

LAB 2 Exercise

Data Taranformation

In [27]:

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import StandardScaler,MinMaxScaler,OneHotEncoder
from sklearn.impute import SimpleImputer
Imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
```

In [28]:

```
cardata = pd.read_csv("D:/SEM7/ML/ML_Lab/Lab_2/CarData.csv" ,
                      usecols=['Price','Age','KM','FuelType','HP','MetColor','Automatic','CC','Doors','Weight'],
                      na_values=['????','??'])
cardata.head(10)
```

Out[28]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23.0	46986.0	Diesel	90.0	1.0	0	2000	three	1165
1	13750	23.0	72937.0	Diesel	90.0	1.0	0	2000	3	1165
2	13950	24.0	41711.0	Diesel	90.0	NaN	0	2000	3	1165
3	14950	26.0	48000.0	Diesel	90.0	0.0	0	2000	3	1165
4	13750	30.0	38500.0	Diesel	90.0	0.0	0	2000	3	1170
5	12950	32.0	61000.0	Diesel	90.0	0.0	0	2000	3	1170
6	16900	27.0	NaN	Diesel	NaN	NaN	0	2000	3	1245
7	18600	30.0	75889.0	NaN	90.0	1.0	0	2000	3	1245
8	21500	27.0	19700.0	Petrol	192.0	0.0	0	1800	3	1185
9	12950	23.0	71138.0	Diesel	NaN	NaN	0	1900	3	1105

Getting Information

In [29]:

```
cardata.info()
cardata.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Price       1436 non-null   int64
1   Age        1336 non-null   float64
2   KM         1421 non-null   float64
3   FuelType    1336 non-null   object
4   HP         1430 non-null   float64
5   MetColor    1286 non-null   float64
6   Automatic   1436 non-null   int64
7   CC         1436 non-null   int64
8   Doors       1436 non-null   object
9   Weight      1436 non-null   int64
dtypes: float64(4), int64(4), object(2)
memory usage: 112.3+ KB
```

Out[29]:

	Price	Age	KM	HP	MetColor	Automatic	
count	1436.000000	1336.000000	1421.000000	1430.000000	1286.000000	1436.000000	1436.
mean	10730.824513	55.672156	68647.239972	101.478322	0.674961	0.055710	1566.
std	3626.964585	18.589804	37333.023589	14.768255	0.468572	0.229441	187.
min	4350.000000	1.000000	1.000000	69.000000	0.000000	0.000000	1300.
25%	8450.000000	43.000000	43210.000000	90.000000	0.000000	0.000000	1400.
50%	9900.000000	60.000000	63634.000000	110.000000	1.000000	0.000000	1600.
75%	11950.000000	70.000000	87000.000000	110.000000	1.000000	0.000000	1600.
max	32500.000000	80.000000	243000.000000	192.000000	1.000000	1.000000	2000.

In [30]:

```
#remove rows of empty box values
cardata = cardata.drop(columns = ['Price'],axis=1)
cardata.head()
```

Out[30]:

	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	23.0	46986.0	Diesel	90.0	1.0	0	2000	three	1165
1	23.0	72937.0	Diesel	90.0	1.0	0	2000	3	1165
2	24.0	41711.0	Diesel	90.0	NaN	0	2000	3	1165
3	26.0	48000.0	Diesel	90.0	0.0	0	2000	3	1165
4	30.0	38500.0	Diesel	90.0	0.0	0	2000	3	1170

In [31]:

```
cardata.dropna(axis=0,how='any',subset=['MetColor','FuelType','Age'],inplace=True)
cardata.head(10)
```

Out[31]:

	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	23.0	46986.0	Diesel	90.0	1.0	0	2000	three	1165
1	23.0	72937.0	Diesel	90.0	1.0	0	2000	3	1165
3	26.0	48000.0	Diesel	90.0	0.0	0	2000	3	1165
4	30.0	38500.0	Diesel	90.0	0.0	0	2000	3	1170
5	32.0	61000.0	Diesel	90.0	0.0	0	2000	3	1170
8	27.0	19700.0	Petrol	192.0	0.0	0	1800	3	1185
10	25.0	31461.0	Petrol	192.0	0.0	0	1800	3	1185
11	22.0	43610.0	Petrol	192.0	0.0	0	1800	3	1185
12	25.0	32189.0	Petrol	192.0	0.0	0	1800	3	1185
13	31.0	23000.0	Petrol	192.0	1.0	0	1800	3	1185

Handling missing values

In [33]:

