

Importing the libraries

```
In [8]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
```

Importing the data

```
In [9]: data = pd.read_csv('car performance.csv')
```

```
In [10]: data
```

	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 [11]: 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 [12]: data.isnull().sum()
```

```
Out[12]: mpg              0
cylinders              0
displacement          0
horsepower            0
weight                0
acceleration          0
model year            0
origin                0
car name              0
dtype: int64
```

```
In [152... l = []
for i in data["car name"]:
    l.append(i.split(' ')[0])
```

```
In [153... data.insert(9,"Brand",l)
```

Handling Irrelevant Values

```
In [154... make_typo_correction = {
    'vw': 'volswagen',
    'chevy': 'chevrolet',
    'maxda': 'mazda',
    'vokswagen': 'volkswagen',
    'toyouta': 'toyota',
    'chevroelt': 'chevrolet'
}

data['Brand'] = data['Brand'].replace(make_typo_correction)
```

```
In [155... data.Brand.unique()
```

```
Out[155]: array(['chevrolet', 'buick', 'plymouth', 'amc', 'ford', 'pontiac',
        'dodge', 'toyota', 'datsum', 'volkswagen', 'peugeot', 'audi',
        'saab', 'bmw', 'hi', 'mercury', 'opel', 'fiat', 'oldsmobile',
        'chrysler', 'mazda', 'volvo', 'renault', 'honda', 'subaru',
        'capri', 'mercedes-benz', 'cadillac', 'mercedes', 'triumph',
        'nissan'], dtype=object)
```

```
In [156... temp_file = pd.DataFrame(data.Brand.unique(),columns=["Brand"])
```

```
In [157... data.drop('car name',axis=1,inplace=True)
```

Label Encoding

```
In [158... from sklearn.preprocessing import LabelEncoder
```

```
In [160... temp_file
```

	Brand
0	chevrolet
1	buick
2	plymouth
3	amc
4	ford
5	pontiac
6	dodge
7	toyota
8	datsum
9	volkswagen
10	peugeot
11	audi
12	saab
13	bmw
14	hi
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	mercedes
29	triumph
30	nissan

```
In [161... data["Brand"] = LabelEncoder.fit_transform(data,data["Brand"])
```

D:\Anaconda\lib\site-packages\sklearn\preprocessing_label.py:117: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access

```
self.classes_, y = _unique(y, return_inverse=True)
```

```
In [162... temp_file["Encoded"] = pd.DataFrame(data["Brand"].unique())
```

```
In [163... temp_file.to_csv('Temp_file.csv')
```

```
In [164... temp_file
```

	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	datsum	8
9	volkswagen	29
10	peugeot	21
11	audi	1
12	saab	25
13	bmw	2
14	hi	12
15	mercury	17
16	opel	20
17	fiat	10
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	16
27	cadillac	4
28	mercedes	15
29	triumph	28
30	nissan	18

```
In [166... def correlation(car, threshold):
    col_corr = set()
    corr_matrix = car.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 [168... correlation(data,0.9)
```

```
Out[168]: {'displacement', 'weight'}
```