Final Assesment Project 1 Car Analysis

August 9, 2022

This is the analysis of the car data for the Data Engineering training assesment for Lyxantha White.

[2]:
```
import pandas as pd
import numpy as np
```

[3]:
```
df=pd.read_csv(r"C:\Users\18324\Documents\Hexaware\training\Evaluation␣
↪Project\Final Project\Final Project\cars.csv")
```

[4]:
```
df.head(10)
```

[4]:
```
   symboling normalized losses         make fuel-type aspiration num of doors  \
0          3                ?  alfa-romero       gas        std          two
1          3                ?  alfa-romero       gas        std          two
2          1                ?  alfa-romero       gas        std          two
3          2              164         audi       gas        std         four
4          2              164         audi       gas        std         four
5          2                ?         audi       gas        std          two
6          1              158         audi       gas        std         four
7          1                ?         audi       gas        std         four
8          1              158         audi       gas      turbo         four
9          0                ?         audi       gas      turbo          two

    body style drive wheels engine location  wheel base  …  engine size  \
0  convertible          rwd           front        88.6  …          130
1  convertible          rwd           front        88.6  …          130
2    hatchback          rwd           front        94.5  …          152
3        sedan          fwd           front        99.8  …          109
4        sedan          4wd           front        99.4  …          136
5        sedan          fwd           front        99.8  …          136
6        sedan          fwd           front       105.8  …          136
7        wagon          fwd           front       105.8  …          136
8        sedan          fwd           front       105.8  …          131
9    hatchback          4wd           front        99.5  …          131

   fuel system  bore  stroke compression ratio horsepower  peak rpm city mpg  \
0         mpfi  3.47    2.68               9.0        111      5000       21
1         mpfi  3.47    2.68               9.0        111      5000       21
2         mpfi  2.68    3.47               9.0        154      5000       19
```

```
3          mpfi  3.19    3.40              10.0        102       5500        24
4          mpfi  3.19    3.40               8.0        115       5500        18
5          mpfi  3.19    3.40               8.5        110       5500        19
6          mpfi  3.19    3.40               8.5        110       5500        19
7          mpfi  3.19    3.40               8.5        110       5500        19
8          mpfi  3.13    3.40               8.3        140       5500        17
9          mpfi  3.13    3.40               7.0        160       5500        16

  highway mpg  price
0          27  13495
1          27  16500
2          26  16500
3          30  13950
4          22  17450
5          25  15250
6          25  17710
7          25  18920
8          20  23875
9          22      ?

[10 rows x 26 columns]
```

[5]:
```
df.dtypes
```

[5]:
```
symboling              int64
normalized losses     object
make                  object
fuel-type             object
aspiration            object
num of doors          object
body style            object
drive wheels          object
engine location       object
wheel base           float64
length               float64
width                float64
height               float64
curb weight            int64
engine type           object
num of cylinders      object
engine size            int64
fuel system           object
bore                  object
stroke                object
compression ratio    float64
horsepower            object
peak rpm              object
```

```
city mpg               int64
highway mpg            int64
price                 object
dtype: object
```

Replace '?' with Null Values and change typing of needed columns to int and float.

[6]:
```
df = df.replace('?', np.NaN)
```

[7]:
```
df['num of doors'].value_counts()
```

[7]:
```
four    114
two      89
Name: num of doors, dtype: int64
```

[8]:
```
df['num of cylinders'].value_counts()
```

[8]:
```
four      159
six        24
five       11
eight       5
two         4
three       1
twelve      1
Name: num of cylinders, dtype: int64
```

[9]:
```
df = df.replace({'two':2, 'three':3,'four':4, 'five':5,'six':6, 'eight':8,␣
 ↪'twelve':12})
```

[10]:
```
df['num of cylinders'].value_counts()
```

[10]:
```
4     159
6      24
5      11
8       5
2       4
3       1
12      1
Name: num of cylinders, dtype: int64
```

[11]:
```
df['num of doors']=pd.to_numeric(df['num of doors'])
df['num of cylinders']=pd.to_numeric(df['num of cylinders'])
df['bore']=pd.to_numeric(df['bore'])
df['horsepower']=pd.to_numeric(df['horsepower'])
df['stroke']=pd.to_numeric(df['stroke'])
df['peak rpm']=pd.to_numeric(df['peak rpm'])
df['price']=pd.to_numeric(df['price'])
```

[12]:
```
df.dtypes
```

[12]:
```
symboling               int64
normalized losses      object
make                   object
fuel-type              object
aspiration             object
num of doors          float64
body style             object
drive wheels           object
engine location        object
wheel base            float64
length                float64
width                 float64
height                float64
curb weight             int64
engine type            object
num of cylinders        int64
engine size             int64
fuel system            object
bore                  float64
stroke                float64
compression ratio     float64
horsepower            float64
peak rpm              float64
city mpg                int64
highway mpg             int64
price                 float64
dtype: object
```

What percentage of cars will be suitable for a family (i.e. num of doors=4, price <20,000 & mileage >17)?

[13]:
```
df['num of doors'].value_counts()
```

[13]:
```
4.0    114
2.0     89
Name: num of doors, dtype: int64
```

[14]:
```
df.loc[(df['num of doors']>=4 )& (df['price'] < 20000 )& (df['city mpg']>17) &(
→df['highway mpg']>17)&(df['fuel-type']=='gas')].shape
```

[14]:
```
(82, 26)
```

[15]:
```
df.shape
```

[15]:
```
(205, 26)
```

[16]:
```
ff_percentage = 82/205
```

[17]:
```
print(ff_percentage)
```

```
0.4
```

The percentage of cars that are suitable for families is 40% of the cars in the dataset.

Which company has generated more car options for customers?

[18]:
```
df['make'].value_counts()
```

[18]:
```
toyota           32
nissan           18
mazda            17
honda            13
mitsubishi       13
subaru           12
volkswagen       12
volvo            11
peugot           11
dodge             9
mercedes-benz     8
bmw               8
audi              7
plymouth          7
saab              6
porsche           5
isuzu             4
chevrolet         3
jaguar            3
alfa-romero       3
renault           2
mercury           1
Name: make, dtype: int64
```

Toyota has the most car options to chose from.

What is the ratio of diesel cars to that gas ones?

[19]:
```
df['fuel-type'].value_counts()
```

[19]:
```
gas       185
diesel     20
Name: fuel-type, dtype: int64
```

The ratio of diesel to gas cars is 4:37, with 10.81% of cars avaliable being diesel.

What is the count of performance cars present in the dataset (horsepower > 150)?

[20]:
```
df[df['horsepower']>=150].shape
```

[20]: (32, 26)

[21]:
```
temp = df[df['horsepower']>=150]
temp['make'].value_counts()
```

[21]:
```
nissan           6
porsche          4
toyota           4
mercedes-benz    4
volvo            3
jaguar           3
bmw              3
saab             2
audi             1
mercury          1
alfa-romero      1
Name: make, dtype: int64
```

[39]:
```
df[df['horsepower']>=150].sort_values(by=['horsepower'],ascending=False).
↪head(10)
```

[39]:
```
     symboling normalized losses           make fuel-type aspiration  \
129          1                NaN        porsche       gas        std
49           0                NaN         jaguar       gas        std
127          3                NaN        porsche       gas        std
126          3                NaN        porsche       gas        std
128          3                NaN        porsche       gas        std
105          3                194         nissan       gas      turbo
74           1                NaN  mercedes-benz       gas        std
73           0                NaN  mercedes-benz       gas        std
17           0                NaN            bmw       gas        std
16           0                NaN            bmw       gas        std

     num of doors   body style drive wheels engine location  wheel base  …  \
129           2.0    hatchback          rwd           front        98.4  …
49            2.0        sedan          rwd           front       102.0  …
127           2.0      hardtop          rwd            rear        89.5  …
126           2.0      hardtop          rwd            rear        89.5  …
128           2.0  convertible          rwd            rear        89.5  …
105           2.0    hatchback          rwd           front        91.3  …
74            2.0      hardtop          rwd           front       112.0  …
73            4.0        sedan          rwd           front       120.9  …
17            4.0        sedan          rwd           front       110.0  …
16            2.0        sedan          rwd           front       103.5  …

     engine size  fuel system  bore  stroke compression ratio  horsepower  \
129          203         mpfi  3.94    3.11              10.0       288.0
```

```
49              326          mpfi   3.54     2.76                   11.5         262.0
127             194          mpfi   3.74     2.90                    9.5         207.0
126             194          mpfi   3.74     2.90                    9.5         207.0
128             194          mpfi   3.74     2.90                    9.5         207.0
105             181          mpfi   3.43     3.27                    7.8         200.0
74              304          mpfi   3.80     3.35                    8.0         184.0
73              308          mpfi   3.80     3.35                    8.0         184.0
17              209          mpfi   3.62     3.39                    8.0         182.0
16              209          mpfi   3.62     3.39                    8.0         182.0

     peak rpm city mpg  highway mpg    price
129    5750.0       17           28      NaN
49     5000.0       13           17  36000.0
127    5900.0       17           25  34028.0
126    5900.0       17           25  32528.0
128    5900.0       17           25  37028.0
105    5200.0       17           23  19699.0
74     4500.0       14           16  45400.0
73     4500.0       14           16  40960.0
17     5400.0       15           20  36880.0
16     5400.0       16           22  41315.0

[10 rows x 26 columns]
```

