

Car Price Predication

Note:

Part 1 Contains :

1. Data cleaning & preprocessing

2. EDA

3. Feature engineering

Part 2 Contains Machine Learning Model Building

- Importing require library for performing EDA, Data Wrangling and data cleaning

```
In [1]: import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # plotting package
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')
```

```
In [2]: # Importing dataset excel file using pandas.
df= pd.read_excel('Final Used Car Scrap data .xlsx')
```

```
In [3]: print('No. of Rows :',df.shape[0])
print('No. of Columns :',df.shape[1])
pd.set_option('display.max_columns',None) # # This will enable us to see truncated
df.head()
```

No. of Rows : 10544
No. of Columns : 24

Out[3]:

	Car Model	Make Year	Fuel Type	KMs driven	Engine Displacement(CC)	Transmission	Milage(kmpl)	Max Power(bhp)	1
0	Honda City i VTEC SV	2014	Petrol	18,600	1497	Manual	17.4	117.3	
1	Jeep Compass Limited Option	2.0	Diesel	15,000	1956	Manual	17.1	170	
2	Chevrolet Tavera B2-7 seats BSIII	2008	Diesel	1,15,000	2499	Manual	14.8	80	
3	Honda City S	2008	Petrol	80,000	1497	Manual	16.8	116.4	
4	Hyundai i10 Sportz Option	2012	Petrol	35,000	1197	Manual	20.36	78.9	

In [4]: `# As we have 24 columns Lets sort columns by their datatype
df.columns.to_series().groupby(df.dtypes).groups`

Out[4]: {int64: ['Make Year'], float64: ['No of Cylinder'], object: ['Car Model', 'Fuel Type', 'KMs driven', 'Engine Displacement(CC)', 'Transmission', 'Milage(kmpl)', 'Max Power(bhp)', 'Torque(Nm)', 'Seating Capacity', 'Color', 'Gear Box', 'Steering Type', 'Front Brake Type', 'Rear Brake Type', 'Tyre Volume', 'Engine Type', 'Turbo Charger', 'Super Charger', 'Length(mm)', 'Width(mm)', 'Height(mm)', 'Price(Rs)']}

Observation:

- This dataset contain 10544 rows and 24 columns.
- We have lot feature here with object datatypes which actually have to be integer type.
- Our target feature is Price.

Feature engineering to convert feature in appropriate datatypes.

1. Transforming datatypes of KMs driven into int type.

In [5]: `df['KMs driven'] = df['KMs driven'].map(lambda x : x.split(' ')[0])`

In [6]: `df['KMs driven'] = df['KMs driven'].map(lambda x : x.replace(',', ''))`

In [7]: `df['KMs driven']=pd.to_numeric(df['KMs driven'])`

```
In [8]: df['KMs driven'].dtypes
```

```
Out[8]: dtype('int64')
```

2. Transforming datatypes of Engine Displacement(CC) into int type.

```
In [9]: df['Engine Displacement(CC)'] = df['Engine Displacement(CC)'].map(lambda x : x.replace('cc', ''))
```

```
In [10]: df['Engine Displacement(CC)'] = pd.to_numeric(df['Engine Displacement(CC)'])
```

```
In [11]: df['Engine Displacement(CC)'].dtypes
```

```
Out[11]: dtype('int64')
```

3. Transforming datatypes of 'Milage(kmpl)' into float type.

```
In [12]: df['Milage(kmpl)'] = df['Milage(kmpl)'].map(lambda x : x.replace('km/kg', ''))
```

```
In [13]: df['Milage(kmpl)'] = df['Milage(kmpl)'].map(lambda x : x.replace('-', ''))
```

```
In [14]: df['Milage(kmpl)'] = df['Milage(kmpl)'].map(lambda x : x.replace('km/hr', ''))
```

```
In [15]: df['Milage(kmpl)'] = pd.to_numeric(df['Milage(kmpl)'])
```

```
In [16]: df['Milage(kmpl)'].dtypes
```

```
Out[16]: dtype('float64')
```

4. Transforming datatypes of Max Power into float type.

```
In [17]: df['Max Power(bhp)'] = df['Max Power(bhp)'].replace('-', '')
```

```
In [18]: df['Max Power(bhp)'] = df['Max Power(bhp)'].replace('null ', '')
```

Some of values in Max power are with unit PS, which need to convert into brake horse power.

```
In [19]: df['Max Power(bhp)'] = df['Max Power(bhp)'].replace('118PS at 6,600 rpm', '116.3')
```

```
In [20]: df['Max Power(bhp)'] = df['Max Power(bhp)'].replace('80 PS at 5200 rpm', '78.9')
```

Another error occur here, ValueError: Unable to parse string "55 Bhp" at position 6513

```
In [21]: df['Max Power(bhp)'] = df['Max Power(bhp)'].replace('55 Bhp', '55')
```

Another error occur here, ValueError: Unable to parse string "85ps" at position 6697

```
In [22]: df['Max Power(bhp)'] = df['Max Power(bhp)'].replace('85ps', '83.8')
```

```
In [23]: df['Max Power(bhp)'] = pd.to_numeric(df['Max Power(bhp)'])
```

```
In [24]: df['Max Power(bhp)'].dtypes
```

```
Out[24]: dtype('float64')
```

5. Transforming datatypes of Torque into float type.

```
In [25]: df['Torque(Nm)'] = df['Torque(Nm)'].map(lambda x : x.split('@')[0])
```

```
In [26]: df['Torque(Nm)'] = df['Torque(Nm)'].map(lambda x : x.rstrip('nm'))
```

Some of values are in units kgm, which need to convert into Nm. For example, 13.9 kgm at 4200 rp

```
In [27]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('13.9 kgm at 4200 rp', '136.32')
```

```
In [28]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('-', '')
```

```
In [29]: df['Torque(Nm)'] = df['Torque(Nm)'].map(lambda x : x.rstrip('NM'))
```

Another entry with 22.4 kgm at 1750-2750rp

```
In [30]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('22.4 kgm at 1750-2750rp', '219.66')
```

Another error occur while converting datatype - ValueError: Unable to parse string "260 at 1800-2200 rp" at position 340

```
In [31]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('260 at 1800-2200 rp', '260')
```

```
In [32]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('180 at 1440-1500rp', '180')
```

```
In [33]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('146 at 4800 rp', '146')
```

```
In [34]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('90.3 at 4,200 rp', '90.3')
```

Another error here, ValueError: Unable to parse string "12.5kg" at position 1164

```
In [35]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('12.5kg', '122.58')
```

Another error here, ValueError: Unable to parse string "11.4 kgm at 4,000 rp" at position 733

```
In [36]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('11.4 kgm at 4,000 rp', '111.79')
```

```
In [37]: df['Torque(Nm)'] = df['Torque(Nm)'].map(lambda x : x.rstrip('at'))
```

```
In [38]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('153 at 3800 rp','153')
```

Another error here, ValueError: Unable to parse string "24 KGM at 1900-2750 RP" at position 1391

```
In [39]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('24 KGM at 1900-2750 RP','235.36')
```

```
In [40]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('190 at 1750 rp','190')
```

Another error here, ValueError: Unable to parse string "13.1kg" at position 1620

```
In [41]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('13.1kg','128.46')
```

```
In [42]: df['Torque(Nm)'] = df['Torque(Nm)'].map(lambda x : x.split('at')[0])
```

```
In [43]: df['Torque(Nm)'] = df['Torque(Nm)'].map(lambda x : x.rstrip('(11.2)'))
```

Another error here, ValueError: Unable to parse string "11.3kg" at position 2269

```
In [44]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('11.3kg','110.81')
```

```
In [45]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('210 / 1900','210')
```

```
In [54]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('6.1kg','59.82')
```

```
In [47]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('18.5kg-','181.42')
```

```
In [48]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('400 /2000 rp','400')
```

```
In [49]: df['Torque(Nm)'] = df['Torque(Nm)'].replace('14.9 KGM ','146.12')
```

```
In [55]: df['Torque(Nm)'] = pd.to_numeric(df['Torque(Nm)'])
```

```
In [56]: df['Torque(Nm)'].dtypes
```

```
Out[56]: dtype('float64')
```

6. Transforming datatypes of Seating capacity into float type.

```
In [57]: df['Seating Capacity'] = df['Seating Capacity'].replace('-', '')
```

```
In [58]: df['Seating Capacity'] = pd.to_numeric(df['Seating Capacity'])
```

```
In [59]: df['Seating Capacity'].dtypes
```

```
Out[59]: dtype('float64')
```

8. Transforming datatypes of Seating length into float type.

```
In [60]: df['Length(mm)'] = df['Length(mm)'].str.replace(',', '', '')
```

```
In [61]: df['Length(mm)'] = df['Length(mm)'].str.replace('-', '', '')
```

```
In [62]: df['Length(mm)'] = pd.to_numeric(df['Length(mm)'])
```

```
In [63]: df['Length(mm)'].dtypes
```

```
Out[63]: dtype('float64')
```

9. Transforming datatypes of Seating Width(mm) into float type.

```
In [64]: df['Width(mm)'] = df['Width(mm)'].str.replace(',', '', '')
```

```
In [65]: df['Width(mm)'] = df['Width(mm)'].str.replace('-', '', '')
```

```
In [66]: df['Width(mm)'] = pd.to_numeric(df['Width(mm)'])
```

```
In [67]: df['Width(mm)'].dtypes
```

```
Out[67]: dtype('float64')
```

10. Transforming datatypes of Seating Height(mm) into float type.

```
In [68]: df['Height(mm)'] = df['Height(mm)'].str.replace(',', '', '')
```

```
In [69]: df['Height(mm)'] = df['Height(mm)'].str.replace('-', '', '')
```

```
In [70]: df['Height(mm)'] = pd.to_numeric(df['Height(mm)'])
```

```
In [71]: df['Height(mm)'].dtypes
```

```
Out[71]: dtype('float64')
```

11. Transforming datatypes of Price(Rs) into float type.

```
In [72]: df['Price(Rs)'] = df['Price(Rs)'].str.replace('Lakh*', '100000')
df['Price(Rs)'] = df['Price(Rs)'].str.replace('Cr*', '100000')
df['Price(Rs)'] = df['Price(Rs)'].str.replace(',', '')
```

```
In [73]: df['Price(Rs)'] = df['Price(Rs)'].str.replace('*', '')
```

```
In [74]: df[['a', 'b']] = df['Price(Rs)'].str.split(expand=True)
df['a'] = df['a'].astype("float")
df['b'] = df['b'].astype("float")
```

```
In [75]: df['b'] = df['b'].fillna(value = 1)
df["Price (Rs.)"] = df['a'] * df['b']
```

```
In [76]: df.drop(columns=['Price(Rs)', 'a', 'b'], inplace = True)
```

11. Feature Engineering on Make Year column to extract Age of Car

```
In [77]: df['Car_Age'] = 2021 - df['Make Year']
```

```
In [78]: df.drop(columns=['Make Year'], inplace = True)
```

- **Feature Engineering on Car Model**

```
In [81]: df['Car_Brand'] = df["Car Model"].str.split(' ').str[:2]
df['Car_Brand'] = df['Car_Brand'].apply(lambda x: ','.join(map(str, x)))
df['Car_Brand'] = df['Car_Brand'].str.replace(',', ' ')
df['Car_Model'] = df["Car Model"].str.split(' ').str[2:]
df['Car_Model'] = df['Car_Model'].apply(lambda x: ','.join(map(str, x)))
df['Car_Model'] = df['Car_Model'].str.replace(',', ' ')
```

```
In [82]: df.drop(columns = 'Car Model', inplace = True)
```

- **Let check datatypes after converting into appropriate datatypes.**

```
In [83]: # As we have 24 columns Lets sort columns by their datatype after conversion into a
df.columns.to_series().groupby(df.dtypes).groups
```

```
Out[83]: {int64: ['KMs driven', 'Engine Displacement(CC)', 'Car_Age'], float64: ['Milage(kmpl)', 'Max Power(bhp)', 'Torque(Nm)', 'Seating Capacity', 'No of Cylinder', 'Length(mm)', 'Width(mm)', 'Height(mm)', 'Price (Rs.)'], object: ['Fuel Type', 'Transmission', 'Color', 'Gear Box', 'Steering Type', 'Front Brake Type', 'Rear Brake Type', 'Tyre Volume', 'Engine Type', 'Turbo Charger', 'Super Charger', 'Car_Brand', 'Car_Model']}
```

Data Integrity Check

```
In [84]: # Splitting data in Numeric and categorical Variable.
Numerical = ['KMs driven', 'Engine Displacement(CC)', 'Car_Age', 'Milage(kmpl)', 'Max Power(bhp)', 'Torque(Nm)', 'Seating Capacity', 'No of Cylinder', 'Length(mm)', 'Width(mm)', 'Height(mm)', 'Price (Rs.)']
Categorical = ['Fuel Type', 'Transmission', 'Color', 'Gear Box', 'Steering Type', 'Front Brake Type', 'Rear Brake Type', 'Tyre Volume', 'Engine Type', 'Turbo Charger', 'Super Charger', 'Car_Brand', 'Car_Model']
```

- **Dataset can contain whitespaces, missing value, duplicates entries, let investigate integrity of data before proceeding for further analysis.**

```
In [85]: df.duplicated().sum() # This will detect duplicate entries in dataset
```

```
Out[85]: 1658
```

- Dropping duplicates entries from dataset.

```
In [88]: # Droping duplicate entries  
df.drop_duplicates(keep='last', inplace =True)
```

```
In [89]: df.shape
```

```
Out[89]: (8886, 25)
```

Check for presence of any whitespaces, '?', 'NA', ' - ', 'null' in dataset

```
In [90]: df.isin(['????', '?????', ' - ', 'null', 'NA', ' ']).sum().any()
```

```
Out[90]: True
```

```
In [91]: df.replace(' - ',np.nan, inplace = True)  
df.replace('null',np.nan, inplace= True)  
df.replace('????', np.nan, inplace = True)  
df.replace('?????', np.nan, inplace = True)  
df.replace(' ', np.nan, inplace = True)
```

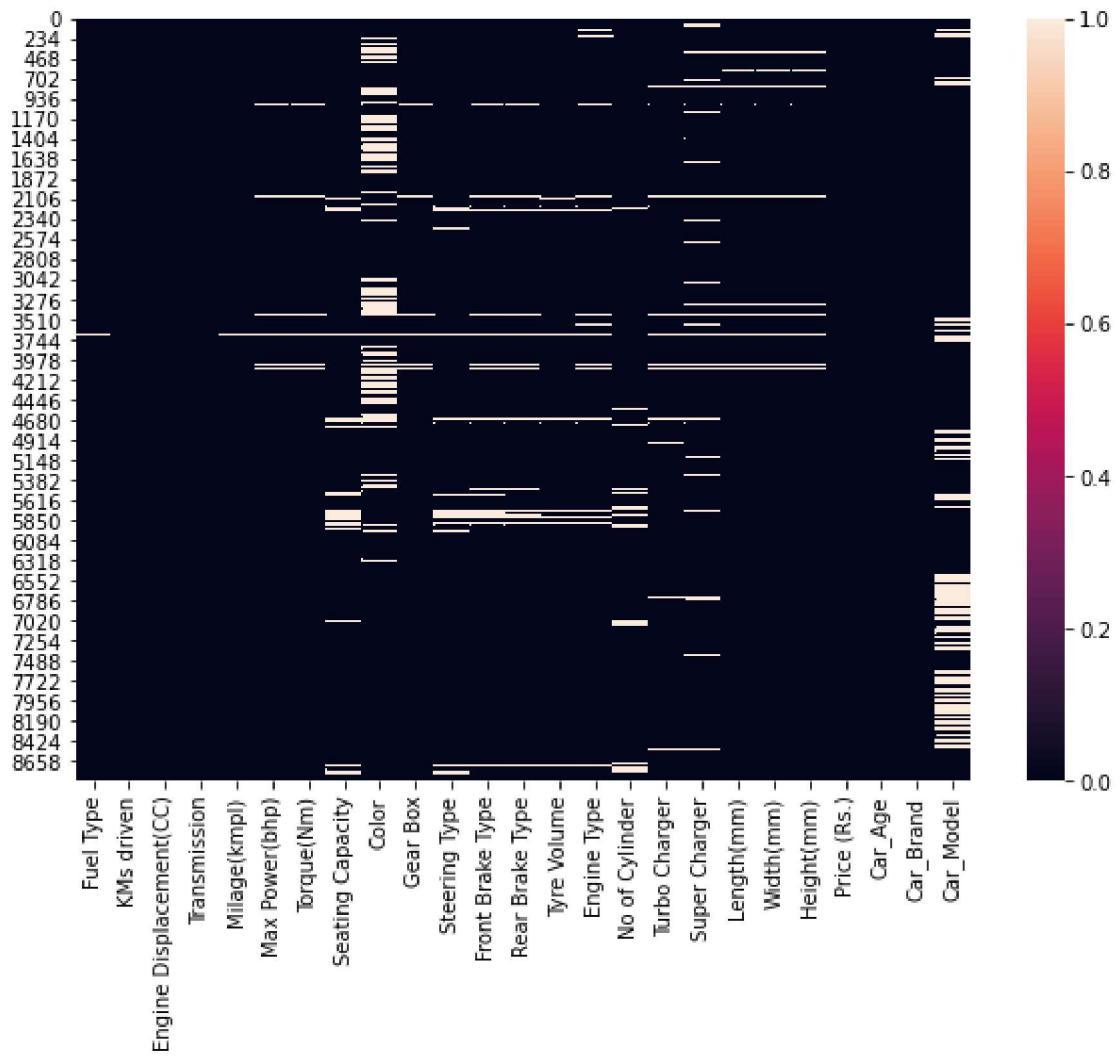
```
In [92]: df.to_excel('clean data phase1.xlsx',index =False)
```

```
In [125...]: df = pd.read_excel('clean data phase1.xlsx')
```

```
In [126...]: df.isnull().sum()
```

```
Out[126]: Fuel Type          26
KMs driven                  0
Engine Displacement(CC)    0
Transmission                0
Milage(kmpl)                27
Max Power(bhp)              121
Torque(Nm)                  145
Seating Capacity            487
Color                        1860
Gear Box                     128
Steering Type                453
Front Brake Type             483
Rear Brake Type              437
Tyre Volume                 350
Engine Type                  378
No of Cylinder               403
Turbo Charger                266
Super Charger                373
Length(mm)                   176
Width(mm)                    173
Height(mm)                   172
Price (Rs.)                  0
Car_Age                      0
Car_Brand                     0
Car_Model                     1660
dtype: int64
```

```
In [127...]: sns.set_palette('cool_r')
plt.figure(figsize=(10,7))
sns.heatmap(df.isnull())
plt.show()
```