```
hpimputer = SimpleImputer(missing_values = np.nan,strategy='mean')
hpimputer = hpimputer.fit(cardata.iloc[:,3:4].values)
cardata.iloc[:,3:4] = hpimputer.transform(cardata.iloc[:,3:4].values)
kmimputer = SimpleImputer(missing_values = np.nan,strategy='mean')
kmimputer = kmimputer.fit(cardata.iloc[:,1:2].values)
cardata.iloc[:,1:2] = kmimputer.transform(cardata.iloc[:,1:2].values)
cardata.head(10)
```

Out[33]:

	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	23.0	46986.0	Diesel	90.0	1.0	0	2000	three	1165
1	23.0	72937.0	Diesel	90.0	1.0	0	2000	3	1165
3	26.0	48000.0	Diesel	90.0	0.0	0	2000	3	1165
4	30.0	38500.0	Diesel	90.0	0.0	0	2000	3	1170
5	32.0	61000.0	Diesel	90.0	0.0	0	2000	3	1170
8	27.0	19700.0	Petrol	192.0	0.0	0	1800	3	1185
10	25.0	31461.0	Petrol	192.0	0.0	0	1800	3	1185
11	22.0	43610.0	Petrol	192.0	0.0	0	1800	3	1185
12	25.0	32189.0	Petrol	192.0	0.0	0	1800	3	1185
13	31.0	23000.0	Petrol	192.0	1.0	0	1800	3	1185

In [34]:

```
#scale age
stdscale = MinMaxScaler()
age = cardata.iloc[:,0:1].values
cardata['Age']= stdscale.fit_transform(age)

#scale cc
stdscale = MinMaxScaler()
cc = cardata.iloc[:,6:7].values
cardata['CC']= stdscale.fit_transform(cc)

#scale weight
stdscale = MinMaxScaler()
weight = cardata.iloc[:,8:9].values
cardata['Weight']= stdscale.fit_transform(weight)

#scale km
stdscale = MinMaxScaler()
km = cardata.iloc[:,1:2].values
cardata['KM']= stdscale.fit_transform(km)

#scale hp
stdscale = MinMaxScaler()
hp = cardata.iloc[:,3:4].values
cardata['HP']= stdscale.fit_transform(hp)
cardata.head()
```

Out[34]:

	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	0.278481	0.193355	Diesel	0.170732	1.0	0	1.0	three	0.268293
1	0.278481	0.300149	Diesel	0.170732	1.0	0	1.0	3	0.268293
3	0.316456	0.197528	Diesel	0.170732	0.0	0	1.0	3	0.268293
4	0.367089	0.158433	Diesel	0.170732	0.0	0	1.0	3	0.276423
5	0.392405	0.251026	Diesel	0.170732	0.0	0	1.0	3	0.276423

In [35]:

```
cardata.describe()
```

Out[35]:

	Age	KM	HP	MetColor	Automatic	CC	Weight
count	1111.000000	1111.000000	1111.000000	1111.000000	1111.000000	1111.000000	1111.000000
mean	0.690962	0.285123	0.266165	0.675068	0.053105	0.384948	0.120114
std	0.238344	0.155902	0.122004	0.468561	0.224344	0.265622	0.085546
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.531646	0.179935	0.170732	0.000000	0.000000	0.142857	0.073171
50%	0.746835	0.262882	0.333333	1.000000	0.000000	0.428571	0.113821
75%	0.873418	0.360897	0.333333	1.000000	0.000000	0.428571	0.146341
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Handling doors column

In [36]:

```
cardata['Doors'] = cardata['Doors'].replace('three',3)
cardata['Doors'] = cardata['Doors'].replace('four',4)
cardata['Doors'] = cardata['Doors'].replace('five',5)
cardata.head(10)
```

Out[36]:

	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	0.278481	0.193355	Diesel	0.170732	1.0	0	1.000000	3	0.268293
1	0.278481	0.300149	Diesel	0.170732	1.0	0	1.000000	3	0.268293
3	0.316456	0.197528	Diesel	0.170732	0.0	0	1.000000	3	0.268293
4	0.367089	0.158433	Diesel	0.170732	0.0	0	1.000000	3	0.276423
5	0.392405	0.251026	Diesel	0.170732	0.0	0	1.000000	3	0.276423
8	0.329114	0.081066	Petrol	1.000000	0.0	0	0.714286	3	0.300813
10	0.303797	0.129466	Petrol	1.000000	0.0	0	0.714286	3	0.300813
11	0.265823	0.179462	Petrol	1.000000	0.0	0	0.714286	3	0.300813
12	0.303797	0.132461	Petrol	1.000000	0.0	0	0.714286	3	0.300813
13	0.379747	0.094646	Petrol	1.000000	1.0	0	0.714286	3	0.300813

In [37]:

