alfa- romero

std two convertible

rwd

front

88.6 ...

**2** 2 1 122 alfa- romero

std two hatchback rwd front 94.5 ...

**3**

3

2

164

audi

std four

sedan

fwd

front

99.8 ...

**4** 4 2 164 audi std four sedan 4wd front 99.4 ...

# 5 rows × 31 columns

In [29]:

*# Data normalization*

*# Techniques*

*#*

*(1) --- Using The min-max feature scaling*

*# The min-max approach (often called normalization) rescales the feature to a hard and f*

*#*

*(2) --- Using The maximum absolute scaling*

*# The maximum absolute scaling rescales each feature between -1 and 1 by dividing every*

*#*

*(3) --- Using The z-score method*

*# The z-score method (often called standardization) transforms the info into distributio*

In [31]:

df['height'] **=** df['width']**/**df['height']**.**max()

df[["length", "width", "height"]]**.**head()

|  |  |  |  |
| --- | --- | --- | --- |
| Out[31]: | **length** | **width** | **height** |
|  | **0** 0.811148 | 0.890278 | 0.014888 |
|  | **1** 0.811148 | 0.890278 | 0.014888 |
|  | **2** 0.822681 | 0.909722 | 0.015213 |
|  | **3** 0.848630 | 0.919444 | 0.015375 |
|  | **4** 0.848630 | 0.922222 | 0.015422 |

In [32]:

Out[32]:

In [33]:

Out[33]:

In [34]:

Index(['Unnamed: 0', 'symboling', 'normalized-losses', 'make', 'aspiration', 'num-of-doors', 'body-style', 'drive-wheels', 'engine-location',

df**.**columns

'wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-type',

'num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'stroke', 'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg',

'highway-mpg', 'price', 'city-L/100km', 'horsepower-binned', 'diesel', 'gas', 'highway-L/100km'],

dtype='object')

df['aspiration']**.**value\_counts()

std 163

turbo 36

Name: aspiration, dtype: int64

dummy\_variable\_1 **=** pd**.**get\_dummies(df["aspiration"]) dummy\_variable\_1**.**head()

|  |  |  |  |
| --- | --- | --- | --- |
| Out[34]: |  | **std** | **turbo** |
|  | **0** | 1 | 0 |
|  | **1** | 1 | 0 |
|  | **2** | 1 | 0 |
|  | **3** | 1 | 0 |
|  | **4** | 1 | 0 |

In [36]:

df **=** pd**.**concat([df, dummy\_variable\_1], axis**=**1) df**.**drop("aspiration", axis **=** 1, inplace**=True**)

In [37]:

df**.**head()

Out[37]:

romero

**num-**

**of- doors**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Unnamed:**  **0** | **symboling** | **normalized-**  **losses make** |
| **0** | 0 | 3 | 122 alfa- romero |
| **1** | 1 | 3 | 122 alfa- romero |
| **2** | 2 | 1 | 122 alfa- |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **body- style** | **drive- wheels** | **engine- location** | **wheel- base** | **length ... price** |
| convertible | rwd | front | 88.6 | 0.811148 ... 13495.0 |
| convertible | rwd | front | 88.6 | 0.811148 ... 16500.0 |
| hatchback | rwd | front | 94.5 | 0.822681 ... 16500.0 |

two

two

two

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3** | 3 | 2 | 164 | audi | four | sedan | fwd | front | 99.8 0.848630 ... 13950.0 |
| **4** | 4 | 2 | 164 | audi | four | sedan | 4wd | front | 99.4 0.848630 ... 17450.0 |

# 5 rows × 34 columns

In [40]:

bins **=** np**.**linspace(min(df["horsepower"]), max(df["horsepower"]), 4) bins

Out[40]:

In [41]:

group\_names **=** ['low', 'Medium', 'High']

array([ 48. , 119.33333333, 190.66666667, 262. ])