There are 32 performance cars present with the highest horsepower available from a porsche at 288 horsepower and the company with the most cars in the category being nissan with 6 cars avaliable.

Which is the most compact among all cars?

[23]:
```
df.loc[:,['length','width','height']].
↪sort_values(by=['length','width','height'])
```

[23]:
```
    length  width  height
18   141.1   60.3    53.2
30   144.6   63.9    50.8
31   144.6   63.9    50.8
32   150.0   64.0    52.6
33   150.0   64.0    52.6
..     …      …      …
47   199.6   69.6    52.8
48   199.6   69.6    52.8
70   202.6   71.7    56.3
71   202.6   71.7    56.5
73   208.1   71.7    56.7

[205 rows x 3 columns]
```

[24]:
```
df.iloc[18]
```

[24]:
```
symboling                        2
normalized losses              121
make                     chevrolet
fuel-type                      gas
aspiration                     std
num of doors                   2.0
body style               hatchback
drive wheels                   fwd
engine location              front
wheel base                    88.4
length                       141.1
width                         60.3
height                        53.2
curb weight                   1488
engine type                      l
num of cylinders                 3
engine size                     61
fuel system                   2bbl
bore                          2.91
stroke                        3.03
compression ratio              9.5
horsepower                    48.0
peak rpm                    5100.0
city mpg                        47
highway mpg                     53
price                       5151.0
Name: 18, dtype: object
```

The most compact vechile based on the body size of length width and height is the Chevrolet Hathback.

What are the main factors that are associated with the mileage of a car?

[25]:
```
df.corrwith(df['city mpg']).sort_values()
```

[25]:
```
horsepower          -0.803620
curb weight         -0.757414
price               -0.686571
length              -0.670909
engine size         -0.653658
width               -0.642704
bore                -0.594584
wheel base          -0.470414
num of cylinders    -0.445837
peak rpm            -0.113788
height              -0.048640
stroke              -0.042906
symboling           -0.035823
```

```
num of doors        -0.020812
compression ratio    0.324701
highway mpg          0.971337
city mpg             1.000000
dtype: float64
```

[26]:
```
df.corrwith(df['highway mpg']).sort_values()
```

[26]:
```
curb weight         -0.797465
horsepower          -0.770908
price               -0.704692
length              -0.704662
engine size         -0.677470
width               -0.677218
bore                -0.594572
wheel base          -0.544082
num of cylinders    -0.466666
height              -0.107358
peak rpm            -0.054257
stroke              -0.044528
num of doors        -0.044507
symboling            0.034606
compression ratio    0.265201
city mpg             0.971337
highway mpg          1.000000
dtype: float64
```

The main factors for the milage are the horsepower and curbweight values of the vechile. As both the horsepower and curbweight increase the mileage goes down. After horsepower and curbweight the size of the vechile is the biggest factor.

What percentage of cars are budget-friendly (price < 10,000)?

[27]:
```
df[df['price']<10000].shape
```

[27]:
```
(98, 26)
```

[28]:
```
bf_percentage = 98/205
print(bf_percentage)
```

```
0.47804878048780486
```

47.8% of the cars are budget friendly.

Which cars are the most efficient of all (city mpg >= 30)?