Before Going for imputation of missing values,

Let first Enlist different subcategories in categorical variable inorder to check presense of similar or repeated category with different sub category name.

```
In [128]: pd.set_option('display.max_rows',None)
for i in Categorical :
    print(i)
    print(df[i].value_counts())
    print('='*100)
```

Fuel Type

Petrol	4767
Diesel	3981
CNG	67
LPG	41
Electric	4

Name: Fuel Type, dtype: int64

Transmission

Manual	7457
Automatic	1429

Name: Transmission, dtype: int64

Color

White	1943
Other	1199
Silver	1052
Grey	870
Red	541
Blue	336
Black	272
Brown	172
Maroon	72
Golden	69
Orange	56
Green	25
SILKY SILVER	23
Gold	22
Star Dust	13
Purple	11
Others	11
Yellow	9
Silky silver	9
Superior white	8
Beige	7
Gray	6
Polar White	6
BEIGE	6
Granite Grey	6
Violet	5
MODERN STEEL METALLIC	4
Arabian Blue Crystal Effect	4
Candy White	4
beige	4
Golden Brown	4
Fiery Red	4
MET ECRU BEIGE	4
Planet grey	4
C White	4
R RED	4
Taffeta White	4
Blazing Red	3
TITANIUM GREY	3
S cooper	3

TAFETA WHITE	3
Blade Silver	3
Arctic white	3
BLAZING RED	3
Metallic silky silver	3
Sleek silver	3
New Granite Grey	3
PLATINUM SILVER	3
S white	3
Metallic Pearl Arctic White	3
R silver metall	3
Urban Titanium	3
Cherry	3
MIDNIGHT BLACK	3
Magma grey	3
Urban titanium	3
I Sand	2
Fire Red	2
Modern Steel Metallic	2
P Met Arc White	2
Z Blue	2
Outback bronze	2
Espresso Brown	2
Raw Copper	2
Metallic Glistening Grey	2
Chocolate	2
Solid White	2
P White	2
Urban Titanium Metallic	2
P Black	2
CARNELIAN RED PEARL	2
P A White	2
Twilight Blue	2
PA White	2
StarDust	2
modern steel	2
Sleek Silver	2
Outback Bronze	2
Deep Sea Blue BMW X	2
Brown Metalic	2
Brown/Biege	2
Bright Red	2
Pearl Supreme White	2
Beach	2
Chill Met	2
Silksilver	2
Mordern Steel	2
Cappuchino Beige	2
Polar white	2
Granite gray	2
Creamy White	1
I c White	1
Stone beige	1
P METAL	1
White orchid pearl	1
SDUST	1

Wine Red	1
Brownse	1
Smoke grey	1
Zanskar blue	1
Premium Amber Metalic	1
Royal Gold	1
SBrown	1
Typhoon Silver	1
F Red	1
Platinum Silver	1
Green cocktail	1
golden brown	1
Magna Grey	1
Dignity brown	1
Dignity Brown	1
Red psn	1
Red/Black	1
PM Artic White	1
Diamond	1
Harmony Beige	1
Golden Brown M	1
P c blue	1
Ivory White	1
P white	1
Ice Cool White	1
V Red	1
O Bronze	1
Symphony silver	1
silver	1
Passion Red	1
Auto grey Metallic	1
Matte brown	1
Silky breeze	1
Aquatea blue	1
Crystal white	1
Satin Gold	1
Punchy orange ?	1
B SILVER	1
Starry Black	1
Arctic White	1
Garnet Red	1
Terra Beige	1
Dark grey	1
brown	1
Sum White	1
Pristine white	1
Pa white	1
magma grey	1
Sunlight copper	1
Sky Blue	1
Ice White	1
Blue Water met	1
Aqua Teal	1
Elephant grey	1
Icy blue	1
Silky Silver	1

Light blue	1
carbon steel	1
Stardust	1
Light green	1
Vermilion Red	1
Cream Color	1
Sky blue	1
B Silk	1
Chill	1
G RED	1
P silver	1
Carbon Steel	1
Premium Silver Metallic	1
Chill Green	1
S White	1
S L Copper	1
Prmestargaze	1
Meroon	1
R passion	1
P Fantasia	1
Moose Grey	1
Fiery Yellow	1
BRONZE	1
TEALITE BLUE	1
Solid Bright Red	1
Metallic colour	1
Modern Steel Metalic	1
Bronze	1
Litish Golden	1
Magma Grey	1
VERMONT RED	1
Candy Whiite	1
Papaya Orange	1
Pearl white	1
CHAMPAGNE	1
Fairy red	1
T Brown Met	1
Purpal	1
Sun Copper	1
Bold Beige	1
Black Sapphire Metallic	1
Light Brown	1
AVANT GRADE BRONZE	1
Mehroom	1
Apple Green	1
Army Colour	1

Name: Color, dtype: int64

Gear Box

5 Speed	6324
6 Speed	1005
5	205
5-Speed	205
4 Speed	202
8 Speed	179

7 Speed	174
6-Speed	58
5 speed	58
9 Speed	50
CVT	47
4-Speed	38
7-Speed S-Tronic	16
7-Speed DCT	11
7 Speed CVT	11
Five Speed Manual Transmission	11
7-Speed	11
5 Speed CVT	11
7 Speed DSG	8
5 Speed AT+ Paddle Shifters	8
8-Speed	8
5-speed	8
6 speed	7
5 Manual	7
8 Speed Sport	6
7 Speed DCT	6
Five Speed	6
7 Speed 7G-DCT	5
7G DCT 7-Speed Dual Clutch Transmission	5
5 Speed Manual	4
6-Speed IVT	4
7-speed Stronic	4
8 Speed Tiptronic	4
9 speed Tronic	4
6-speed DCT	3
6-speed AutoSHIFT	3
Fully Automatic	3
5 Speed Manual Transmission	3
Five Speed Manual Transmission Gearbox	3
8 Speed Tip Tronic S	2
9G-TRONIC	2
5 Speed Manual (Cable Type Gear Shift)	2
6 Speed Automatic	2
6 Speed MT	2
6-Speed IMT	2
6-speed	2
10 speed	2
8-Speed Steptronic	2
Six Speed Gearbox	2
7 Speed S tronic	1
8-Speed DCT	1
5 GearBox	1
6 Speed AT	1
6	1
5 Gears	1
7-Speed DSG	1
Six Speed Manual with Paddle Shifter	1
8 Speed Steptronic	1
6-Speed DCT	1
AGS	1
Automatic Transmission	1
6 Speed iMT	1

Name: Gear Box, dtype: int64

Steering Type

Power	7605
Electric	472
Manual	230
Electrical	80
EPAS	22
Electronic	14
Hydraulic	6
power	4

Name: Steering Type, dtype: int64

Front Brake Type

Disc	4575
Ventilated Disc	3288
Solid Disc	188
Ventilated Discs	94
Disc & Caliper Type	54
Ventilated DIsc	49
Disk	37
Drum	22
Ventilated Disk	20
Disc, 236 mm	13
Ventilated discs	13
Vantilated Disc	12
Ventlated Disc	9
Ventillated Disc	5
Vacuum assisted hydraulic dual circuit w	4
Ventillated Discs	3
disc	3
264mm Ventilated discs	3
Booster assisted ventilated disc	3
Multilateral Disc	1
Tandem master cylinder with Servo assist	1
Ventilated disc	1
Discs	1
Disc,internally ventilated	1
Single Piston Sliding Caliper, Vented Di	1
260mm disc	1
Disc brakes	1

Name: Front Brake Type, dtype: int64

Rear Brake Type

Drum	7269
Disc	671
Ventilated Disc	105
Solid Disc	85
Disc & Caliper Type	54
Leading-Trailing Drum	53
Self-Adjusting Drum	51
Discs	31
Drums	22

Ventilated Discs	18
Disc & Drum	16
Drums 180 mm	13
Self Adjusting Drums	10
Ventilated discs	8
Self Adjusting Drum	7
Self adjusting drums	5
Vacuum assisted hydraulic dual circuit w	4
262mm Disc & Drum Combination	3
Drum in disc	3
self adjusting drums	3
Leading & Trailing Drum	3
Booster assisted drum	3
drum	3
Self adjusting Drums	3
Ventilated Drum	2
228.6 mm dia, drums on rear wheels	1
203mm Drums	1
Single Piston Sliding Caliper, Vented Di	1
Drum in Discs	1

Name: Rear Brake Type, dtype: int64

Tyre Volume

Tubeless,Radial	5379
Tubeless	1530
Tubeless Tyres	603
Tubeless, Radial	601
Radial	125
Tubeless Tyres, Radial	91
Tubeless,Radials	37
Tubeless Radial Tyres	31
Radial, Tubless	29
Tubless, Radial	29
Radial Tubeless	18
Runflat	13
Tubless,Radial	13
Tubeless Radial	6
Runflat Tyre	5
Radial,Tubeless	5
tubeless tyre	4
Runflat Tyres	4
Run-Flat	3
Radial Tyres	3
Tubeless Radial Tyre	2
Tubeless,Runflat	1
Tubless Radial Tyrees	1
Tubeless Radials	1
Tubeless, Runflat	1
Radial, Tubeless	1

Name: Tyre Volume, dtype: int64

Engine Type

In-Line Engine	834
DDiS Diesel Engine	512

Petrol Engine	479
Kappa VTVT Petrol Engine	459
K Series Petrol Engine	369
Diesel Engine	242
F8D Petrol Engine	212
TDCi Diesel Engine	200
K10B Petrol Engine	193
K Series VVT Engine	193
U2 CRDI Diesel Engine	184
i-VTEC Petrol Engine	167
DDiS 200 Diesel Engine	166
mHawk Diesel Engine	148
TDI Diesel Engine	147
VVT Petrol Engine	128
VTVT Engine	120
K10B Engine	110
VTVT Petrol Engine	106
DOHC Kappa2 Engine	103
U2 CRDi Diesel Engine	99
k14B VVT Engine	88
i-VTEC Engine	86
D-4D Diesel Engine	85
i VTEC Engine	85
Revotron Engine	83
DDiS 200 Engine	79
i-DTEC Diesel Engine	78
DDiS 190 Engine	75
MPI Petrol Engine	72
IRDE2 Engine	71
Ti-VCT Petrol Engine	70
U2 CRDI VGT Engine	62
dCi Diesel Engine	60
Kappa Petrol Engine	57
1.2-Litre VVT Petrol Engine	55
i DTEC Diesel Engine	48
K9K Diesel Engine	45
In-Line Diesel Engine	44
1.2L VVT Engine	43
In Line Petrol Engine	40
1-GD FTV Engine	38
In-Line Petrol Engine	38
2KD-FTV Diesel Engine	34
Smartech Engine	33
TSI Petrol Engine	33
Turbocharged Diesel Engin	33
SOHC i-VTEC	32
mHawk CRDe Engine	31
dCi THP Diesel Engine	30
2-GD FTV Engine	30
Common Rail Diesel Engine	29
Hi-Tech Petrol Engine	29
1.0L Petrol Engine	28
0.8L Petrol Engine	28
DOHC Engine	28
Diesel	27
FC Petrol Engine	25

mFALCON D75 Engine	25
U2 VGT Diesel Engine	25
M2DiCR Diesel Engine	24
1.2 l Kappa petrol	23
U2 CRDI Engine	23
1.2-Litre VVT Petrol Engi	22
mCR100 Diesel Engine	20
mHawk 100 Diesel Engine	20
4 Cylinders In-line	20
D13A SHVS Engine	20
DICOR Engine	19
SMARTech Diesel Engine	19
2.0-Litre 4-Cyl Multijet	18
Kappa 1.0 L Turbo GDi Petrol	18
Duratorq Diesel Engine	17
1.5 Dual VVT-i Engine	17
2ZR-FE Petrol Engine	17
Revotorq Engine	17
1.1 L Petrol	16
Quadrajet Engine	16
SMARTech Petrol Engine	15
DDIS Diesel Engine	15
mHAWK D70 Diesel Engine	14
HR15 Petrol Engine	13
DI Turbo Diesel Engine	13
1.2 i-VTEC Petrol	13
K 10B Petrol Engine	13
Bifuel Engine	13
TDi Turbocharged Diesel W	13
V-Type Diesel Engine	13
MULTIJET Engine	12
K15 Smart Hybrid	12
G12B	12
Turbo Diesel Engine	12
mHawk 80 Diesel Engine	11
k15B VVT Engine	11
K4M Petrol Engine	11
In Line Diesel Engine	11
SD4 Diesel Engine	11
CR4 Diesel Engine	11
MDI 3200TC Engine	11
Revotorq, 1.5L Turbocharg	11
mHawk155 Diesel Engine	10
1.2 L Petrol Engine	10
DDIS 200 Smart Hybrid	10
K series Dual Jet	10
CRDi Diesel Engine	10
NEF TCI- CRDe Diesel Engi	10
D15 1.5 Liter Diesel Engi	10
1.1 litre Petrol Engine	10
VGT CRDi Diesel Engine	9
mFALCON G80 Engine	9
i-DTEC Engine	9
Kappa Dual VTG Petrol En	9
1.5 L Diesel Engine	9
V-Type Engine	9

1.5L i-VTEC DOHC with VTC Petrol Engine	9
TFSI Petrol Engine	8
sDrive20d Diesel Engine	8
D4 Diesel Engine	8
TwinPower Turbo 4 Cylinde	8
m2DiCR Diesel Engine	8
Revotron 1.2L Turbocharge	8
CDI In-line Engine	8
Front Wheel Drive	8
4-in-line	7
Kappa Engine	7
In-Line	7
1.5 litre Diesel Engine	7
1.5 Diesel Engine	7
In Line Engine	7
DOHC Dual VVT-i Petrol En	7
1.0L MPI Petrol	7
RX270 XVT Diesel Engine	6
i-VTEC	6
Ecoboost Petrol Engine	6
1.4-Litre 4-Cyl Multiair	6
TwinPower Turbo 4 Cylinder Engine	6
D-4D Diesel engine	6
V6 Diesel Engine	6
800 L SOHC 3 Cylinder	6
TCDi Diesel Engine	6
mDI CRDe	6
xDrive20d Diesel Engine	6
K12M Petrol Engine	6
VGT Diesel Engine	6
U2 Diesel Engine	6
Twin Turbo Diesel Engine	6
Smartstream G1.5	6
1.2L U2 CRDi Diesel	6
4 Cylinder, VCDi	5
TD4 Diesel Engine	5
K15 Smart Hybrid Petrol Engine	5
1.2L Kappa Dual VTET Petr	5
16V DOHC VTT DICOR	5
TDI Diesel engine	5
1.5L Turbo Diesel	5
Duratorq	5
TwinPower Turbo 4-cylinder engine	5
2 cylinder, MPFI	5
1.2 litre Petrol Engine	5
1.5L CRDi VGT	5
In-line Diesel Engine	5
Revotorq, 1.5L Turbocharged engine	5
4 Cylinder, Common Rail,	5
K15B Smart Hybrid	5
GDi Petrol Engine	5
1.2L Turbo Petrol	5
TwinPower Turbo 4-Cylinde	5
1.2 Litre Petrol Engine	5
K14B Petrol Engine	5
1.5 l U2 diesel	5

DDiS 320 Diesel Engine	4
dCi CRDi Diesel Engine	4
MULTIJET	4
DDIS Smart Hybrid Diesel Engine	4
DDIS Diesel Engin	4
1.0 l Turbo GDi petrol	4
mStallion 150 TGDi Engine	4
TDi Engine	4
Diesel S Engine	4
Naturally Aspirated 12V DOHC EFI	4
Dicor Diesel Engine	4
Isuzu Ddi VGS TurboEngine	4
Dual VVTi Petrol Engine	4
XSDE Engine	4
OM 654 Diesel engine	4
VTT VariCOR Diesel Engine	4
NEF TCI-CRDe Engine	4
2.2-litre Diesel Engine	4
CGI Petrol Engine	4
1.2 L Kappa Petrol	4
F8D Engine	4
1.1L, 4-Cyl, F10D Petrol	4
SZ CRDe, 4 stroke, Turbo	4
Boosterjet Petrol Engine	3
TDI Quattro Engine	3
TDI Engine	3
K15B ISG Petrol Engine	3
mDI CRDe Diesel Engine	3
Dual VTVT Petrol Engine	3
2AR-FXE Petrol Engine	3
TDi Diesel Engine	3
2.0-Litre 4-Cyl Multijet II Engine	3
CDI Blue Efficiency Engin	3
Smartstream G1.4 T-GDi	3
TSi Petrol Engine	3
1.5 L TDCi	3
Dicor Complaint With Dual	3
1.5L Turbocharged Intercooled	3
TCDI Diesel Engine	3
Turbocharged Petrol Engin	3
TFSI Quattro Engine	3
VCDi Diesel Engine	3
2.0 Liter, 4-Cylinder Tur	3
2-TR-FE Dual VVT-i Engine	3
2.0L Mid-engine	3
1.2L Revotron	3
mEagle Diesel Engine	3
1.4L GDi Turbo Petrol	3
TDI Quattro Diesel Engine	3
Petrol	3
U2 VGT CRDi Diesel Engine	3
T-Jet Petrol Engine	3
dCI Engine	3
4 Cylinder 8 valve	3
Naturally Aspirated 12V D	3
In Line Twin Turbo Engine	2

2.0L Turbocharged Diesel	2
2.2 LTR. VARICOR 400	2
VARICOR Diesel Engine	2
Blue Efficiency Engine	2
DI-Diesel Engine	2
1.0 l Kappa Turbo GDi Petrol	2
4.0L V8 biturbo	2
K9K In-Line Diesel Engine	2
G13BB MPFI Gasoline Engin	2
2NR-FE ,Gasoline, 4 Cylin	2
U2 CRDi Diesel	2
1.0L TSI Petrol	2
In-Line 4 Cylinder diesel Engine	2
Blue Efficiency Petrol En	2
Spark ignition, multipoin	2
m2DiCR Engine	2
i-VTEC Petrol engine	2
1.6-litre i-DTEC diesel engine	2
K Series Engine	2
VGS Turbo Diesel Engine	2
1.5L MPi Petrol	2
xDrive30d Diesel Engine	2
Quadrajet Diesel Engine	2
1.3-Litre DDiS Diesel Eng	2
Kappa 1.2 L MPi Petrol	2
V8 Diesel Engine	2
5.0L Ti-VCT V8	2
Kryotec 2.0 L Turbocharged	2
1.5L CRDi Diesel	2
3.0L TD6 Diesel Engine	2
All Aluminium light weigh	2
Naturally Aspirated 12V	2
475 SI Multi Point Fuel I	2
1.4-Litre 4-Cyl Multiair Engine	2
1.5L i-DTEC DOHC Diesel Engine	2
1.4-litre TDi Engine	2
Electric Engine	2
K9K dCi Diesel Engine	2
dCi In-Line Diesel Engine	2
16V DOHC Common Rail CR4	2
1.5L Revotorq Turbocharged	1
k14B Petrol Engine	1
1.0 L ENERGY	1
EcoBlue engine	1
STRIDE Diesel Engine	1
Gamma VTG Petrol Engine	1
FIRE Engine	1
4 Cylinder, Common Rail S	1
FMTECH Engine	1
1.2 Petrol Engine	1
SOHC D-4D Diesel Engine	1
TwinPower Turbo Inline 6 Cylinder Engine	1
Smartstream G1.0 T-GDi	1
1.1 L Bi-Fuel	1
M9R Diesel Engine	1
sDrive 18i	1