```
cardata['Doors'] = cardata['Doors'].astype(int)
cardata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1111 entries, 0 to 1435
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Age         1111 non-null   float64
1   KM          1111 non-null   float64
2   FuelType    1111 non-null   object
3   HP          1111 non-null   float64
4   MetColor    1111 non-null   float64
5   Automatic   1111 non-null   int64
6   CC          1111 non-null   float64
7   Doors       1111 non-null   int32
8   Weight      1111 non-null   float64
dtypes: float64(6), int32(1), int64(1), object(1)
memory usage: 82.5+ KB
```

feature selection

In [39]:

```
mat = cardata.corr()
mat
```

Out[39]:

	Age	KM	HP	MetColor	Automatic	CC	Doors	Weight
Age	1.000000	0.520458	-0.166037	-0.108185	0.031442	-0.092548	-0.172241	-0.45327
KM	0.520458	1.000000	-0.366112	-0.105365	-0.054317	0.316058	-0.078913	-0.02995
HP	-0.166037	-0.366112	1.000000	0.065443	0.019653	0.038990	0.064210	0.08660
MetColor	-0.108185	-0.105365	0.065443	1.000000	-0.007105	0.015111	0.098079	0.05895
Automatic	0.031442	-0.054317	0.019653	-0.007105	1.000000	-0.058839	0.001188	0.06347
CC	-0.092548	0.316058	0.038990	0.015111	-0.058839	1.000000	0.119031	0.62821
Doors	-0.172241	-0.078913	0.064210	0.098079	0.001188	0.119031	1.000000	0.30057
Weight	-0.453275	-0.029959	0.086601	0.058956	0.063479	0.628212	0.300571	1.00000

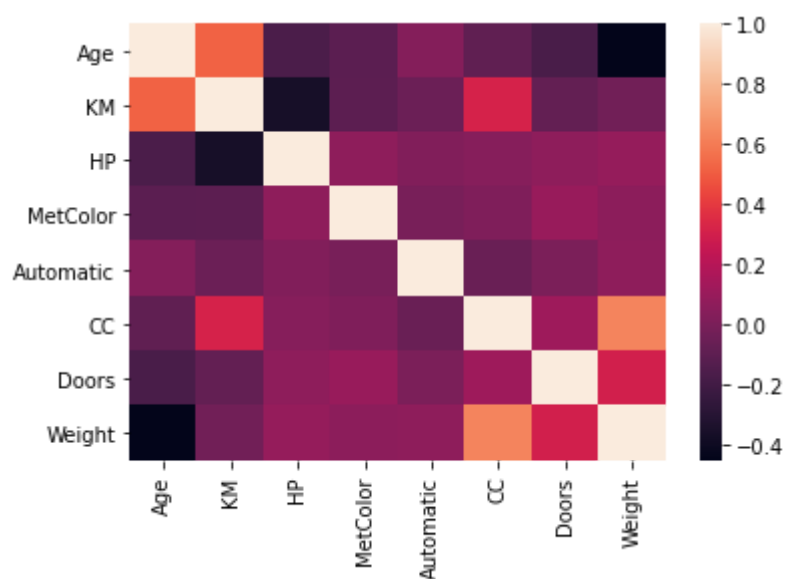


In [40]:

```
sns.heatmap(mat)
```

Out[40]:

<matplotlib.axes._subplots.AxesSubplot at 0x2e7e6454ac8>



In [43]:

```
cols = np.full((mat.shape[0],), True, dtype=bool)
for i in range(mat.shape[0]):
    for j in range(i+1, mat.shape[0]):
        if mat.iloc[i,j] >= 0.6:
            if cols[j]:
                cols[j] = False
```

In [44]:

```
dummy = pd.get_dummies(car_data['FuelType'])
cardata = cardata.drop('FuelType',axis=1)
selected_cols = cardata.columns[cols]
cardata = pd.concat([dummy,cardata[selected_cols]],axis=1)
cardata.head(10)
```

Out[44]:

	CNG	Diesel	Petrol	Age	KM	HP	MetColor	Automatic	CC	Doors
0	0	1	0	0.278481	0.193355	0.170732	1.0	0.0	1.000000	3.0
1	0	1	0	0.278481	0.300149	0.170732	1.0	0.0	1.000000	3.0
2	0	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	0	1	0	0.316456	0.197528	0.170732	0.0	0.0	1.000000	3.0
4	0	1	0	0.367089	0.158433	0.170732	0.0	0.0	1.000000	3.0
5	0	1	0	0.392405	0.251026	0.170732	0.0	0.0	1.000000	3.0
6	0	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	0	0	1	0.329114	0.081066	1.000000	0.0	0.0	0.714286	3.0
9	0	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In []: