Importing the libraries

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
```

Importing the data

In [2]:
```
data = pd.read_csv("car_performance.csv")
```

In [3]:
```
data
```

Out[3]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82	2720	19.4	82	1	chevy s-10

398 rows × **9** columns

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    int64
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model year    398 non-null    int64
 7   origin        398 non-null    int64
 8   car name      398 non-null    object
dtypes: float64(3), int64(5), object(1)
memory usage: 28.1 KB
```

No Null Values Found

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
mpg             0
cylinders       0
displacement    0
horsepower      0
weight          0
acceleration    0
model year      0
origin          0
car name        0
dtype: int64
```

In [6]:
```
a = 1
for i in data["car name"]:
    i.append(str(a))
```

In [7]:
```
data.insert(0,"brand",1)
```

Handling Irrelevent Values

In [8]:
```
make_type_correction = {
    "chevy":"chevrolet",
    "vw":"volkswagen",
    "mazda":"maxda",
    "toyouta":"toyota",
    "mercedes":"mercedes-benz",
    "chevroelt":"chevrolet",
}

data["brand"] = data['brand'].replace(make_type_correction)
```

In [9]:
```
data.brand.unique()
```

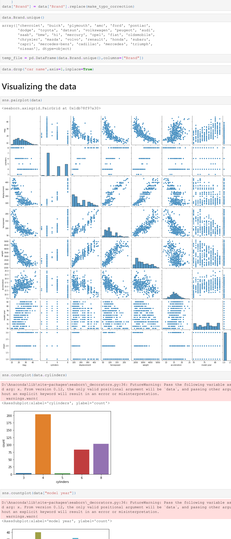
Out[9]:
```
array(['chevrolet', 'buick', 'plymouth', 'amc', 'ford', 'pontiac',
       'dodge', 'toyota', 'datsun', 'volkswagen', 'peugeot', 'audi',
       'saab', 'bmw', 'chrysler', 'maxda', 'mercury', 'fiat', 'oldsmobile',
       'honda', 'subaru', 'capri', 'mercedes-benz', 'triumph', 'nissan'],
      dtype=object)
```

In [10]:
```
temp_file = pd.DataFrame(data.brand.unique(),columns=["Brand"])
```

In [11]:
```
data.drop("car name",axis=1,inplace=True)
```

Visualizing the data

In [12]:
```
sns.pairplot(data)
```

Out[12]:


In [13]:
```
sns.countplot(data.cylinders)
```

Out[13]:
```
C:\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments will be
ost an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel="cylinders", ylabel="count">
```

In [14]:
```
sns.countplot(data["model year"])
```

Out[14]:
```
C:\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments will be
ost an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel="model year", ylabel="count">
```

In [15]:
```
sns.countplot(data.origin)
```

Out[15]:
```
C:\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments will be
ost an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel="origin", ylabel="count">
```

In [16]:
```
sns.distplot(data.displacement)
```

Out[16]:
```
C:\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
<AxesSubplot:xlabel="displacement", ylabel="Density">
```

In [17]:
```
sns.distplot(data.horsepower)
```

Out[17]:
```
C:\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
<AxesSubplot:xlabel="horsepower", ylabel="Density">
```


In [18]:
```
sns.distplot(data.weight)
```

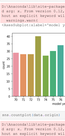
Out[18]:
```
C:\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
<AxesSubplot:xlabel="weight", ylabel="Density">
```


In [19]:
```
sns.distplot(data.acceleration)
```

Out[19]:
```
C:\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
<AxesSubplot:xlabel="acceleration", ylabel="Density">
```


In [20]:
```
sns.scatterplot(data.acceleration,data.horsepower)
```

Out[20]:
```
C:\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments w
ost an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel="acceleration", ylabel="horsepower">
```



In [21]:
```
sns.scatterplot(data.acceleration,data.displacement)
```

Out[21]:
```
C:\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments w
ost an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel="acceleration", ylabel="displacement">
```

In [22]:
```
from sklearn.preprocessing import LabelEncoder
```

In [23]:
```
temp_file
```

Out[23]:

	Brand
0	chevrolet
1	buick
2	plymouth
3	amc
4	ford
5	pontiac
6	dodge
7	toyota
8	datsun
9	volkswagen
10	peugeot
11	audi
12	saab
13	bmw
14	fiat
15	mercury
16	opel
17	fiat
18	oldsmobile
19	chrysler
20	mazda
21	volvo
22	renault
23	honda
24	subaru
25	capri
26	mercedes-benz
27	cadillac
28	triumph
29	nissan

In [24]:
```
data["brand"] = labelEncoder.fit_transform(data.brand)
```

In [25]:
```
temp_file["Encoded"] = pd.DataFrame(data["brand"].unique())
```

In [26]:
```
temp_file.to_csv("temp_file.csv")
```

In [27]:
```
temp_file
```

Out[27]:

	Brand	Encoded
0	chevrolet	6
1	buick	3
2	plymouth	22
3	amc	0
4	ford	11
5	pontiac	23
6	dodge	9
7	toyota	27
8	datsun	8
9	volkswagen	29
10	peugeot	21
11	audi	1
12	saab	25
13	bmw	2
14	fiat	16
15	mercury	17
16	opel	20
17	fiat	16
18	oldsmobile	19
19	chrysler	7
20	mazda	14
21	volvo	30
22	renault	24
23	honda	13
24	subaru	26
25	capri	5
26	mercedes-benz	15
27	cadillac	4
28	mercedes	15
29	triumph	28
30	nissan	18

In [28]:
```
def correlation(data, threshold):
    col_corr = set()
    corr_matrix = data.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i,j]) > threshold:
                colname = corr_matrix.columns[i]
                col_corr.add(colname)
    return col_corr
```

In [29]:
```
correlation(data,0.9)
```

Out[29]:
```
{'displacement', 'weight'}
```