Revotron 1.2 l	1
1.2-litre Petrol Engine	1
1.5L i-DTEC	1
SMARTech Engine	1
2.0 Turbocharged petrol engine	1
TwinPower Turbo inline 4-cylinder engine	1
1.5 Litre mHAWK 75 BSVI Diesel Engine	1
Duramax Engine	1
Smartstream G1.2	1
Advanced Multijet Diesel	1
S-TEC II Engine	1
1.8-litre i-VTEC petrol engine	1
D15 1.5 Litre Diesel Engine	1
VTT VARICOR Diesel Engine	1
In Line	1
L4 200d	1
H4K Petrol Engine	1
2.0 litre 4-cylinder in-l	1
TwinPower Turbo Inline 4	1
1.3-Litre DDiS Diesel Engine	1
Common Rail Direct Inject	1
1.5L CRDi WGT	1
K15B Petrol Engine	1
Twin Turbo & Super Charge	1
1ND-TV,Diesel 4-Cylinder	1
Kryotec 2.0LTurbocharged	1
OM616 Turbo Diesel Engine	1
mHAWK Diesel Engine	1
V4 Twin Turbo Engine	1
Common Rail CR4	1
4Cylinder Turbo Intercool	1
CR4 Turbo Intercooled Eng	1
1.2L Revotron Turbocharged	1
1VD FTV Diesel Engine	1
DOHC i-VTEC	1
3.0L Ti6 Petrol Engine	1
1.5 L Turbocharged Intercooled	1
1.0 L Petrol Engine	1
2.2L VGT Diesel Engine	1
1.2 L U2 CRDi Diesel	1
VVT	1
DDiS 225 Diesel Engine	1
Revotron 1.2L Turbocharged engine	1
RX270 XDi Diesel Engine	1
1.5L Turbo Diesel Engine	1
Gasoline Hybrid and E-FOUR	1
HRA0 1.0 TURBO PETROL	1
m2DICR Diesel Engine	1
2.2 FMTech CRDI Engine	1
V6 Petrol Engine	1
KB Series	1
V6-cylinder diesel engin	1
4.8-litre V8 Engine	1
1.2 L Revotron	1
K10C	1
Si4 Petrol Engine	1

1.5 l U2 CRDi Diesel Engine 1

Name: Engine Type, dtype: int64

Turbo Charger

No 5034

Yes 3570

no 6

YES 5

Twin 3

yes 2

Name: Turbo Charger, dtype: int64

Super Charger

No 8497

Yes 12

NO 4

Name: Super Charger, dtype: int64

Car_Brand

Maruti Swift 956

Maruti Alto 525

Hyundai i20 463

Hyundai Grand 354

Maruti Wagon 295

Hyundai i10 280

Honda City 246

Maruti Baleno 218

Maruti Ciaz 199

Hyundai Verna 179

Volkswagen Polo 179

Renault KWID 172

Toyota Innova 156

Honda Amaze 154

Ford EcoSport 149

Maruti Vitara 141

Toyota Etios 131

Hyundai Creta 128

Mahindra XUV500 124

Hyundai EON 121

Maruti Celerio 118

Hyundai Santro 116

Mahindra Scorpio 109

Renault Duster 104

Maruti Ertiga 104

Ford Figo 101

Volkswagen Vento 94

Maruti Ritz 93

Tata Tiago 88

Honda Jazz 81

Honda Brio 81

Ford Ecosport 79

Mahindra Bolero 79

Maruti Ignis 78

Toyota Fortuner	77
Skoda Rapid	74
Hyundai Xcent	67
Maruti SX4	54
Hyundai Venue	52
Tata Indica	52
Maruti Eeco	49
Chevrolet Beat	47
Maruti Zen	46
Mercedes-Benz E-Class	41
Tata Indigo	40
Ford Fiesta	40
BMW 3	39
Volkswagen Ameo	37
Datsun RediGO	35
Nissan Terrano	34
Jeep Compass	32
Ford Endeavour	32
Mahindra TUV	31
Nissan Micra	31
Mahindra Thar	31
Tata Safari	31
Toyota Corolla	31
Maruti Omni	31
Honda Civic	30
Nissan Sunny	30
Tata Nexon	29
Audi A6	28
Land Rover	28
Mahindra KUV	28
Maruti 800	28
Ford Aspire	27
Honda WR-V	26
Mercedes-Benz C-Class	25
BMW X1	25
Mahindra Xylo	23
Tata Nano	22
Datsun GO	22
Chevrolet Spark	22
Audi Q3	21
Ford Ikon	21
Maruti S-Cross	21
BMW 5	20
Toyota Glanza	20
Hyundai Accent	20
Audi A4	20
Renault Triber	20
Mahindra Quanto	20
Maruti S-Presso	19
Mahindra XUV300	18
Honda Mobilio	17
MG Hector	17
Skoda Superb	17
Toyota Yaris	16
Tata Zest	15
Fiat Grande	15

Skoda Octavia	15
Kia Seltos	15
Chevrolet Tavera	14
Audi Q5	13
Hyundai Elantra	13
Volkswagen Jetta	13
Skoda Fabia	12
Fiat Linea	12
Skoda Laura	12
Ford Freestyle	12
Chevrolet Sail	12
Maruti A	12
Mercedes-Benz CLA	12
Tata Manza	12
Volvo S60	11
Hyundai Getz	11
Tata Tigor	11
Chevrolet Cruze	10
Mahindra Marazzo	10
Maruti Dzire	10
Mercedes-Benz GLA	10
Chevrolet Enjoy	10
Hyundai Santa	9
Mahindra Verito	9
BMW X3	9
Jaguar F-Pace	9
BMW X5	8
Tata Aria	8
Jaguar XF	8
Honda CR-V	8
Toyota Platinum	8
Mahindra KUV100	8
Porsche Cayenne	8
Audi Q7	7
Hyundai Sonata	7
Mercedes-Benz GL-Class	7
Mercedes-Benz M-Class	7
BMW X6	7
Mahindra Renault	7
Mercedes-Benz S-Class	7
Mini Cooper	7
Toyota Camry	6
Tata Bolt	6
Ford Classic	6
Honda BRV	6
Mahindra Ssangyong	6
Volvo S90	6
Mahindra NuvoSport	5
Mitsubishi Pajero	5
Porsche Panamera	5
Maruti Esteem	5
Volvo XC90	5
Jaguar XE	5
Tata Sumo	5
Toyota Qualis	5
Jaguar XJ	5

Isuzu MU-X	5
Maruti Gypsy	5
BMW 7	5
Audi Q2	4
Maruti XL6	4
Datsun redi-GO	4
Mahindra Logan	4
Renault Pulse	4
Kia Sonet	4
Fiat Avventura	4
Ford Mustang	3
Porsche 718	3
Volvo V90	3
Renault Scala	3
Mercedes-Benz GLC	3
Volvo S	3
Volvo V40	3
Chevrolet Captiva	3
Renault Lodgy	3
Audi A3	3
Tata Harrier	3
Volvo XC	3
Volvo XC60	2
Volkswagen Passat	2
Honda Accord	2
Chevrolet Aveo	2
Maruti Estilo	2
Mercedes-Benz CLS	2
Ford Fusion	2
Isuzu MU	2
Mercedes-Benz B-Class	2
Renault Fluence	2
Porsche 911	2
Tata Vista	2
Chevrolet Optra	2
BMW 6	2
Fiat 500	2
Mahindra Supro	2
Force One	2
Hyundai Aura	2
Ambassador Grand	2
Tata Hexa	2
Renault Kiger	1
Mercedes-Benz B	1
Volvo XC40	1
ICML Rhino	1
Porsche Macan	1
Renault Koleos	1
Kia Carnival	1
Force Gurkha	1
Mahindra Jeep	1
Ambassador Classic	1
Maruti A-Star	1
Honda New	1
New Skoda	1
Honda BR-V	1

Mahindra E	1
Volkswagen Tiguan	1
Mercedes-Benz GLE	1
Ambassador CLASSIC	1
Volkswagen Beetle	1
BMW Z4	1
Toyota Vellfire	1
BMW X4	1
Nissan Magnite	1
Fiat Punto	1
Mitsubishi Outlander	1
Tata Altroz	1
Maruti Versa	1
Mercedes-Benz A-Class	1
Nissan Evalia	1
Volkswagen CrossPolo	1
Skoda Yeti	1
Aston Martin	1
Chevrolet Trailblazer	1
Mitsubishi Lancer	1
Honda CR	1
Tata Sierra	1
Mercedes-Benz A	1

Name: Car_Brand, dtype: int64

Car_Model

Dzire	169
i10	166
K10	99
800	94
VDI	85
800 LXI	79
R	78
VXI	63
LXi	63
Magna	62
Dzire VDI	60
R VXI	57
Brezza	56
VDI BSIV	43
K10 2010-2014 VXI	40
Dzire VDi	40
LX	37
1.5 TDI Highline	35
VDi	32
i10 Sportz	31
Liva	30
Xing GLS	30
RXT	29
Era Plus	29
i10 CRDi Sportz	28
1.6 SX	27
Sportz	27
LXI	26
ZDI	26

K10 LXI	25
Diesel Titanium	25
Dzire VXI	25
i10 Magna	24
800 VXI	24
S MT	22
Liva GD	22
Sportz 1.4 CRDi	22
RXL	22
Asta 1.4 CRDi	20
Delta 1.2	20
S i-Vtech	20
LDi	20
R LXI Minor	20
D Lite Plus	20
Brezza ZDi	20
Sportz AT	20
R LXI	20
K10 VXI	19
Magna 1.2	19
S	19
2.0 TDI Premium Plus	19
3.2 Titanium AT 4X4	19
1.5 Diesel Titanium	19
1.5 TDI Comfortline	19
Diesel Highline	18
110PS Diesel RxZ	18
Sportz 1.2	17
Brezza VDi	17
VXI AT	17
Dzire LDI	16
1.3 VXi	16
W8 2WD	16
Delta 1.3	16
Era	16
Dzire ZDI	16
LS	15
Active	15
Delta	15
1.2 V i VTEC	15
1.0 RXT Optional	15
1.6 CRDi SX	15
S i-Dtech	15
AC	15
i10 1.2 CRDi Sportz	15
Series 320d Luxury Line	15
Diesel ZXI	15
1.5 DV5 MT Titanium	14
Asta 1.2	14
Brezza ZDi Plus	14
Magna 1.2 iTech SE	14
100	14
4x2 AT	14
Vdi BSIII	14
R LXI CNG	14
1.5 S MT	14

G	14
Altis	14
Diesel LS	14
85PS Diesel RxL	14
ZDi	13
LXi BSIII	13
Zeta 1.2	13
C8	13
1.2 Sportz	13
5 Seater AC BSIV	13
1.5 TDCi Titanium BSIV	12
V	12
Dzire Tour LXI	12
2011-2015 W8 FWD	12
i VTEC V	12
300	12
Diesel EXI	12
1.5 TDCi Titanium	12
1.5 DV5 MT Trend	12
3.0 Diesel	12
Asta	12
7 Seater Standard BSIV	12
i10 CRDi Magna	12
Alpha 1.2	12
E MPI STD BS IV	12
4x2 4 Speed AT	12
1.2 MPI Comfortline	12
i10 1.2 Kappa Sportz BSIV	11
Zxi BSIII	11
2015-2017 Sportz 1.2	11
Crysta 2016-2020 2.8 ZX AT BSIV	11
Diesel XV	11
i-DTEC VX	11
1.4 Sportz	11
ZDi SHVS	11
R AMT VXI	11
1.2 Asta	11
GD	11
4x2 Manual	11
1.8 V AT	11
Petrol Trendline 1.2L	11
1.2 Revotron XT	11
VXI Optional	11
sDrive 20d xLine	11
V MT	10
Diesel	10
GLS	10
GT TSI	10
2011-2019 SLX	10
1.6 CRDi AT SX Plus	10
Crysta 2016-2020 2.8 GX AT BSIV	10
Xing GL Plus	10
2.5 VX (Diesel) 8 Seater	10
1.2 Spotz	10
VXI BSIV	10
2.5 VX (Diesel) 7 Seater	10

1.1 CRDi S	10
Dzire Tour 2012-2018 LDI	9
LS (TDI) BS III	9
Diesel Comfortline 1.2L	9
1.5 Diesel Titanium BSIV	9
Petrol Highline 1.2L	9
W6	9
2017-2020 Zeta DDIS 200 SH	9
1.3 Flair	9
CRDi	9
Magna 1.4 CRDi	9
Dzire VXi	9
XL	9
1.5 V MT	9
Star Vxi	9
1.2 Magna	9
LK 1.8 TSI AT	9
VDI (ABS) BS IV	9
2005-2017 DICOR 2.2 EX 4x2	9
2016-2021 S i-VTEC	9
Sportz 1.1L	9
LT	9
1.6 CRDI	9
1.2 MPI Highline	9
2.5 G (Diesel) 7 Seater	9
2.0 TDI Quattro Premium Plus	9
110PS Diesel RxL	9
ZXI	8
i10 Asta Option	8
1.2 Magna Executive	8
T Option	8
1.5 V AT	8
Diesel LT	8
1.0 LS	8
R Stingray	8
i-VTEC V	8
Diesel Trendline 1.2L	8
2016-2021 Prestige 2.0 AWD	8
1.5 Diesel Titanium Plus BSIV	8
Estilo VXI BSIV	8
1.5 TDI Ambition BSIV	8
V CVT	8
4x4 MT	8
Diesel Comfortline	8
ZXi	8
S Cross DDIS 200 Zeta	8
1.2 Revotron XZ	8
W4	8
Brezza ZDi Plus Dual Tone	8
VX	8
1.4 CRDi Magna	7
ZXI AT	7
2011-2019 DI - AC BS III	7
i VTEC CVT VX	7
1.2 VX MT	7
2.5 V Diesel 8-seater	7

xDrive30d	7
1.5 Diesel Trend	7
GLE	7
XV	7
CRDi 1.6 SX Option	7
W8 4WD	7
VDI Optional	7
LDI	7
Magna Plus	7
E250 CDI Avantgrade	7
R LXI DUO BSIII	7
LDI Optional	7
SHVS VDI	7
VLX 2WD AIRBAG BSIV	7
K10 VXI AGS	7
W8 AWD	7
Sigma 1.2	7
i-VTEC VX	7
1.2 VX i VTEC	7
2011-2019 ZLX	7
4X4	6
2015-2019 1.5 TDI Highline	6
ZDi Plus SHVS	6
1.0 RXT	6
1.6 MPI Ambition Plus	6
35 TDI Premium	6
2005-2017 DICOR 2.2 EX 4x2 BS IV	6
i10 Asta	6
2017-2021 ZXI Optional AMT BSIV	6
LX (TDI) BS-III	6
Rexton RX7	6
Era 1.1 iTech SE	6
2.5 G (Diesel) 8 Seater	6
Alpha Diesel	6
XL 110 PS	6
2.0 TDI	6
Punto Active (Diesel)	6
Power Plus SLE	6
K10 VXI Airbag	6
1.5 TDI AT Style BSIV	6
1.2 Sigma BSIV	6
i10 1.2 Kappa Asta	6
Zeta CVT	6
R VXI AMT	6
1.2 Delta BSIV	6
Diesel Celebration Edition	6
85PS Diesel RxE Adventure	6
1.0 LT	6
Delta Automatic	6
Brezza LDi Option	6
Estilo LXI BSIII	6
Alpha 1.3	6
Xing GL	6
1.5 Petrol Titanium Plus AT BSIV	6
1.2 Revotron XM	6
W10 2WD	6

1.2 Revotron XZA	6
VXi	6
M2DI	6
1.0 RXL	6
Power Plus ZLX	6
5 STR With AC Plus HTR CNG	6
Estilo LXI BS IV	6
CRDi SX	6
2016-2021 S Petrol	6
2.5 V Diesel 7-seater	6
XV 110 PS	6
C 220 CDI Avantgarde	6
T	6
Alpha	6
Twist XT	6
1.8 S MT	5
Dzire ZXi	5
Xing	5
Active 1.2 S	5
2016-2021 D4 Inscription BSIV	5
LXI Optional-O	5
AT ZXi	5
1.4 Duratorq ZXI	5
85PS Diesel RxE	5
i10 1.2 CRDi Asta	5
85PS Diesel RxL Optional	5
1.6 CRDi SX Option	5
Magna 1.4 CRDi (Diesel)	5
VDi Plus SHVS	5
2015-2019 1.0 MPI Comfortline	5
1.2 V AT i VTEC	5
1.5 E MT	5
A	5
35 TDI Quattro Premium Plus	5
Xing GLS LPG	5
sDrive20d	5
1.4 TDCi DuraTorq	5
35 TDI	5
2016-2021 S Diesel	5
R VXI BS IV	5
2018-2020 45 TFSI Technology	5
1.2 S MT	5
2016-2021 2.8 4WD AT BSIV	5
XT	5
1.6 TDI Ambition	5
i10 1.2 Kappa Magna BSIV	5
Brezza ZDi AMT	5
T BSIV	5
85PS Diesel RxS	5
110PS Diesel RxL AMT	5
LTZ AT	5
2.0 Limited Option	5
Magna Optional 1.2	5
DI	5
2017-2020 EXi	5
S Cross	5

2017-2020 2WD	5
Asta Option BSIV	5
GT TDI	5
W6 2WD	5
1.2 Kappa AT SX Option	5
1.2 Revotron XZ Plus	5
8 Seater BSIV	5
Aspire	5
Std	5
Petrol Spotz	5
Diesel Highline 1.2L	5
Diesel XL	5
1.2 Era	5
DLS	5
xDrive 30d Design Pure Experience 5 Seater	5
Crysta 2016-2020 2.4 VX MT BSIV	5
1.0 Era Plus	5
200 CDI Style	5
2015-2019 GT TSI	5
1.5 TDI Highline AT	5
2.5 G4 Diesel 8-seater	5
Petrol Highline AT	5
i DTEC SV	5
i DTEC S	5
R W10 FWD	5
Exclusive E 220 d BSIV	5
Range Rover Evoque 2.2L Dynamic	5
Magna Plus BSIV	5
Series 320d Sport Line	5
1.0 T Option	5
2.0L TDI Highline AT	5
Brezza VDi Option	5
2017-2020 GXi	5
Dzire VDI Optional	5
1.5 Ti VCT AT Titanium	5
VX i-DTEC	5
SX Opt Turbo	5
i DTEC V	5
S11 BSIV	5
Petrol Comfortline 1.2L	5
SX CRDi AT	5
Dzire VXI 1.2	5
EX i-Dtech	5
800 CNG LXI	5
2.0 Limited	5
Crysta 2016-2020 2.4 VX MT	5
SLX	5
i DTec V	5
EX	5
Classic 1.4 Duratorq CLXI	5
R CNG LXI	5
1.0 RXT 02 Anniversary Edition	5
CRDi 1.6 SX	5
1.5 TDCi Trend BSIV	5
1.4 Asta	5
Dzire ZXI	5

1.5 TDI Trendline	5
DLS BSIII	5
Altis 1.8 VL AT	5
R VXI Minor	4
AMT VDI	4
SX i-DTEC	4
VTVT 1.6 SX	4
LX BSIII	4
RXT Optional	4
XV D Premium AMT	4
SHVS ZDI Plus	4
K10 VXI AGS Optional	4
Dzire LDi	4
V AT	4
1.2 Asta Dual Tone	4
VXI BSIII	4
Xing XL	4
LX (TDI) BS III	4
E250 CDI Avantgarde	4
Zeta	4
SLE BSIV	4
Storme VX	4
1.3 LS	4
1.3 VXI ABS	4
C6	4
1.2 Revotron XT Option	4
Crysta 2016-2020 2.4 ZX AT	4
Active XV	4
Dzire VXI 1.2 BS IV	4
Xing ABS	4
Logan 1.5 DLE Diesel	4
2.5 EV (Diesel) PS 7 Seater BS IV	4
2016-2021 Ace Edition CVT Petrol BSIV	4
1.2 Revotron XE	4
1.99 S6 Plus	4
1.2 Kappa SX	4
VXI 2018	4
300 T8	4
1.6 CRDi AT S	4
Aura (ABS) Quadrajet BS IV	4
SLE	4
S2	4
2.5 G (Diesel) 8 Seater BS IV	4
1.4 CRDi	4
Estilo VXI BSIII	4
2015-2017 Sportz 1.4 CRDi	4
2.0L TDI Comfortline	4
i VTEC E	4
1.5 TDI Ambition	4
2015-2017 Asta 1.2	4
1.1 CRDi SX	4
R VXI Optional	4
AC BSIII	4
R LXI Optional	4
XV CVT	4
VX CVT	4

Sharp DCT	4
Asta Option 1.2	4
Aura (ABS) Quadrajet	4
D5 Inscription BSIV	4
Active 1.4 SX	4
Class 200 CDI SPORT	4
1.6 SX Automatic Diesel	4
R Stingray VXI	4
R Duo Lxi	4
Diesel PS	4
Petrol Titanium	4
300 mHAWK100 T8	4
ZDI Plus	4
1.4 CRDi S	4
1.3 DX	4
1.4 CRDi Sportz	4
Zeta Diesel	4
Vxi BSIII	4
2.4 GDi AT	4
AT W10 AWD	4
1.6 SX CRDI (O) AT	4
1.4 CRDi Asta	4
DDiS VDI	4
VXi BSIV	4
i10 CRDi Asta	4
Series 320d Prestige	4
eLS	4
1.2 Kappa SX Option	4
eLX BS IV	4
Ldi BSIII	4
CNG 5 Seater AC	4
Punto 1.3 Dynamic (Diesel)	4
Fe 4WD AT	4
Magna 1.1L	4
ML 350 4Matic	4
2.6 Turbo 9 Str	4
CX	4
E250 CDI Blue Efficiency	4
Petrol Highline	4
Diesel Sportz	4
Sportz AMT BSIV	4
1.2 Zeta BSIV	4
ZXi BSIV	4
Dzire ZDi	4
i10 SportZ Edition	4
Diesel 1.5 DLE	4
1.6 SX Option	4
1.2 SV i VTEC	4
2016-2021 VX i-VTEC	4
1.6 SX CRDi (O)	4
R CNG LXI BSIV	4
Range Rover Sport HSE	4
ZXI Plus	4
1.5 DV5 MT Titanium Optional	4
Style 1.8 TSI AT	4
Cx BSIII	4

1.3 GLS	4
1.5 TDCi Ambiente BSIV	4
Xing XG	4
2017-2021 ZXI Optional	4
2005-2017 DICOR 2.2 VX 4x2 BS IV	4
1.1 CRDi SX Option	4
1.2 LS	4
1.8 S AT	4
Prestige	4
CRDi 1.4 EX	4
2.5L 4X2 MT	4
1.5 V MT Exclusive	4
VXI Plus	4
1.6 SX Automatic	4
2016-2021 VX CVT i-VTEC	4
Standard With Sunroof	4
Plus Sharp AT	4
Xi (Petrol)	4
Altis 1.8 G	4
D3	4
Sportz Option 1.4 CRDi	4
1.2 AMT Delta BSIV	4
V8	3
S Cross DDiS 320 Alpha	3
VD	3
Zxi with Leather BSIII	3
1.6 SX VTG	3
DLX	3
1.6 MPI Style BSIV	3
AC BSII	3
1.2 Ti-VCT Trend	3
2.0 Limited 4X4	3
Asta Option 1.4 CRDi	3
W410 4X4	3
ZXI AT Optional	3
C 220 CDI Sport Edition	3
XL CVT	3
1.6 EXi NXT	3
2016-2021 2.8 2WD AT BSIV	3
D Lite	3
E MT	3
R AMT VXI Plus	3
100 mFALCON G80 K4 Plus	3
V i DTEC	3
SHVS ZDI	3
Ambiente 1.9 TDI MT	3
E 200 CGI	3
1.5 TDI Highline Plus	3
SLE BSIII	3
S11 4WD BSIV	3
LPG STD BSIV	3
Series 520d Sedan	3
1.3 Alpha	3
ZXI Optional	3
1.5 TDI Style BSIV	3
100 mFALCON D75 K4	3

Liva 1.2 G	3
Era Executive	3
V i VTEC	3
LXI BSIV	3
Series 320d	3
1.2 MPI Comfortline Plus	3
Diesel LXI	3
2015-2019 1.5 TDI Highline Plus	3
New 2.0 TDI Multitronic	3
1.5 Diesel Ambiente BSIV	3
1.2 S AT i VTEC	3
RS 1.0 Petrol	3
1.6 VTG SX Plus	3
E200 CGI Blue Efficiency	3
200 CDI Sport	3
Magna 1.1	3
1.6 ZXi ABS	3
VXi Plus	3
CNG 5 Seater AC BSIV	3
D2 BS IV	3
Liva G	3
2.0 Limited Option 4X4	3
1.0 MPI Comfortline	3
1.4 Magna Executive	3
Brezza VDi AMT	3
xDrive 20d Luxury Line	3
Elegance 2.0 TDI CR AT	3
1.05 Revotorq XM	3
Dzire Vdi BSIV	3
5 Seater AC	3
Sportz Diesel	3
Series 520d Luxury Line	3
Elegance 2.0 TDI AT	3
Thunder Edition Diesel BSIV	3
2015-2019 1.0 MPI Trendline	3
1.2 CRDi S	3
S7 140 BSIV	3
1.4 Asta Option	3
2017-2020 Sigma DDIS 200 SH	3
2.5 GX (Diesel) 8 Seater	3
NXT D75 K2	3
i VTEC SV	3
A/T	3
S10 7 Seater	3
i10 1.2 Kappa Era	3
2017-2020 Alpha DDIS 200 SH	3
1.4 AT Alpha	3
2015-2017 Magna 1.2	3
1.6 Highline	3
1.5 TDCi Titanium Plus BSIV	3
LDI SP Limited Edition	3
1.6 TDI Active	3
300 T6 Plus	3
S Diesel	3
ZXI MT BSIV	3
VX Diesel	3

Dzire AMT VXI	3
i-VTEC CVT ZX	3
Dzire Ldi BSIV	3
1.6 MPI AT Ambition	3
1.6 SX Diesel	3
FS F3	3
ZXi AT	3
Brezza LDi	3
2015-2017 Asta Option 1.2	3
R LX BSIII	3
1.2P Titanium MT	3
Emotion	3
1.2 Ti-VCT Titanium	3
R LX Minor	3
Series 525d Sedan	3
1.5 CRDi GVS	3
1.6 VTG SX	3
W8 AMT Optional Diesel	3
35 TFSi Premium	3
2006-2009 VLX 2WD 7 Str BSIII	3
Revotron 1.2T XM	3
Active 1.2 SX	3
Pure 4x2	3
R AMT VXI Option	3
1.6 CRDi EX MT	3
2.5 GX (Diesel) 7 Seater BS IV	3
R VXI 1.2	3
2009-2014 SLE 7S BSIV	3
VX i-Vtech	3
1.2 S i VTEC	3
Titanium	3
Estilo 1.1 VXI BSIII	3
1.5 Ti VCT MT Titanium	3
R LXI BSIII	3
AMT ZXi	3
Series GT Luxury Line	3
1.4 Zeta	3
Vxi BSIV	3
RXZ	3
E4 BS IV	3
D Lite Optional	3
Maybach S560	3
2.5 VX 7 STR	3
Class 200 d Sport	3
Cross 1.2L G	3
1.0 RXT AMT Opt BSIV	3
2.6 CRDe	3
Xing XO	3
Crysta 2016-2020 2.4 ZX MT	3
1.4 EX	3
i10 1.2 Kappa Sportz AT	3
220d 4MATIC	3
i10 Nios Sportz	3
Petrol RxE	3
1.5 Revotorq XM	3
1.6 SX VTG AT	3

2.5 E Diesel MS 7-seater	3
HTX IVT G	3
i VTEC VX	3
Dzire AMT ZDI	3
1.4 Duratorq CLXI	3
Punto 1.3 Emotion Pack 90HP (Diesel)	3
LDI BSIV	3
i10 AT Sportz	3
Diesel Trendline	3
xDrive 20d xLine	3
S11	3
Sigma	3
Boxster	3
2017-2020 Delta DDIS 200 SH	3
1.4 E Plus	3
2006-2020 35 TDI Quattro Premium Plus	3
1.6 Trendline	3
1.2 Kappa S	3
Cross 1.4L VD	3
1.4 Alpha	3
1.1 CRDi Base	3
1.0	3
1.6 MPI Active	3
3.0 TDI Quattro Premium Plus	3
Diesel XV Premium	3
GTX Plus	3
Magna LPG	3
VX MT	3
Cross Country D5 Inscription BSIV	3
1.4 Emotion	3
Exclusive E 220d	3
1.4 CRDi Base	3
XV Special Edition	3
1.99 S10	3
35 TDI Premium Plus	3
Crysta	3
Lei	3
XV Premium 110 PS	3
RxL Optional	3
CRDe	3
i10 AT Asta	3
Xing XP	3
xDrive20d	3
220 CDI AT	3
2.5 VX (Diesel) 8 Seater BS IV	3
R LXI BS IV	3
LTZ	3
1.6 TDI Ambition Plus	3
Monte Carlo 1.5 TDI AT BSIV	2
90 D5 Inscription	2
1.5 Diesel Ambiente	2
LX 4-Str Hard Top AT	2
Transform VTWT	2
1.0 S	2
Asta w/Sun Roof	2
DI 4X2 PS	2

Signature Edition Diesel BSIV	2
RS VDI	2
Executive D4	2
1.2L Diesel Classic	2
Discovery 2017-2021 HSE 3.0 TD6	2
1.5D Trend MT	2
Quadrajet 1.3 75PS XM	2
W10 AWD	2
1.4 Sigma	2
1.5 Ti VCT MT Titanium BSIV	2
1.2 CRDi E	2
C 200 CGI Elegance	2
1.2 MPI Ambition	2
1.5 Ti VCT AT Titanium BE BSIV	2
Ambition 2.0 TDI AT	2
1.2 TDI Active	2
Neo 3 LS 10 Seats BSIII	2
Alpha BSIV	2
Climber 1.0 MT	2
1.8 TFSI Premium Plus	2
Dzire LXI 1.2	2
W11 Option BSIV	2
i DTec VX Option	2
S10 AT 2WD	2
1.2 Ti-VCT Ambiente	2
SLX 4WD BSIII	2
D1	2
Series 320d GT Luxury Line	2
Sportz Option	2
M8 8Str	2
GLS 1.6 ABS	2
Petrol Asta Dual Tone	2
AMT ZXI BSIV	2
RXT BSIV	2
1.8 (E) MT	2
Titanium Diesel BSIV	2
Quadrajet XM	2
E4	2
2016-2021 i-VTEC Privilege Edition	2
M Sport sDrive 20d	2
Liva 1.4 GD	2
XXi (Petrol)	2
1.2 VTG S AT	2
Plus 1.4 TDCi Diesel	2
Pride 4x4	2
1.8 TSI Ambiente	2
1.2 Revotron XMA	2
800 Base	2
Era 1.1	2
LPG Era Plus	2
8 Seater BSII	2
Titanium Petrol BSIV	2
ZXI AMT	2
1.2 VX AT	2
RS Option i DTEC	2
LX 4-Str Hard Top Diesel AT	2

Etios 1.5 G	2
LX 4-Str Hard Top	2
Transform SX VGT CRDi	2
1.6 SX Option Diesel	2
Captiva XTREME	2
CRDe AC	2
AMT	2
S i DTEC	2
30 TFSI Premium FWD	2
2016-2021 2.7 2WD MT BSIV	2
AMT DDIs VDI	2
M4 8Str	2
Fe 4X4	2
DI Non AC BSIII	2
Progressive C 220d	2
Revotron XMS	2
4x4 XLT	2
C 250 CDI Elegance	2
XZ BSIV	2
1.6 E Plus	2
Lounge	2
2.4 AT	2
1.05 Revotorq XZ	2
S Option MT	2
1.6 CRDI AT SX Option	2
2015-2017 30 TDI	2
S Cross DDIs 200 Sigma	2
Sport 4X2 AT	2
2017-2020 CVT	2
1.1 CRDi S Option	2
CRDi SX ABS	2
2019-2020 XZ	2
LXi BSII	2
2016-2021 2.8 4WD MT BSIV	2
Fe 4x4 AT	2
Liva VD	2
1.6 SX VTVT (O) AT	2
E i-Vtech	2
1.3 LXI	2
1.4 CRDi S Plus	2
XE 85 PS	2
1.2 MPI Trendline	2
Dzire ZDI Plus	2
1.5 SV i DTEC	2
1.5 Diesel Trend Plus BSIV	2
i DTec E	2
1.6 ZXi Leather	2
W6 Diesel BSIV	2
43 AMG Coupe	2
Estilo	2
1.0 Ecoboost Trend Plus BE BSIV	2
M2 BSIV	2
1.2 MPI Classic	2
1.6 MPI Ambition BSIV	2
S 350 CDI	2
1.6 Duratec EXI	2

RXZ AWD	2
2016-2021 V CVT Diesel BSIV	2
i DTEC VX	2
3 DOOR S	2
R LX	2
B200 CDI	2
Series 530d M Sport	2
T Jet Dynamic	2
R Pro	2
Neo 3 10 Seats BSIII	2
AT ZXi Plus	2
Plus A	2
2009-2014 VLX 2WD 7S BSIV	2
300 T10	2
HTX Plus D	2
C4	2
X ZXi Option BSIV	2
1.4 Duratec EXI	2
1.3 Delta	2
i-VTEC V CVT	2
RXL BSIV	2
Dzire LXi	2
Plus T BSIV	2
1.5 TDCi Trend	2
1.4	2
Active XV Petrol	2
Titanium BSIV	2
Brezza ZDi Plus AMT	2
V MT Diesel	2
R LXI LPG BSIV	2
VLX SE BSIV	2
Dzire ZXi Plus	2
2005-2017 EX 4x2	2
2017-2021 ZXi MT BSIV	2
Titanium Diesel	2
D2	2
2.0 V	2
Aqua Quadrajet	2
Vista Aqua 1.3 Quadrajet	2
1.5 Petrol Titanium Plus AT	2
W11 Option AT	2
1.0 Ecoboost Titanium	2
S Diesel BSIV	2
KRAZ Plus AMT Diesel	2
2.0L Portfolio	2
N6	2
2011-2019 DI NON AC BS III White	2
1.6 TDI Active Plus	2
1.4 Limited Option Black	2
1.3 TCDi LS 8	2
Discovery Sport 2015-2020 SD4 HSE Luxury 7S	2
Brezza ZXi	2
1.5 D4 BSIII	2
Power Plus SLX	2
Pleasure 4x2	2
RXL AWD	2

1.8 V MT	2
Quadrajet 1.3 75PS XE	2
AT W6 1.99 mHawk	2
LXI Option	2
2016-2019 2.0L Diesel Portfolio	2
Hatchback 1.2 LS	2
LPG STD BSIII 5-STR W IMMOBILISER	2
2015-2017 Asta Option 1.4 CRDi	2
1.4 Duratorq Titanium	2
1.2P Trend MT	2
85PS Diesel RxL Plus	2
Ultima 1.6 MPI AT Elegance	2
2.0 Longitude BSIV	2
2.0 Longitude	2
2006-2020 45 TDI Quattro Technology	2
Punto EVO Power UP 1.3 Emotion	2
2018-2020 40 TDI Technology	2
Quadrajet 1.3 75PS XMS	2
Liva GD Xclusive Edition	2
RXE BSIV	2
Vista Aqua 1.3 Quadrajet ABS BSIV	2
2.5 VX (Diesel) 7 Seater BS IV	2
4x4	2
EXi 1.4 TDCi Ltd	2
R LX DUO BSIII	2
Magna Plus Option	2
R VXI BSIII	2
TCDi LS 8 Seater	2
Series GT 320d Luxury Line	2
2.5 Hybrid	2
1.9 TDI AT Ambiente	2
GLS BS IV	2
VLX 2WD AIRBAG AT BSIV	2
350 CDI Blue Efficiency	2
Titanium 1.5 TDCi	2
2.6 Turbo 7 Str	2
VLX 2WD BSIV	2
80 D5	2
2.0 Diesel Prestige	2
1.0 PS LPG	2
Crysta 2016-2020 2.4 G MT 8 STR	2
2016-2021 V Petrol BSIV	2
Xing XG AT eRLX Euro II	2
AMT Quadrajet 1.3 XMA	2
Elegance 220 CDI	2
Plus T	2
Magna AT	2
2015-2017 Asta 1.4 CRDi	2
1.5 EXI	2
Series 525d Luxury Line	2
MPI STD BSIV	2
2011-2019 Plus AC	2
Diesel Platinum Edition	2
Zeta Automatic	2
Power Plus Plus AC BSIV PS	2
2.5 G1 Diesel 8-seater	2