In [46]:

df['horsepower-binned'] **=** pd**.**cut(df['horsepower'], bins, labels**=**group\_names, include\_low df[['horsepower', 'horsepower-binned']]**.**head(20)

|  |  |  |  |
| --- | --- | --- | --- |
| Out[46]: |  | **horsepower** | **horsepower-binned** |
|  | **0** | 111.0 | low |
|  | **1** | 111.0 | low |
|  | **2** | 154.0 | Medium |
|  | **3** | 102.0 | low |
|  | **4** | 115.0 | low |
|  | **5** | 110.0 | low |
|  | **6** | 110.0 | low |
|  | **7** | 110.0 | low |
|  | **8** | 140.0 | Medium |
|  | **9** | 101.0 | low |
|  | **10** | 101.0 | low |
|  | **11** | 121.0 | Medium |
|  | **12** | 121.0 | Medium |
|  | **13** | 121.0 | Medium |
|  | **14** | 182.0 | Medium |
|  | **15** | 182.0 | Medium |
|  | **16** | 182.0 | Medium |
|  | **17** | 48.0 | low |
|  | **18** | 70.0 | low |
|  | **19** | 70.0 | low |

In [47]:

df["horsepower-binned"]**.**value\_counts()

Out[47]:

In [49]:

low 151

Medium 43

High 5

Name: horsepower-binned, dtype: int64

df["peak-rpm"]**=**df["peak-rpm"]**.**astype(float, copy**=True**)

In [51]:

bins **=** np**.**linspace(min(df["peak-rpm"]), max(df["peak-rpm"]), 4) bins

Out[51]:

In [52]:

group\_names1 **=** ['Low', 'Medium', 'High']

array([4150. , 4966.66666667, 5783.33333333, 6600. ])

In [53]:

df['peakrpm-binned'] **=** pd**.**cut(df['peak-rpm'], bins, labels**=**group\_names, include\_lowest**=T** df[['peak-rpm','peakrpm-binned']]**.**head(20)

|  |  |  |  |
| --- | --- | --- | --- |
| Out[53]: |  | **peak-rpm** | **peakrpm-binned** |
|  | **0** | 5000.0 | Medium |
|  | **1** | 5000.0 | Medium |
|  | **2** | 5000.0 | Medium |
|  | **3** | 5500.0 | Medium |
|  | **4** | 5500.0 | Medium |
|  | **5** | 5500.0 | Medium |
|  | **6** | 5500.0 | Medium |
|  | **7** | 5500.0 | Medium |
|  | **8** | 5500.0 | Medium |
|  | **9** | 5800.0 | High |
|  | **10** | 5800.0 | High |
|  | **11** | 4250.0 | low |
|  | **12** | 4250.0 | low |
|  | **13** | 4250.0 | low |
|  | **14** | 5400.0 | Medium |
|  | **15** | 5400.0 | Medium |
|  | **16** | 5400.0 | Medium |
|  | **17** | 5100.0 | Medium |
|  | **18** | 5400.0 | Medium |
|  | **19** | 5400.0 | Medium |

In [54]:

df["peakrpm-binned"]**.**value\_counts()

Out[54]:

In [56]:

Medium 109

low 69

High 21

Name: peakrpm-binned, dtype: int64

df["wheel-base"]**=**df["wheel-base"]**.**astype(float, copy**=True**)

In [58]:

bins **=** np**.**linspace(min(df["wheel-base"]), max(df["wheel-base"]), 4) bins

Out[58]:

array([ 86.6 , 98.03333333, 109.46666667, 120.9 ])

In [59]:

group\_names **=** ['Low', 'Medium', 'High']

In [60]:

df['wheelbase-binned'] **=** pd**.**cut(df['wheel-base'], bins, labels**=**group\_names, include\_lowe df[['wheel-base','wheelbase-binned']]**.**head(20)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Out[60]: |  | **wheel-base** | **wheelbase-binned** |  |
|  | **0** | 88.6 | Low |  |
|  | **1** | 88.6 | Low |  |
|  | **2** | 94.5 | Low |  |
|  | **3** | 99.8 | Medium |  |
|  | **4** | 99.4 | Medium |  |
|  | **5** | 99.8 | Medium |  |
|  | **6** | 105.8 | Medium |  |
|  | **7** | 105.8 | Medium |  |
|  | **8** | 105.8 | Medium |  |
|  | **9** | 101.2 | Medium |  |
|  | **10** | 101.2 | Medium |  |
|  | **11** | 101.2 | Medium |  |
|  | **12** | 101.2 | Medium |  |
|  | **13** | 103.5 | Medium |  |
|  | **14** | 103.5 | Medium |  |
|  | **15** | 103.5 | Medium |  |
|  | **16** | 110.0 | High |  |
|  | **17** | 88.4 | Low |  |
|  | **18** | 94.5 | Low |  |
|  | **19** | 94.5 | Low |  |
|  |  |  |  |  |
| In [ ]: |  |  |  |  |