[29]:
```
df[df['city mpg']>=30].sort_values(by=['city mpg'], ascending =False).head(10)
```

[29]:
```
    symboling normalized losses      make fuel-type aspiration  \
30          2               137     honda       gas        std
```

```
18           2              121  chevrolet       gas        std
90           1              128     nissan    diesel        std
20           0               81  chevrolet       gas        std
160          0               91     toyota       gas        std
32           1              101      honda       gas        std
159          0               91     toyota    diesel        std
44           1              NaN      isuzu       gas        std
45           0              NaN      isuzu       gas        std
19           1               98  chevrolet       gas        std

     num of doors body style drive wheels engine location  wheel base  …  \
30            2.0  hatchback          fwd           front        86.6  …
18            2.0  hatchback          fwd           front        88.4  …
90            2.0      sedan          fwd           front        94.5  …
20            4.0      sedan          fwd           front        94.5  …
160           4.0      sedan          fwd           front        95.7  …
32            2.0  hatchback          fwd           front        93.7  …
159           4.0  hatchback          fwd           front        95.7  …
44            2.0      sedan          fwd           front        94.5  …
45            4.0      sedan          fwd           front        94.5  …
19            2.0  hatchback          fwd           front        94.5  …

     engine size  fuel system  bore  stroke compression ratio  horsepower  \
30            92         1bbl  2.91    3.41               9.6        58.0
18            61         2bbl  2.91    3.03               9.5        48.0
90           103          idi  2.99    3.47              21.9        55.0
20            90         2bbl  3.03    3.11               9.6        70.0
160           98         2bbl  3.19    3.03               9.0        70.0
32            79         1bbl  2.91    3.07              10.1        60.0
159          110          idi  3.27    3.35              22.5        56.0
44            90         2bbl  3.03    3.11               9.6        70.0
45            90         2bbl  3.03    3.11               9.6        70.0
19            90         2bbl  3.03    3.11               9.6        70.0

     peak rpm city mpg  highway mpg   price
30     4800.0       49           54  6479.0
18     5100.0       47           53  5151.0
90     4800.0       45           50  7099.0
20     5400.0       38           43  6575.0
160    4800.0       38           47  7738.0
32     5500.0       38           42  5399.0
159    4500.0       38           47  7788.0
44     5400.0       38           43     NaN
45     5400.0       38           43     NaN
19     5400.0       38           43  6295.0

[10 rows x 26 columns]
```

[30]:
```
temp =df[df['city mpg']>=30]
```

[31]:
```
temp['body style'].value_counts()
```

[31]:
```
sedan        27
hatchback    25
wagon         4
hardtop       1
Name: body style, dtype: int64
```

The most fuel efficent cars are the sedan and hatchback models, with the Honda Hatchback being the most efficent at 49 mpg and the Chevrolet Hatchback and the Nissan Sedan following close behind with 47 mpg and 45 mpg respectivly.

What percentage of data is missing from the dataset?

[32]:
```
print(df.isnull().sum().sum())
df.isnull().sum()
```

```
59
```

[32]:
```
symboling             0
normalized losses    41
make                  0
fuel-type             0
aspiration            0
num of doors          2
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                  4
stroke                4
compression ratio     0
horsepower            2
peak rpm              2
city mpg              0
highway mpg           0
price                 4
dtype: int64
```

[33]:
```
df.size
```

[33]: 5330

[34]:
```
dm_percentage=59/5330
print(dm_percentage)
nlm_percentage = 41/50
print(nlm_percentage)
```

```
0.011069418386491557
0.82
```

The percentage of missing data is 1.106% of the entire dataset with 82% of the missing data coming from the normalized loss column.

Which feature of the car affects the most to the pricing?

[35]:
```
df.corrwith(df['price']).sort_values()
```

[35]:
```
highway mpg         -0.704692
city mpg            -0.686571
peak rpm            -0.101649
symboling           -0.082391
num of doors         0.046532
compression ratio    0.071107
stroke               0.082310
height               0.135486
bore                 0.543436
wheel base           0.584642
length               0.690628
num of cylinders     0.708645
width                0.751265
horsepower           0.810533
curb weight          0.834415
engine size          0.872335
price                1.000000
dtype: float64
```

How powerful the engine is effects the pricing the most. The higher the horsepower and the higher the engine size the higher the price. This is also true of the number of cylinders, as they effect the power of the engine.

[ ]:
```

```