Vista Terra 1.4 TDI	2
Range Rover Evoque 2.0 TD4 HSE Dynamic	2
Discovery Sport TD4 HSE 7S	2
Countryman Cooper D	2
Diesel Comfortline AT	2
Cross 1.4L GD	2
G Smart Hybrid	2
3.0L Portfolio	2
Era Plus Option	2
VVT LXI	2
E	2
AMT ZDI	2
W9 1.99	2
800 LX Optional	2
Asta Sunroof AT	2
V Option i VTEC	2
Crysta 2016-2020 2.4 VX MT 8S BSIV	2
2017-2021 ZXI Optional MT BSIV	2
K10 LXI CNG Optional	2
1.5 Ti VCT MT Ambiente	2
Petrol RxL	2
H2	2
Series 520d	2
1.6 VTG	2
2.6 DX	2
300 T8 AMT	2
i-DTEC V MT	2
2016-2021 2.7 2WD AT BSIV	2
Progressive 220d	2
Quadrajet 1.3 XM	2
1.05 Revotorq XE	2
ML 250 CDI	2
Series 320d Sedan	2
1.2 E i VTEC	2
Vista Aura Safire Anniversary Edition	2
Carrera S	2
i10 1.2 Kappa Sportz Dual Tone	2
Dzire 1.2 Vxi BSIV	2
LX 2.6 Turbo	2
250 CDI	2
Etios 1.4 VD	2
i10 1.2 CRDi Sportz Option	2
Titanium Plus Petrol BSIV	2
TCDi LTZ 7 Seater	2
Xing GLS CNG	2
Altis D-4D GL	2
2011-2019 SLE BSIII	2
35 TDI Technology	2
Diesel E4	2
Vista Aqua 1.2 Safire	2
i-VTEC V MT	2
1.4 CX	2
1.2 VTG S	2
ZXI BSIV	2
LS TDI BSIII	2
3.0L 4X4 AT	2

Cross Country D3	2
R ZXI 1.2	2
1.6 SX VTVT (O)	2
N4	2
2.5 G (Diesel) 7 Seater BS IV	2
Active SX Dual Tone Petrol	2
Alpha CVT	2
Classic 1.4 Duratorq LXI	2
Hybrid	2
1.2 AMT Zeta BSIV	2
Series 520d Sport Line	2
2016-2021 S Option CVT i-VTEC	2
Xing GL PLUS CNG	2
1.4 S	2
2.5 VX 8 STR	2
Executive CNG	2
1.4 VTVT	2
1.4 Limited Plus BSIV	2
7 4x2	2
GT 1.5 TDI	2
SX AT	2
2.0 TDI 177 Bhp Premium Plus	2
D5 Summum	2
EX MT	2
GD SP	2
DL BSIII	2
W8 1.99 mHawk	2
2016-2021 S CVT i-VTEC	2
VXI AT Optional	2
Green LXi (CNG)	2
Crysta 2016-2020 2.4 GX MT	2
Delta Diesel	2
110PS Diesel RxZ Pack	2
DLE	2
2.6 SLX CRDe	2
1.5D Titanium MT	2
A Petrol	2
Diesel RxL	2
2011-2019 SLE	2
RS ZDi Plus SHVS	2
TDI	2
Series 320d Sport	2
Green VXI	2
1.5 Revotorq XT	2
1.4 EX Diesel	2
GLS BSIII	2
Elegance 1.9 TDI	2
NRG 2018-2020 Petrol	2
1.5 TDCi Ambiente	2
Star Lxi	2
Prime C 220d	2
300 T4 Plus	2
Storme EX	2
Emotion Pack (Diesel)	2
Urban Sport 200	2
xDrive 30d	2

C300 Cabriolet	2
2.5 GX 8 STR BSIV	2
DI 4X4	2
2016-2021 S Petrol BSIV	2
1.2 Kappa Base	2
SX	2
Neo 3 LS 10 Str	2
1.2 Revotron XZ Option	2
Estilo LX BSIV	2
King HT BSIV	2
1.5 S Elegance	2
1.4 ZXi Duratec	1
Logan 1.4 GL Petrol	1
1.6 SX Option AT	1
ZDi Plus	1
D4 BSIV	1
Petrol RXS CVT	1
W9	1
Zeta 1.3	1
2017-2021 ZXi AMT BSIV	1
Vibe 1.5 dCi D4	1
2016-2021 VX Petrol BSIV	1
1.6 EX VTVT	1
3.0L AT 4x2	1
Class 200 CDI	1
200 CGI Sport	1
Asta 1.4 CRDi (Diesel)	1
35 TDI Quattro Premium	1
Cross Country D4 AWD BSIV	1
i VTEC S	1
D5 Momentum BSIV	1
Range Rover Sport SE	1
Range Rover Evoque 2.2L Pure	1
ZDI BSIV	1
1.3 CLXi	1
2005-2017 DICOR 2.2 EX 4x4	1
1.2 Active	1
100 mFALCON G80 K2 Plus	1
1.2 TDI Ambition Plus	1
Series 525d	1
2.2 Litre Luxury	1
R LXI DUO	1
E8 ABS Airbag BSIV	1
LPG	1
Active 1.2	1
Diesel LT Option	1
2.4	1
Inscription D5 BSIV	1
RVi MT	1
sDrive 20i	1
1.4 TDCi EXI	1
AT W10 FWD	1
800 LXI BSIV	1
1500 DSZ BSIII	1
NXT D75 K6 Plus	1
2.0 Longitude Option BSIV	1

Sportz Plus CVT BSIV	1
2016-2021 E Petrol BSIV	1
2005-2017 Dicor EX 4X2	1
1.6 TDI Elegance	1
HTK Plus G	1
Vxi - BSIII	1
1.6 EX Petrol	1
i10 Magna CNG	1
W10 1.99 mHawk	1
1.6 SXi ABS	1
Logan 1.5 DLX Diesel	1
2.4 4WD AT	1
2.4L AT	1
SX Diesel AT	1
Expression E 220d	1
Estilo LX BSIII	1
W5	1
M8	1
2.5 VX 7 STR BSIV	1
VX BSIV	1
Limousine A 200d	1
Crysta 2016-2020 2.4 G MT	1
eXeta eGLX	1
H1	1
Stepway 110PS RXZ 7S	1
BSIV	1
3 DOOR D	1
Diesel XV Primo	1
LS BSII	1
Cross 1.2 G X Edition	1
Series 730Ld Sedan	1
S2 9 Seater	1
Xclusive Edition	1
S3 9 Seater BSIV	1
GS	1
XM	1
2016-2021 2.8 4WD AT	1
Classic 1.4 SXI Duratorq	1
i-DTEC V	1
Discovery Sport 2015-2020 TD4 HSE Luxury	1
Etios 1.5 V	1
MPI CARGO BSIV	1
4X4 Diesel AT	1
2016-2021 Prestige 2.0 Petrol	1
Class B180 CDI	1
Plus Sharp Diesel MT	1
Power Up 1.3 Emotion	1
Range Rover Evoque 2.0L Dynamic	1
2.5si	1
Asta Opt Turbo DCT DT	1
S4 9 Seater	1
1.4 Durasport EXI	1
2.8 GLX CRZ	1
1.5 D6 BSIII	1
2016-2021 D4 Momentum BSIV	1
eGLS	1

Series 730Ld Design Pure Excellence	1
Super Diesel MT	1
1.6 VGT CRDi	1
1.99 S10 4WD	1
1.5 Revotorq XZ Plus	1
1.6 ZXi Duratec	1
4x2	1
1.2 Kappa S Option	1
Lxi	1
Series GT 630i Luxury Line 2018-2021	1
300d	1
5 Seater STD	1
1.9 Highline TDI	1
Muzic 1.0 LS	1
DLX BSII	1
B2-7 seats BSIII	1
W7	1
MT With Sun Roof	1
2015-2017 Sportz Option 1.4 CRDi	1
230 Avantgarde	1
Plus T Option	1
1.5 TDI Ambition With Alloy Wheel	1
1.5 GLXi	1
SX IVT	1
1.3 TCDi LT 7	1
Titanium Plus 4X2 AT	1
2016-2021 S Diesel BSIV	1
Diesel RxZ	1
R LXI CNG Optional	1
DLE BSII	1
2.4L 4WD AT	1
RXZ BSIV	1
GS zipPlus	1
M4	1
Estilo 1.1 LXI BSIII	1
GS C1	1
D4 KINETIC	1
Plus Anniversary Edition	1
1.4 G4 BSIV	1
1.5D Titanium Plus MT	1
XE	1
2017-2021 LXI Optional MT BSIV	1
S400d 4Matic	1
VXi Option	1
VLX CRDe	1
MKII EX BS IV	1
220 CDI	1
1.5 TDI	1
i10 1.2 CRDi Era	1
1.6 EXI josh 100	1
Discovery Sport R-Dynamic SE Diesel	1
EX i-Vtech	1
ZXi Option	1
Crysta 2016-2020 2.4 VX MT 8 STR	1
VTVT 1.4 EX	1
Delta Automatic BSIV	1

Magnum 2.0 LT	1
Executive LPG	1
E250 CDI Elegance	1
Freelander 2 TD4 SE	1
Dzire LXI Option	1
ZXi Plus	1
J CVT BSIV	1
Diesel 4WD	1
D4 BS III	1
1.5 Revotorq XZA Plus	1
Dzire ZXI 1.2 BS IV	1
800 VXI BSIV	1
2017-2021 ZXI	1
W8 Diesel BSIV	1
CRDi SX AT	1
i-VTEC ZX	1
Hard Top BS3 4WD	1
1.5 TDCi Titanium Plus BE BSIV	1
Dzire AMT VXI BS IV	1
SX ABS 7 Seating	1
K10 VXI Optional	1
Turbo	1
NXT G80 K6 Plus	1
Rider 1.9 TDI MT	1
BSIV ZXI Plus	1
1.2 TSI Highline BSIV	1
1.4 Asta (AT)	1
2005-2017 DICOR 2.2 VX 4x2	1
1.6 VTVT EX AT	1
B6	1
i-DTEC S	1
2011-2019 DI BSIII	1
2017-2021 VXI MT BSIV	1
Rapide V12	1
XL Plus 85 PS	1
1.6 SXi	1
Diesel Breeze	1
D3 R-Design	1
2016-2021 2.8 2WD AT	1
2009-2014 EX 2WD 9S	1
1.2 HTK Plus	1
RxZ	1
220d 4MATIC Sport	1
Crysta 2016-2020 2.4 GX AT	1
2.2 Trend MT 4X4	1
Crysta 2016-2020 2.4 GX MT BSIV	1
Storme LX	1
sDrive20d xLine	1
RXZ EASY-R AMT	1
1.6 Highline Plus 16 Alloy	1
1.3 Sigma	1
H8 Airbag	1
S10 8 Seater	1
D4 R-Design	1
VDi Plus	1
1.0 Neotech	1

Ambiente 1.9 TDI	1
1.0 TSI Highline	1
Liva V	1
S Turbo BSIV	1
Quadrajet XE	1
DLX BSIII	1
W11 BSIV	1
Base 9 Str BS IV	1
Cx BSIV	1
D2 BSIV	1
Series 750Li	1
GS zipDrive - Euro I	1
IPL II Petrol Highline	1
1.5 Petrol Titanium AT	1
D5 Inscription	1
1.6 MPI Style Plus	1
2009-2014 LX 2WD 7S	1
Altis G	1
Sportz Plus BSIV	1
XTA	1
i DTec SV	1
1.6 VTG SX Option	1
2015-2020 D4 KINETIC	1
Revotron 1.2T XMS	1
1.0 Kappa Magna Plus	1
AT	1
1.2P Titanium Plus MT	1
CRDi ABS	1
1.8 TSI	1
Aqua Quadrajet BS IV	1
Genus VXi	1
1.6 E Plus Diesel	1
1.5D Ambiente MT	1
1.2 TSI Highline AT	1
Asta (o)	1
K10 Knighttracer	1
2.2 Titanium AT 4X2	1
3.0 Litre S Premium Luxury	1
i-VTEC S	1
GTX Option	1
1.5 TDI Elegance Plus Black Package	1
Dzire Tour S	1
1.0 Ecoboost Titanium Plus BSIV	1
2011-2019 ZLX BSIII	1
800 VXI Optional	1
1.8 TSI Active	1
C2	1
SX AT Diesel	1
DI DX 8 Seater	1
Vista Aura 1.3 Quadrajet (ABS) BS IV	1
R VXI BSII	1
Dzire LXI	1
E250 Edition E	1
1.2 CRDi SX	1
Celebration Petrol	1
eVX	1

Etios 1.4 VXD	1
VLX SE BSIII	1
VLX 2WD BSIII	1
Crysta 2016-2020 2.8 AT Touring Sport BSIV	1
1.5	1
AMG 45	1
Ultima 1.6 TDI Elegance	1
Altis G AT	1
S Cross DDiS 320 Delta	1
1.5 HTX Plus Diesel	1
SX Option	1
V 2.4L 4WD AT AVN	1
Class 200 d Style	1
W8 Option Diesel	1
2006-2020 40 TFSI Quattro Technology	1
Zeta AT	1
VX AT i-Vtech	1
SX Plus Turbo DCT BSIV	1
1.5 Diesel Titanium Plus	1
B6 Opt	1
1.5 VDI	1
Expression E 220	1
Titanium Plus Diesel BSIV	1
SX Diesel BSIV	1
VLX 2WD AIRBAG BSIII	1
Crysta 2016-2020 2.4 GX MT 8 STR	1
280 CDI	1
100 mFALCON G80 K6 Plus	1
XZ Plus DualTone Roof	1
LEI	1
1.5 TDI AT Style Plus	1
SX iMT	1
N8	1
Magna Executive CNG	1
SLX 2WD BSIII	1
1500 DSL AC	1
Magna AMT BSIV	1
Dzire Tour 1.2 S STD	1
Diesel 250hp	1
i10 Nios Sportz CNG	1
100 mFALCON G80 K4	1
1.5 Ti-VCT Titanium	1
GT TSI BSIV	1
3.0 TDI Quattro Technology	1
1.4 TSI	1
SX Turbo	1
2016-2021 VX CVT Petrol BSIV	1
FS B1	1
1.5 TDI Highline Plus 16 AT	1
1.0 Neotech AMT	1
1.5 TDI AT Elegance Plus	1
1.2 Petrol	1
eLX	1
Sidewalk Edition	1
Club Class Quadrajet90 LS	1
2.5 E Diesel MS 8-seater	1

xDrive 30d M Sport	1
Turbo XV Premium	1
W9 BSIV	1
DX2 8-SEATER BSIII TWIN A/C	1
Ambiente 2.0 TDI CR AT	1
2.5 TDi AT	1
R Stingray LXI	1
3.0L Premium Luxury LWB	1
Neo 3 LS 10 Seats BSIV	1
1.4 Delta	1
E 8 Str STD	1
2.2 Litre Executive	1
i10 1.2 Kappa Sportz Option	1
Dzire VXi AT	1
1.5 GTX Plus Diesel	1
2016-2021 E i-DTEC	1
1.3 EXI	1
TSI Ambition AT	1
1.6 CRDi SX Plus Dual Tone	1
Titanium Plus	1
2.5 GX (Diesel) 7 Seater	1
Xing GL LPG	1
Dzire AMT ZDI Plus	1
SX Diesel	1
2L	1
100 mFALCON D75 K2	1
Countryman One	1
XMA	1
1.8 ZXi	1
VTVT 1.6 AT SX Option	1
S3 Plus 9 Seater	1
C 220CDIBE Avantgarde Command	1
Vista Aura 1.3 Quadrajet	1
1.5 V AT Exclusive	1
E270 CDI	1
Pride	1
S Petrol BSIV	1
VDI BSIV W ABS	1
2011-2019 SLX 2WD BSIII	1
TCDi LS 7 Seater	1
ZXI MT BSIV Leather	1
R LXI BSII	1
VTi-L (AT)	1
W6 1.99 mHawk	1
1.5 Petrol Titanium Plus	1
E-Hybrid	1
Petrol Asta	1
Neo LS B3 - 7 seats BSIII	1
Series 530d Highline	1
1.2 Alpha BSIV	1
1.5 TDCi Sports Edition	1
1.5 TDCi Titanium Plus	1
Executive Lounge	1
2015-2017 Magna 1.4 CRDi	1
2.0 Limited Plus	1
VVT ZXI Plus	1

SX i-VTEC	1
AE	1
1.0 LS LPG	1
Magnum 1.6 LT Petrol	1
S Cross DDiS 200 Delta	1
Series 730Ld	1
BSIV ZXI	1
Xing GLS Audio LPG	1
M Sport X xDrive20d	1
Accord 2011-2014 2.4 M/T	1
Plus T Option Petrol	1
1.0 Magna Plus Option O	1
1.2 MPI Highline Plus	1
H8 ABS with Airbags	1
Plus Select Diesel MT 7 STR	1
AT W10 1.99 mHawk	1
Verito D2	1
i10 1.2 Kappa Magna AT	1
SV MT	1
2.2 Titanium AT 4X2 Sunroof	1
AT With Sun Roof	1
Executive	1
Diesel Asta Dual Tone	1
GT 1.0 TSI	1
NXT D75 K8	1
2000 DSZ	1
Vista Quadrajet LS	1
Urban Cross 1.3 Multijet Active	1
Dzire AMT ZXI Plus	1
300 mHAWK100 T8 Dual Tone	1
Class A180 CDI	1
VDI Limited Edition	1
V MT Exclusive	1
Series 320i	1
1.3 Multijet Dynamic	1
1.5 V i DTEC	1
Petrol LS 8 Seater	1
Monte Carlo 1.6 MPI BSIV	1
1.2 Anniversary Edition	1
Prime T Plus CNG BSIV	1
100 mFALCON G80 K8	1
1.4 MPI Elegance	1
2016-2021 VX CVT Diesel	1
Etios 1.5 VX	1
Vista TDI LX	1
D4 SUMMUM	1
100 mFALCON D75 K8 AW	1
2.0 Diesel Pure	1
Asta AT	1
2009-2014 VLX 2WD AT 7S	1
2009-2014 EX 2WD 7S	1
C 250 CDI Avantgarde	1
Active S Petrol	1
1.2 Ti-VCT Titanium Plus	1
85PS RxL	1
Vista Terra Quadrajet 1.3L	1

1.9 TDI AT Elegance	1
2017-2021 VXI AMT BSIV	1
Dzire AMT ZXI Plus BS IV	1
VVT VXI	1
Asta Option Diesel	1
LX Non AC BSIII	1
80 D4 KINETIC	1
1.0 RXT AMT	1
2.0 Diesel	1
W8 AMT Optional Diesel BSIV	1
ZXI AT Leather	1
T-Jet Dynamic	1
Plus T Petrol	1
J CVT	1
MT Leather	1
VX 8 Str	1
1.5 Revotorq XMA	1
MM 540 DP	1
D2 BS III	1
S 350 d	1
Etios 1.4 GD	1
2018-2020 35TDI Technology	1
sDrive20d Expedition	1
H4	1
1.05 Revotorq XZ W0 Alloy	1
800 LXI Anniversary Edition	1
2009-2014 SLE 7S BSIII	1
Series 530d Sedan 3.0	1
NXT G80 K8 Dual Tone	1
110PS Diesel RxZ AWD	1
Zeta Automatic BSIV	1
2.0L TDI Highline	1
AMT Quadrajet 1.3 XTA	1
RXL 02 Anniversary Edition	1
Diesel Asta Option	1
Crysta 2016-2020 2.4 ZX MT BSIV	1
Crysta 2016-2020 2.4 G Plus MT 8 STR	1
2.0 Sport Plus	1
Dzire ZXI 1.2	1
H4 ABS	1
S4 Plus	1
J Optional CVT BSIV	1
Alpha Automatic BSIV	1
AMT ZXI Plus	1
LE (TDI) BS-III	1
D3 Summum	1
Series GT 630i M Sport	1
1.9 TDI MT Ambiente	1
eLS BS IV	1
i10 Asta Option AT	1
Xing XE	1
2.0 TDI Highline	1
GTX Plus AT D	1
1.4 Limited Option	1
110PS Diesel RXZ Optional with Nav	1
TDI LS	1

MULTIJET Emotion	1
DLE BSIII	1
2016-2021 SX i DTEC	1
1.6 CRDi AT S Plus	1
EX BS IV	1
Sharp AT BSIV	1
2011-2019 mHAWK D70 SLE	1
L and K 1.9 TDI MT	1
DLS TC	1
Star Zxi	1
S6 Plus 7 Seater	1
Alpha AT	1
Ambiente Diesel	1
1.6 MPI Ambition	1
2016-2021 TRD 4X4 AT	1
Crysta 2016-2020 Touring Sport 2.4 ZX AT	1
Dzire VXI AT Optional	1
BSIV LXI	1
LX 8 Str	1
1.5 GXI CVT	1
Active 4X2	1
NXT G80 K2	1
HTK Plus D	1
Rapid 1.0 TSI Ambition	1
HTK Plus AT D	1
Series 740Li DPE Signature	1
Neo 2 LT L 9 Str	1
CR4	1
ZX CVT	1
i-VTEC CVT V	1
Sportz Plus Diesel	1
V MT AVN	1
M2 8Str	1
Adventure Edition 85PS RXE	1
LT LPG	1
XE D	1
S i VTEC	1
Magna Plus Diesel	1
Ambiente 2.0 TDI CR MT	1
LTZ 4X2 AT	1
1.4 ZXi TDCi ABS	1
W7 BSIV	1
2016-2021 S i-DTEC	1
Cross	1
2.5 EV Diesel MS 7 Str BSIII	1
1.5 Diesel Trend BSIV	1
Quadrajet XMS	1
Transform VGT CRDi with Audio BS III	1
Sportz AT 1.4	1
1.2 Ti-VCT Sports Edition	1
E350 Petrol	1
Hatchback 1.2	1
Alpha 1.5	1
1.6 CRDi AT SX	1
sDrive 20d Sportline	1
2015-2019 1.2 MPI Highline	1

300 mHAWK100 T8 AMT	1
85PS Diesel RxL Option	1
2015-2019 1.2 MPI Highline Plus	1
100 mFALCON D75 K6 Plus	1
Rx S1	1
W4 Diesel	1
SV	1
Transform CRDi VGT ABS	1
Intelli Hybrid S6 Plus	1
2015-2017 Era 1.2	1
1.2L Diesel Ambiente	1
Sportz AMT	1
LXI CNG	1
M6 8Str	1
ZXI ABS	1
Expression E 220 d BSIV	1
S Turbo	1
Onyx 1.5 TDI MT BSIV	1
Petrol LT 7 Seater	1
Climber 1.0 AMT	1
T Plus CNG	1
HTX G	1
1.5 Ti VCT MT Trend	1
1.0 E	1
2017-2021 VXi AMT	1
1.5 Revotorq XZ	1
Liva VX	1
SX Opt Turbo Dualtone	1
1.2 Highline Plus AT 16 Alloy	1
Magnific 1.6 Comfortline	1
Diesel Style	1
2.4 GDi MT	1
90 Excellence Lounge	1
1.6 VTVT AT SX Plus	1
VTi-L MT	1
xDrive 20d M Sport	1
2016-2021 2.8 2WD MT BSIV	1
1.4 CX VTVT	1
Plus A Petrol	1
R LX BSII	1
E4 BS III	1
2015-2019 1.0 MPI Highline Plus	1
Range Rover Sport 3.0 D SE	1
Discovery 2017-2021 HSE 3.0 Si6	1
E Diesel	1
B1-10 seats BSII	1
2015-2019 1.0 MPI Highline	1
Fe 2WD AT	1
Class 200 Sport	1
1.0 LT Option Pack w/ Airbag	1
1.05 Revotorq XT	1
1.5 Petrol Titanium BSIV	1
1.2 Asta Option	1
SX Opt Diesel AT	1
800 LXI Optional	1
ML 350 CDI	1

Lxi - BSIII	1
2012-2014 Asta 1.4 AT	1
D4	1
1.5 TDI Highline Plus AT BSIV	1
Range Rover Evoque 2.0 TD4 SE Dynamic	1
KRAZ Plus	1
E i-Dtech	1
S CNG	1
i10 1.2 CRDi Magna	1
VTVT 1.6 SX Option	1
1.2 VTWT SX	1
Dzire Tour S CNG	1
sDrive 20d M Sport	1
E8	1
Vibe 1.5 dCi D6	1
Asta Option CVT BSIV	1
AT VXi Plus	1
2.8 SFX BSIV Dual Tone	1
1.2 S Option MT	1
B2 8 Seats BSIII	1
RS	1
RS245	1
Petrol ZXI	1
2016-2021 TRD Sportivo 2.8 2WD AT	1
VDI SHVS	1
2.5 ZX Diesel 7 Seater BSIII	1
E350 CDI Elegance	1
1.5 Petrol Titanium	1
Alpha Automatic	1
Elegance 2.0 TDI CR MT	1
1.6 CRDi SX Plus	1
1.0 MPI Comfortline BSIV	1
1.5 DV5 MT Ambiente	1
W8 Option Diesel BSIV	1
2006-2020 3.0 TDI Quattro Premium Plus	1
Crysta 2016-2020 2.4 G MT 8S BSIV	1
1.5 Petrol Trend Plus AT BSIV	1
Era Petrol	1
1.6 SX Option Executive Diesel	1
LS 9 Str BS IV	1
Lx BSIV	1
2.0 TDI Multitronic	1
Power Up 1.3 Dynamic	1
RS ZXi Plus	1
Active SX Diesel	1
R DUO LPG	1
3.0 TDI Quattro	1
100 mFALCON D75 K4 Plus	1
Petrol LXI	1
i DTec VX	1
E6	1
LS ABS	1
sDrive 18i	1
i10 CRDi SportZ Edition	1
Dzire AMT VDI	1
W11 Option	1

1.5 D4 BSIV	1
2.5 G4 Diesel 7-seater	1
Stepway 110PS RXZ 8S	1
Range Rover Evoque 2.0 TD4 HSE	1
2017-2021 VXI CNG	1
30 TDI quattro Technology	1
85PS Diesel RxZ	1
i10 Nios Sportz CRDi	1
1.4 Limited	1
1.4 Duratorq LXI	1
1.6 CRDi S	1
Magna Optional 1.4 CRDi	1
Getaway	1
2016-2021 VX Diesel	1
1.5 V AT Sunroof	1
1.4 Asta Dual Tone	1
Diesel Trend	1
2019-2020 XM	1
Liva Diesel	1
800 LX	1
XV D	1
2000 DSZ AC	1
Logan 1.4 GLE Petrol	1
K10 LX	1
Ambiente 2.0 MPI	1
SX Opt Diesel BSIV	1
Sports Diesel	1
Sportz BSIV	1
Std MPFi	1
Lxi BSIII	1
XM Diesel S	1
Style Edition V CVT	1
Sportz Plus Dual Tone BSIV	1
i10 Magna AT	1
Discovery Sport SD4 HSE Luxury	1
Xing XG AT eRLX Euro III	1
2011-2019 DI NON AC BS III SILVER	1
i-DTEC ZX	1
800 LXI Airbag	1
VXi BSIII CNG	1
2017-2021 ZXI AMT	1
2.5 EV MS 8 STR BSIV	1
Quadrajet LS	1
X VXI	1
2.8 SFX BSIII Dual Tone	1
2.5 VX 8 STR BSIV	1
110PS Diesel RxZ AMT	1
5 Seater Standard BSIV	1
1.4 TDI Active	1

Name: Car_Model, dtype: int64

1. Value counts and Sub-category Check For 'Super Charger'

```
In [129... df['Super Charger'].value_counts()
```

```
Out[129]: No      8497  
Yes      12  
NO       4  
Name: Super Charger, dtype: int64
```

There is correction in subcategories of Super Charger.

```
In [130... df['Super Charger'].replace('NO', 'No', inplace= True)
```

```
In [131... df['Super Charger'].value_counts()
```

```
Out[131]: No      8501  
Yes      12  
Name: Super Charger, dtype: int64
```

2. Value counts and Sub-category Check For 'Turbo Charger'

```
In [132... df['Turbo Charger'].value_counts()
```

```
Out[132]: No      5034  
Yes      3570  
no       6  
YES      5  
Twin     3  
yes      2  
Name: Turbo Charger, dtype: int64
```

```
In [133... df['Turbo Charger'].replace('no', 'No', inplace= True)  
df['Turbo Charger'].replace('YES', 'Yes', inplace= True)  
df['Turbo Charger'].replace('yes', 'Yes', inplace= True)
```

```
In [134... df['Turbo Charger'].value_counts()
```

```
Out[134]: No      5040  
Yes      3577  
Twin     3  
Name: Turbo Charger, dtype: int64
```

3. Value counts and Sub-category Check For 'Front Brake Type'.

```
In [135... df['Front Brake Type'].value_counts()
```

```
Out[135]: Disc          4575
          Ventilated Disc      3288
          Solid Disc           188
          Ventilated Discs     94
          Disc & Caliper Type   54
          Ventilated DIsc      49
          Disk                  37
          Drum                 22
          Ventilated Disk       20
          Disc, 236 mm          13
          Ventilated discs      13
          Vantilated Disc       12
          Ventlated Disc        9
          Ventillated Disc      5
          Vacuum assisted hydraulic dual circuit w 4
          Ventillated Discs     3
          disc                  3
          264mm Ventilated discs 3
          Booster assisted ventilated disc 3
          Multilateral Disc     1
          Tandem master cylinder with Servo assist 1
          Ventilated disc        1
          Discs                 1
          Disc,internally ventilated 1
          Single Piston Sliding Caliper, Vented Di 1
          260mm discs            1
          Disc brakes             1
Name: Front Brake Type, dtype: int64
```

Correction in subcategies

```
In [136...]: df['Front Brake Type'].replace('Ventilated Discs','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('Ventilated DIsc','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('Disk','Disc', inplace = True)
df['Front Brake Type'].replace('Ventilated Disk','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('Ventilated discs','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('Disc, 236 mm','Disc (236 mm)', inplace = True)
df['Front Brake Type'].replace('Vantilated Disc','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('Ventlated Disc','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('Ventillated Disc','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('Ventillated Discs','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('264mm Ventilated discs','Ventilated Disc (264mm)', inplace = True)
df['Front Brake Type'].replace('disc','Disc', inplace = True)
df['Front Brake Type'].replace('260mm discs','Disc (260mm)', inplace = True)
df['Front Brake Type'].replace('Disc brakes','Disc', inplace = True)
df['Front Brake Type'].replace('Discs','Disc', inplace = True)
df['Front Brake Type'].replace('Disc,internally ventilated','Ventilated Disc', inplace = True)
df['Front Brake Type'].replace('Ventilated disc','Ventilated Disc', inplace = True)
```

```
In [137...]: df['Front Brake Type'].value_counts()
```

```
Out[137]: Disc          4617
          Ventilated Disc      3495
          Solid Disc           188
          Disc & Caliper Type    54
          Drum                  22
          Disc (236 mm)         13
          Vacuum assisted hydraulic dual circuit w   4
          Ventilated Disc (264mm) 3
          Booster assisted ventilated disc       3
          Multilateral Disc      1
          Tandem master cylinder with Servo assist  1
          Single Piston Sliding Caliper, Vented Di  1
          Disc (260mm)           1
          Name: Front Brake Type, dtype: int64
```

4. Value counts and Sub-category Check For 'Rear Brake Type'.

```
In [138... df['Rear Brake Type'].value_counts()
```

```
Out[138]: Drum          7269
          Disc           671
          Ventilated Disc 105
          Solid Disc        85
          Disc & Caliper Type 54
          Leading-Trailing Drum 53
          Self-Adjusting Drum 51
          Discs            31
          Drums             22
          Ventilated Discs 18
          Disc & Drum        16
          Drums 180 mm       13
          Self Adjusting Drums 10
          Ventilated discs   8
          Self Adjusting Drum 7
          Self adjusting drums 5
          Vacuum assisted hydraulic dual circuit w   4
          262mm Disc & Drum Combination 3
          Drum in disc        3
          self adjusting drums 3
          Leading & Trailing Drum 3
          Booster assisted drum 3
          drum               3
          Self adjusting Drums 3
          Ventilated Drum     2
          228.6 mm dia, drums on rear wheels 1
          203mm Drums          1
          Single Piston Sliding Caliper, Vented Di  1
          Drum in Discs        1
          Name: Rear Brake Type, dtype: int64
```

Correction In Sub Categories

```
In [139... df['Rear Brake Type'].replace('Discs','Disc',inplace= True)
df['Rear Brake Type'].replace('Drums','Drum',inplace= True)
df['Rear Brake Type'].replace('Ventilated Discs','Ventilated Disc',inplace= True)
```

```
df['Rear Brake Type'].replace('Self Adjusting Drums','Self-Adjusting Drum',inplace=True)
df['Rear Brake Type'].replace('Self Adjusting Drum','Self-Adjusting Drum',inplace=True)
df['Rear Brake Type'].replace('Self adjusting drums','Self-Adjusting Drum',inplace=True)
df['Rear Brake Type'].replace('Self adjusting Drums','Self-Adjusting Drum',inplace=True)
df['Rear Brake Type'].replace('Ventilated discs','Ventilated Disc',inplace=True)
df['Rear Brake Type'].replace('Drum in disc','Disc & Drum',inplace=True)
df['Rear Brake Type'].replace('self adjusting drums','Self-Adjusting Drum',inplace=True)
df['Rear Brake Type'].replace('drum','Drum',inplace=True)
df['Rear Brake Type'].replace('203mm Drums','Drum (203mm)',inplace=True)
df['Rear Brake Type'].replace('228.6 mm dia, drums on rear wheels','Drum (228.6 mm)')
df['Rear Brake Type'].replace('Drum in Discs','Disc & Drum',inplace=True)
```

In [140... df['Rear Brake Type'].value_counts()

```
Out[140]:
```

Drum	7294
Disc	702
Ventilated Disc	131
Solid Disc	85
Self-Adjusting Drum	79
Disc & Caliper Type	54
Leading-Trailing Drum	53
Disc & Drum	20
Drums 180 mm	13
Vacuum assisted hydraulic dual circuit w	4
262mm Disc & Drum Combination	3
Booster assisted drum	3
Leading & Trailing Drum	3
Ventilated Drum	2
Drum (228.6 mm)	1
Single Piston Sliding Caliper, Vented Di	1
Drum (203mm)	1
Name: Rear Brake Type, dtype: int64	

5. Value counts and Sub-category Check For 'Steering Type'.

In [141... df['Steering Type'].value_counts()

```
Out[141]:
```

Power	7605
Electric	472
Manual	230
Electrical	80
EPAS	22
Electronic	14
Hydraulic	6
power	4
Name: Steering Type, dtype: int64	

In [142... df['Steering Type'].replace('power','Power', inplace =True)
df['Steering Type'].replace('Electrical','Electric', inplace =True)

In [143... df['Steering Type'].value_counts()

```
Out[143]: Power      7609  
Electric     552  
Manual       230  
EPAS          22  
Electronic    14  
Hydraulic     6  
Name: Steering Type, dtype: int64
```

6. Value counts and Sub-category Check For 'Gear Box'.

```
In [144...]  
pd.set_option('display.max_rows',None)  
df['Gear Box'].value_counts()
```

Out[144]:	5 Speed	6324
	6 Speed	1005
	5	205
	5-Speed	205
	4 Speed	202
	8 Speed	179
	7 Speed	174
	6-Speed	58
	5 speed	58
	9 Speed	50
	CVT	47
	4-Speed	38
	7-Speed S-Tronic	16
	7-Speed DCT	11
	7 Speed CVT	11
	Five Speed Manual Transmission	11
	7-Speed	11
	5 Speed CVT	11
	7 Speed DSG	8
	5 Speed AT+ Paddle Shifters	8
	8-Speed	8
	5-speed	8
	6 speed	7
	5 Manual	7
	8 Speed Sport	6
	7 Speed DCT	6
	Five Speed	6
	7 Speed 7G-DCT	5
	7G DCT 7-Speed Dual Clutch Transmission	5
	5 Speed Manual	4
	6-Speed IVT	4
	7-speed Stronic	4
	8 Speed Tiptronic	4
	9 speed Tronic	4
	6-speed DCT	3
	6-speed AutoSHIFT	3
	Fully Automatic	3
	5 Speed Manual Transmission	3
	Five Speed Manual Transmission Gearbox	3
	8 Speed Tip Tronic S	2
	9G-TRONIC	2
	5 Speed Manual (Cable Type Gear Shift)	2
	6 Speed Automatic	2
	6 Speed MT	2
	6-Speed IMT	2
	6-speed	2
	10 speed	2
	8-Speed Steptronic	2
	Six Speed Gearbox	2
	7 Speed S tronic	1
	8-Speed DCT	1
	5 GearBox	1
	6 Speed AT	1
	6	1
	5 Gears	1
	7-Speed DSG	1

Six Speed Manual with Paddle Shifter	1
8 Speed Steptronic	1
6-Speed DCT	1
AGS	1
Automatic Transmission	1
6 Speed iMT	1
Name: Gear Box, dtype: int64	

In [145...]

```
df['Gear Box'].replace('5','5 Speed',inplace= True)
df['Gear Box'].replace('5-Speed','5 Speed',inplace= True)
df['Gear Box'].replace('5 Gears','5 Speed',inplace= True)
df['Gear Box'].replace('Five Speed Manual Transmission','5 Speed',inplace= True)
df['Gear Box'].replace('5-speed','5 Speed',inplace= True)
df['Gear Box'].replace('Five Speed ','5 Speed',inplace= True)
df['Gear Box'].replace('6-Speed','6 Speed',inplace= True)
df['Gear Box'].replace('5 GearBox','5 Speed',inplace= True)
df['Gear Box'].replace('6','6 Speed',inplace= True)
df['Gear Box'].replace('7-Speed','7 Speed',inplace= True)
df['Gear Box'].replace('7-speed Stronic','7 Speed S tronic',inplace= True)
df['Gear Box'].replace('7 Speed S tronic','7 Speed',inplace= True)
```

In [146...]

```
df['Gear Box'].replace('5 speed','5 Speed',inplace= True)
df['Gear Box'].replace('5 Manual','5 Speed',inplace= True)
df['Gear Box'].replace('Five Speed','5 Speed',inplace= True)
df['Gear Box'].replace('5 Speed Manual','5 Speed',inplace= True)
df['Gear Box'].replace('5 Speed Manual Transmission','5 Speed',inplace= True)
df['Gear Box'].replace('Five Speed Manual Transmission Gearbox','5 Speed',inplace= True)
df['Gear Box'].replace('5 Speed Manual (Cable Type Gear Shift)','5 Speed',inplace= True)
```

In [147...]

```
df['Gear Box'].replace('4-Speed','4 Speed',inplace= True)
df['Gear Box'].replace('7-Speed DCT','7 Speed DCT',inplace= True)
df['Gear Box'].replace('7G DCT 7-Speed Dual Clutch Transmission','7 Speed DCT',inplace= True)
df['Gear Box'].replace('7 Speed 7G-DCT','7 Speed DCT',inplace= True)
df['Gear Box'].replace('7-Speed DSG','7 Speed DSG',inplace= True)
df['Gear Box'].replace('Six Speed Manual with Paddle Shifter','6 Speed',inplace= True)
df['Gear Box'].replace('7-Speed DSG','7 Speed DSG',inplace= True)
df['Gear Box'].replace('6-speed DCT','6 Speed DCT',inplace= True)
df['Gear Box'].replace('6 Speed MT','6 Speed',inplace= True)
df['Gear Box'].replace('6-Speed DCT','6 Speed DCT',inplace= True)
df['Gear Box'].replace('8-Speed','8 Speed',inplace= True)
df['Gear Box'].replace('8-Speed DCT','8 Speed DCT',inplace= True)
```

In [148...]

```
df['Gear Box'].replace('6 speed','6 Speed',inplace= True)
df['Gear Box'].replace('6-Speed IMT','6 Speed iMT',inplace= True)
df['Gear Box'].replace('6 Speed Automatic','6 Speed AT',inplace= True)
df['Gear Box'].replace('6-speed','6 Speed',inplace= True)
df['Gear Box'].replace('Six Speed Gearbox','6 Speed',inplace= True)
df['Gear Box'].replace('9 speed Tronic','9 Speed Tronic',inplace= True)
```

In [149...]

```
df['Gear Box'].replace('5 Speed AT+ Paddle Shifters','5 Speed',inplace= True)
df['Gear Box'].replace('8-Speed Steptronic','8 Speed Steptronic',inplace= True)
```

In [150...]

```
df['Gear Box'].replace('Automatic Transmission','Fully Automatic',inplace= True)
df['Gear Box'].replace('AGS','Auto Shift Gear',inplace= True)
```

```
df['Gear Box'].replace('9G-TRONIC', '9 Speed Tronic', inplace= True)
```

```
In [151... df['Gear Box'].value_counts()
```

```
Out[151]: 5 Speed          6846  
6 Speed           1078  
4 Speed            240  
7 Speed            190  
8 Speed            187  
9 Speed             50  
CVT                 47  
7 Speed DCT         27  
7-Speed S-Tronic     16  
5 Speed CVT          11  
7 Speed CVT          11  
7 Speed DSG           9  
8 Speed Sport          6  
9 Speed Tronic          6  
8 Speed Tiptronic        4  
6-Speed IVT            4  
Fully Automatic        4  
6 Speed DCT            4  
8 Speed Steptronic        3  
6 Speed AT              3  
6-speed AutoSHIFT        3  
6 Speed iMT              3  
10 speed                2  
8 Speed Tip Tronic S       2  
8 Speed DCT              1  
Auto Shift Gear           1  
Name: Gear Box, dtype: int64
```

7. Value counts and Sub-category Check For 'Tyre Volume'.

```
In [123... df['Tyre Volume'].value_counts()
```

```
Out[123]: Tyre Volume      5379
Tubeless                               1530
Tubeless Tyres                         603
Tubeless, Radial                       601
Radial                                 125
Tubeless, Radial Tyre                  91
Tubeless, Radials                      37
Tubeless Radial Tyres                 31
Radial, Tubless                        29
Tubless, Radial                        29
Radial Tubeless                        18
Runflat                               13
Tubless, Radial                        13
Tubeless Radial                        6
Runflat Tyre                           5
Radial, Tubeless                       5
tubeless tyre                          4
Runflat Tyres                          4
Run-Flat                              3
Radial Tyres                           3
Tubeless Radial Tyre                  2
Tubeless, Runflat                     1
Tubless Radial Tyrees                 1
Tubeless Radials                       1
Tubeless, Runflat                     1
Radial, Tubeless                       1
Name: Tyre Volume, dtype: int64
```

```
In [152... df['Tyre Volume'].replace('Tubeless, Radial', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Tubeless Tyres', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Tubeless, Radial', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Radial', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Tubeless Tyres, Radial', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Tubeless, Radials', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Tubeless Radial Tyres', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Radial, Tubless', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Tubless, Radial', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Radial Tubeless', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Tubeless Radial', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Radial, Tubeless', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('tubeless tyre', 'Tubeless', inplace =True)
df['Tyre Volume'].replace('Tubeless Radials', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Tubless Radial Tyrees', 'Tubeless Radial Tyre', inplace =True)
```

```
In [153... df['Tyre Volume'].replace('Tubless, Radial', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Radial, Tubeless', 'Tubeless Radial Tyre', inplace =True)
df['Tyre Volume'].replace('Runflat', 'Runflat Tyre', inplace =True)
df['Tyre Volume'].replace('Runflat Tyres', 'Runflat Tyre', inplace =True)
df['Tyre Volume'].replace('Run-Flat', 'Runflat Tyre', inplace =True)
df['Tyre Volume'].replace('Tubeless, Runflat', 'Tubeless Runflat Tyre', inplace =True)
df['Tyre Volume'].replace('Tubeless, Runflat', 'Tubeless Runflat Tyre', inplace =True)
```

Missing value Imputation

- Numerical value can be impute with Mean and Median. We can decide imputation method based on boxplot & Distplot.
- Categorical variable & ordinal variable can be impute with mode.

In [156...]: `df.isnull().sum()`

```
Out[156]: Fuel Type          26
KMs driven                  0
Engine Displacement(CC)     0
Transmission                0
Milage(kmpl)                27
Max Power(bhp)              121
Torque(Nm)                  145
Seating Capacity             487
Color                        1860
Gear Box                     128
Steering Type               453
Front Brake Type            483
Rear Brake Type             437
Tyre Volume                 350
Engine Type                  378
No of Cylinder               403
Turbo Charger                266
Super Charger                373
Length(mm)                  176
Width(mm)                   173
Height(mm)                  172
Price (Rs.)                  0
Car_Age                      0
Car_Brand                    0
Car_Model                    1660
dtype: int64
```

In [157...]: `# Imputation of Categorical variable or ordinal variable with mode of category`

```
df['Fuel Type'].fillna(df['Fuel Type'].mode()[0], inplace = True)
df['Seating Capacity'].fillna(df['Seating Capacity'].mode()[0], inplace = True)
df['Color'].fillna(df['Color'].mode()[0], inplace = True)
df['Gear Box'].fillna(df['Gear Box'].mode()[0], inplace = True)
df['Steering Type'].fillna(df['Steering Type'].mode()[0], inplace = True)
df['Front Brake Type'].fillna(df['Front Brake Type'].mode()[0], inplace = True)
df['Rear Brake Type'].fillna(df['Rear Brake Type'].mode()[0], inplace = True)
df['Color'].fillna(df['Color'].mode()[0], inplace = True)
df['Tyre Volume'].fillna(df['Tyre Volume'].mode()[0], inplace = True)
df['Engine Type'].fillna(df['Engine Type'].mode()[0], inplace = True)
df['No of Cylinder'].fillna(df['No of Cylinder'].mode()[0], inplace = True)
df['Turbo Charger'].fillna(df['Turbo Charger'].mode()[0], inplace = True)
df['Super Charger'].fillna(df['Super Charger'].mode()[0], inplace = True)
```

In [178...]: `df['Car_Model'].fillna('Unknown', inplace = True)`

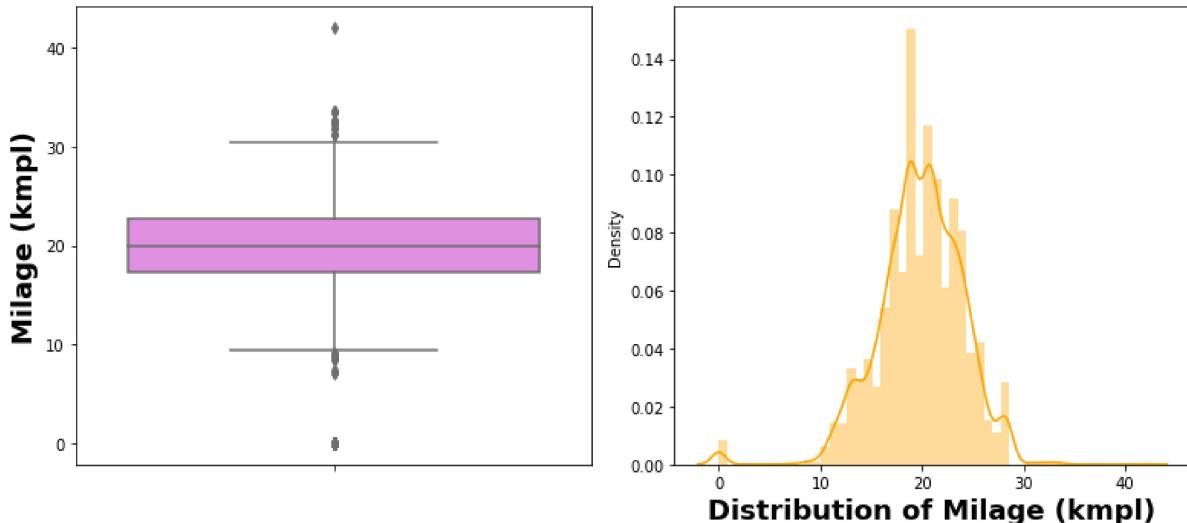
1. Impuation of Missing values in Milage

In [158...]: `plt.figure(figsize = (11,5))`
`plt.subplot(1,2,1)`

```

sns.boxplot(y='Milage(kmpl)', data= df, color = 'violet')
plt.ylabel('Milage (kmpl)',fontweight = 'bold',fontsize= 18)
plt.subplot(1,2,2)
sns.distplot(df['Milage(kmpl)'], color ='Orange')
plt.xlabel('Distribution of Milage (kmpl)',fontweight = 'bold',fontsize= 18)
plt.tight_layout()
plt.show()

```



```

In [159... print("Mean of Milage(kmpl):",df['Milage(kmpl)'].mean(),'kmpl')
print("Median of Milage(kmpl):",df['Milage(kmpl)'].median(),'kmpl')

```

Mean of Milage(kmpl): 19.82472965345972 kmpl
 Median of Milage(kmpl): 20.0 kmpl

Milage (kmpl) is almost symmetrical in nature. Outliers are also spread to both lower & upper bound.

So, We will be impute Milage (kmpl) with mean.

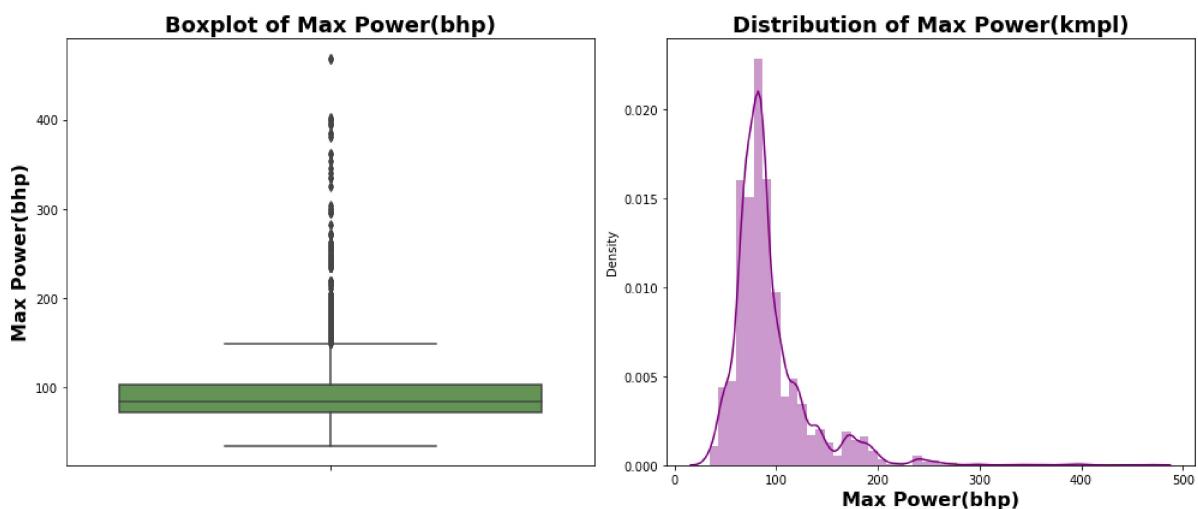
```
In [160... df['Milage(kmpl)'].fillna(df['Milage(kmpl)'].mean(), inplace =True)
```

2. Imputation of missing values in Max Power

```

In [161... plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.boxplot(y = df['Max Power(bhp)'], palette='gist_earth_r')
plt.title('Boxplot of Max Power(bhp)',fontweight ='bold', fontsize = 18)
plt.ylabel('Max Power(bhp)', fontweight ='bold', fontsize = 16 )
plt.subplot(1,2,2)
sns.distplot( x = df['Max Power(bhp)'],color ='Purple' )
plt.title('Distribution of Max Power(kmpl)',fontweight ='bold', fontsize = 18)
plt.xlabel('Max Power(bhp)',fontweight ='bold', fontsize = 16)
plt.tight_layout()
plt.show()

```



```
In [162...]: print("Mean of Max Power:", df['Max Power(bhp)'].mean(), 'bhp')
print("Median of Max Power:", df['Max Power(bhp)'].median(), 'bhp')
```

Mean of Max Power: 92.73147233314377 bhp
 Median of Max Power: 83.1 bhp

1. Distribution of Max Power is right Skewed and from boxplot we can see presence of outliers.

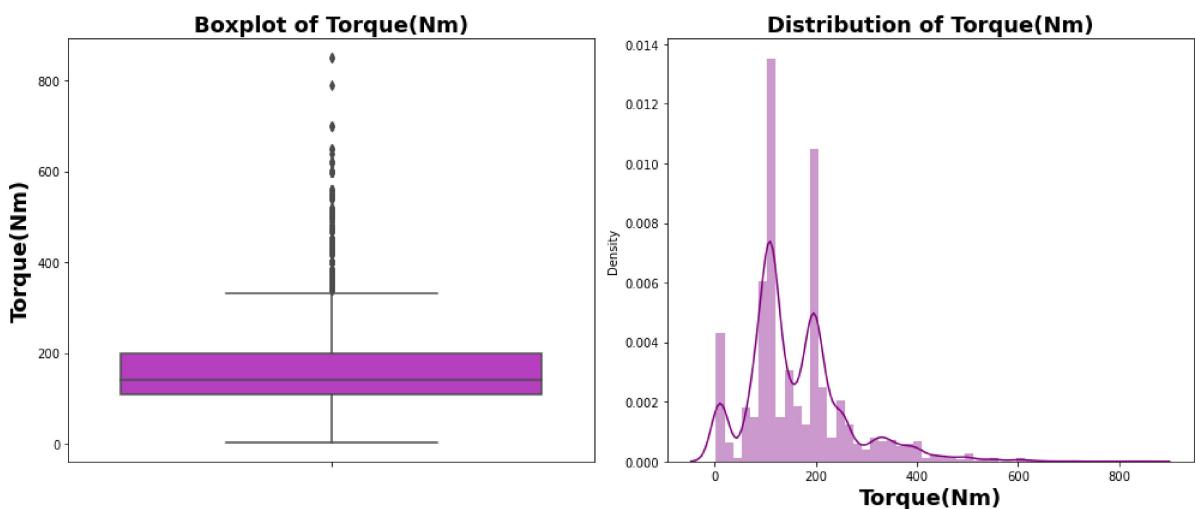
2. Mean is greater than median.

We gone impute Max Power with Median as it is less sensitive to outliers.

```
In [163...]: df['Max Power(bhp)'].fillna(df['Max Power(bhp)'].median(), inplace = True)
```

3. Imputation of missing values in Torque

```
In [164...]: plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.boxplot(y = df['Torque(Nm)'], palette='gnuplot2')
plt.title('Boxplot of Torque(Nm)', fontweight ='bold', fontsize = 18)
plt.ylabel('Torque(Nm)', fontweight ='bold', fontsize = 18 )
plt.subplot(1,2,2)
sns.distplot( x = df['Torque(Nm)'], color ='Purple' )
plt.title('Distribution of Torque(Nm)', fontweight ='bold', fontsize = 18)
plt.xlabel('Torque(Nm)', fontweight ='bold', fontsize = 18)
plt.tight_layout()
plt.show()
```



```
In [165...]: print("Mean of Torque(Nm):",df['Torque(Nm)'].mean(),'Nm')
print("Median of Torque(Nm):",df['Torque(Nm)'].median(),'Nm')
```

Mean of Torque(Nm): 162.3707607825189 Nm

Median of Torque(Nm): 140.0 Nm

1. Outliers are present in upper bound of Torque(Nm)

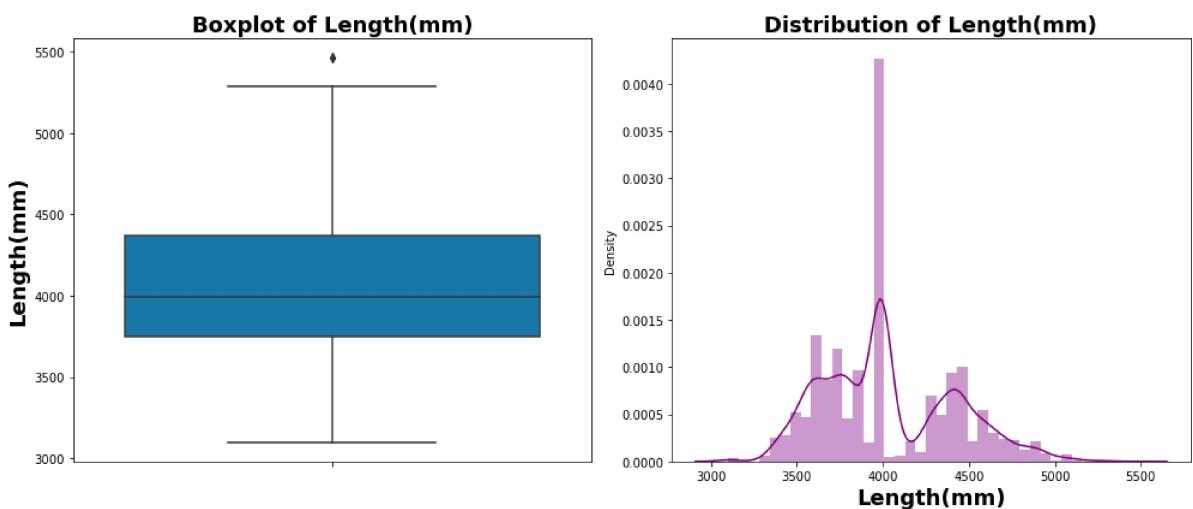
2. Torque is right skewed in nature.

We gone imputate Torque with Median as it is less sensitive to outliers.

```
In [166...]: df['Torque(Nm)'].fillna(df['Torque(Nm)'].median(), inplace =True)
```

4. Imputation of missing values in Length(mm)

```
In [167...]: plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.boxplot(y = df['Length(mm)'], palette='winter_r')
plt.title('Boxplot of Length(mm)',fontweight ='bold', fontsize = 18)
plt.ylabel('Length(mm)', fontweight ='bold', fontsize = 18 )
plt.subplot(1,2,2)
sns.distplot( x = df['Length(mm)'],color ='Purple' )
plt.title('Distribution of Length(mm)',fontweight ='bold', fontsize = 18)
plt.xlabel('Length(mm)',fontweight ='bold', fontsize = 18)
plt.tight_layout()
plt.show()
```



```
In [168...]: print("Mean of Length(mm):", df['Length(mm)'].mean(), 'mm')
print("Median of Length(mm):", df['Length(mm)'].median(), 'mm')
```

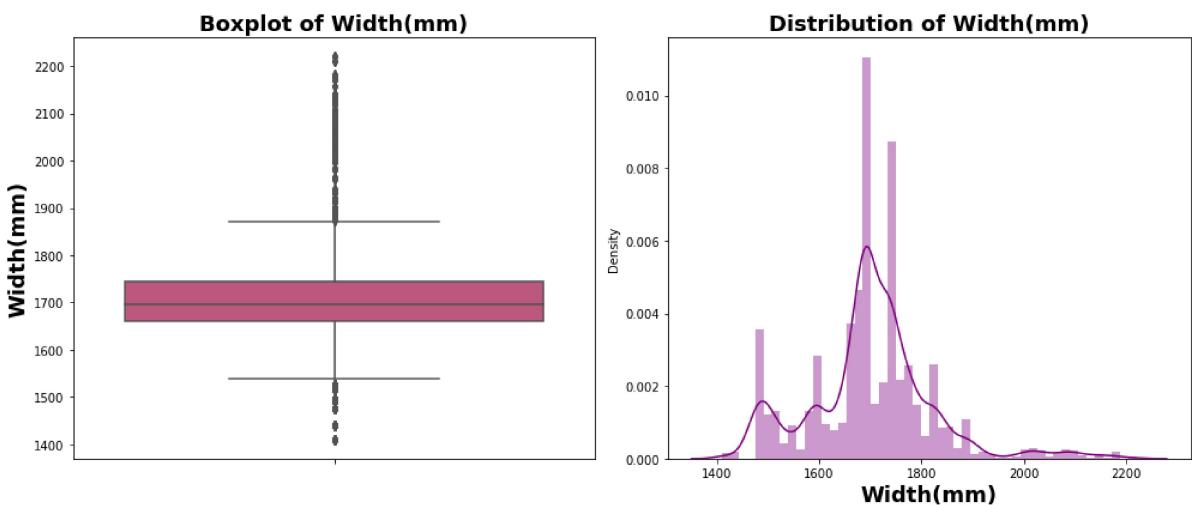
Mean of Length(mm): 4028.7541905855337 mm
 Median of Length(mm): 3995.0 mm

We gone impute length with mean as almost no outliers present.

```
In [169...]: df['Length(mm)'].fillna(df['Length(mm)'].mean(), inplace = True)
```

5. Imputation of missing values in Width(mm)

```
In [170...]: plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.boxplot(y = df['Width(mm)'], palette='plasma')
plt.title('Boxplot of Width(mm)', fontweight ='bold', fontsize = 18)
plt.ylabel('Width(mm)', fontweight ='bold', fontsize = 18 )
plt.subplot(1,2,2)
sns.distplot( x = df['Width(mm)'], color ='Purple' )
plt.title('Distribution of Width(mm)', fontweight ='bold', fontsize = 18)
plt.xlabel('Width(mm)', fontweight ='bold', fontsize = 18)
plt.tight_layout()
plt.show()
```



```
In [171... print("Mean of Width(mm):",df['Width(mm)'].mean(),'mm')
print("Median of Width(mm):",df['Width(mm)'].median(),'mm')
```

Mean of Width(mm): 1701.6738207276483 mm
 Median of Width(mm): 1695.0 mm

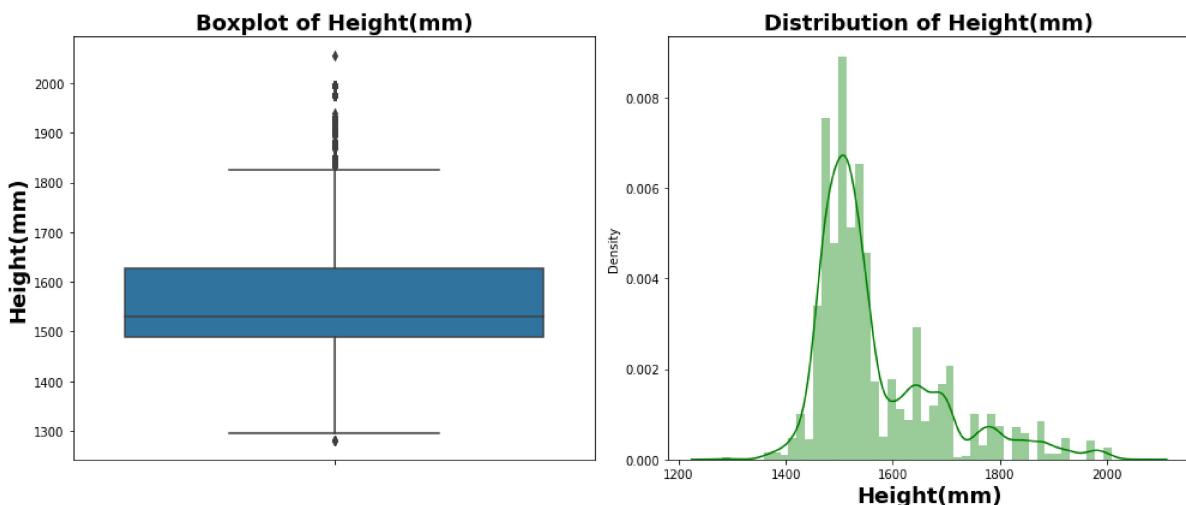
Outliers are present in width, as mean is sensitive to outliers we gone imputate outliers with Median.

```
In [172... df['Width(mm)'].fillna(df['Width(mm)'].mean(), inplace = True)
```

6. Imputation of missing values in Height (mm)

```
In [173... df.drop(df[df['Height(mm)']>5000].index, inplace =True)
```

```
In [174... plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.boxplot(y = df['Height(mm)'], palette='tab10')
plt.title('Boxplot of Height(mm)',fontweight ='bold', fontsize = 18)
plt.ylabel('Height(mm)', fontweight ='bold', fontsize = 18 )
plt.subplot(1,2,2)
sns.distplot( x = df['Height(mm)'],color ='Green' )
plt.title('Distribution of Height(mm)',fontweight ='bold', fontsize = 18)
plt.xlabel('Height(mm)',fontweight ='bold', fontsize = 18)
plt.tight_layout()
plt.show()
```



```
In [175... print("Mean of Height(mm):",df['Height(mm)'].mean(),'mm')
print("Median of Height(mm):",df['Height(mm)'].median(),'mm')
```

Mean of Height(mm): 1567.698232323233 mm
 Median of Height(mm): 1530.0 mm

1. Outliers are present in upper bound of Height(mm)

2. Height is right skewed in nature.

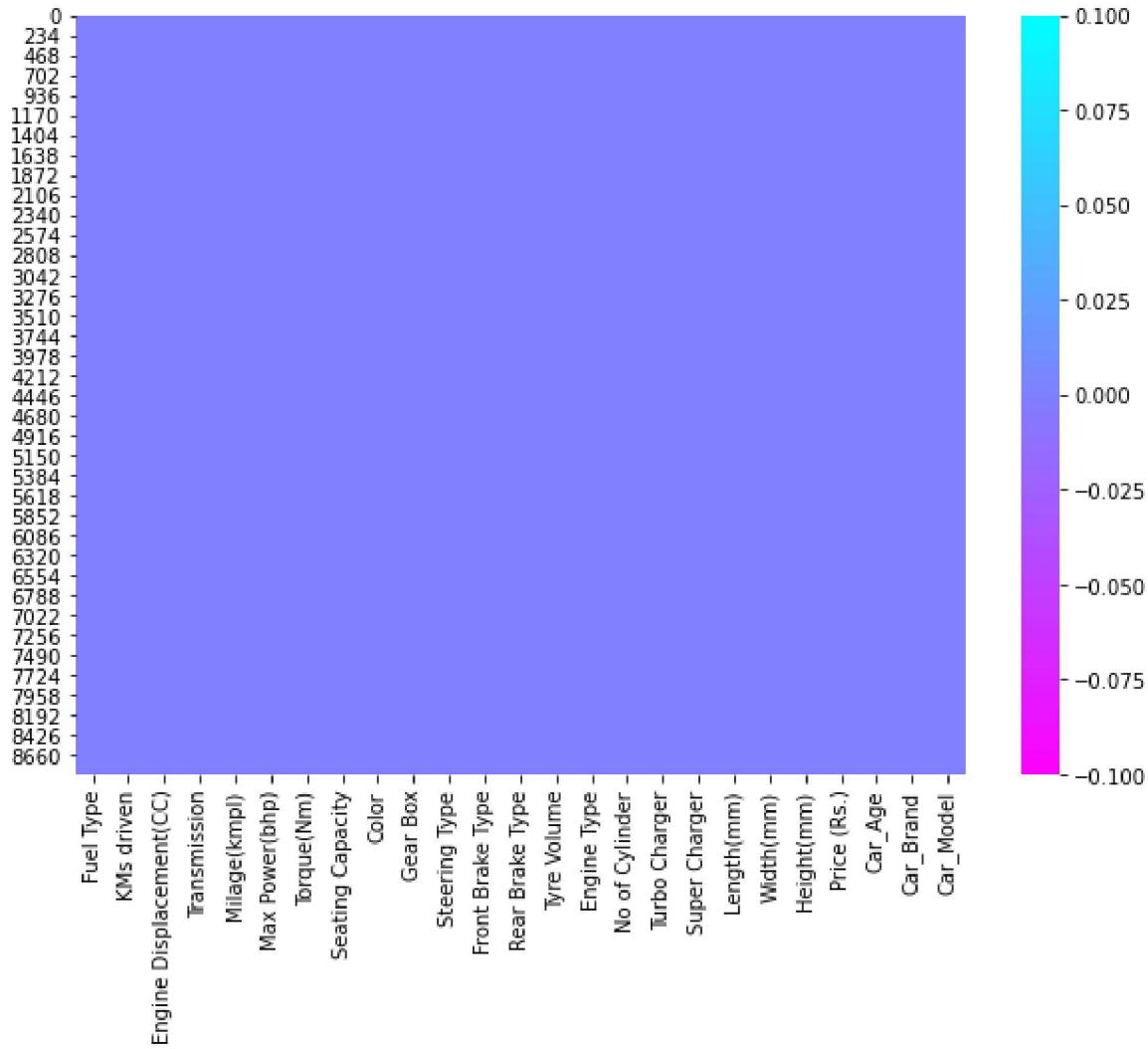
3. Mean is greater than median.

We gone impute Height with Median as it is less sensitive to outliers.

```
In [176...]: df['Height(mm)'].fillna(df['Height(mm)'].median(), inplace=True)
```

Missing Value check after imputation

```
In [179...]: plt.figure(figsize=(10,7))
sns.heatmap(df.isnull(), cmap="cool_r")
plt.show()
```



Comment :

Finally, No Missing Value is Present.

We are Now Yes To Go Further !!!

```
In [180...]: # Importing Final Clean data in excel file
df.to_excel('Car_used_price_Clean.xlsx', index = False)
```

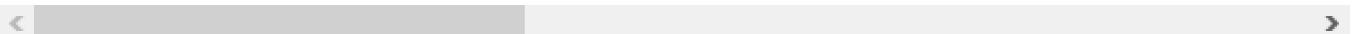
Importing Final clean data Excel file

```
In [2]: df = pd.read_excel('Car_used_price_Clean.xlsx')
```

```
In [3]: pd.set_option('display.max_columns',None)
df.head()
```

Out[3]:

	Fuel Type	KMs driven	Engine Displacement(CC)	Transmission	Milage(kmpl)	Max Power(bhp)	Torque(Nm)	Seating Capacity
0	Petrol	18600	1497	Manual	17.40	117.3	145.0	
1	Diesel	15000	1956	Manual	17.10	170.0	350.0	
2	Diesel	115000	2499	Manual	14.80	80.0	19.0	
3	Petrol	80000	1497	Manual	16.80	116.4	146.0	
4	Petrol	35000	1197	Manual	20.36	78.9	111.8	



Statistical Matrix

```
In [84]: df.describe().T.style.background_gradient(subset=['mean','std','50%','count'], cmap
```

Out[84]:

	count	mean	std	min	25%
KMs driven	8884.000000	66460.392954	48663.642255	201.000000	37000.000000
Engine Displacement(CC)	8884.000000	1391.822490	467.384613	0.000000	1197.000000
Milage(kmpl)	8884.000000	19.823632	4.197076	0.000000	17.400000
Max Power(bhp)	8884.000000	92.599130	37.438784	34.200000	73.000000
Torque(Nm)	8884.000000	161.997166	99.817097	3.000000	109.000000
No of Cylinder	8884.000000	3.792098	0.511602	0.000000	4.000000
Length(mm)	8884.000000	4028.761789	381.821466	3099.000000	3746.000000
Width(mm)	8884.000000	1701.675323	120.047417	1410.000000	1660.000000
Height(mm)	8884.000000	1566.968370	115.762687	1281.000000	1490.000000
Price (Rs.)	8884.000000	634944.644079	771983.056555	35000.000000	300000.000000
Car_Age	8884.000000	6.380009	3.516659	0.000000	4.000000



Observation :

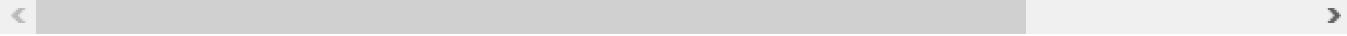
1. All Numerical value except Milage(kmpl) have value of mean greater than median.
2. If we look at minimum value column, clearly there exist error in columns like Engine Displacement(CC), Milage(kmpl), Seating Capacity, No of Cylinder as there minimum value cannot be zero.
3. Some of Columns contain outliers as seen by comparison of 75% column & max column.
4. Average Used car price is 6.34 lakh while maximum price is 95 lakh.

In [185...]

```
df.describe(include=object).style.background_gradient(cmap='summer_r')
```

Out[185]:

	Fuel Type	Transmission	Color	Gear Box	Steering Type	Front Brake Type	Rear Brake Type	Tyre Volume	Engine Type	Turbo Charger	Ch
count	8884	8884	8884	8884	8884	8884	8884	8884	8884	8884	8884
unique	5	2	198	26	6	13	17	5	338	3	1
top	Petrol	Manual	White	5 Speed	Power	Disc	Drum	Tubeless Radial Tyre	In-Line Engine	No	1
freq	4793	7455	3802	6972	8061	5098	7729	7320	1211	5306	1



Observation :

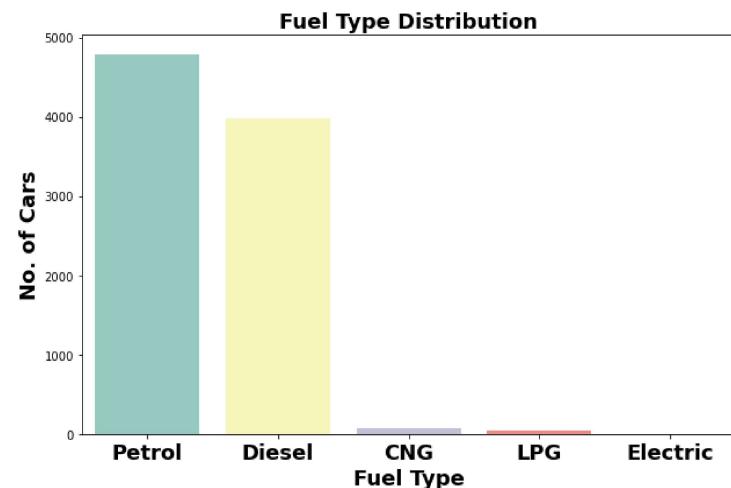
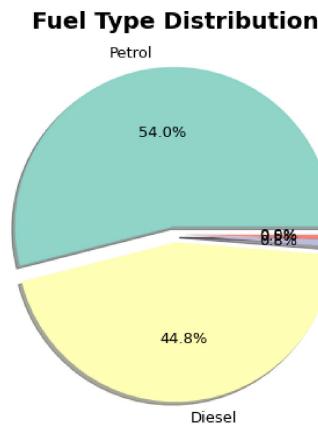
1. Most of car used Tubeless radial tyre compare to ran-flat tyre which are costly compare to earlier.
2. Majority of car are 5 seater with 5 speed gearbox having power steering facility. **We can conclude that most of used car low price range small car and there are very few SUV category car.**
3. **Most prefer color is white with manual transmission.**
4. Very few car contain Super Charger while majority of do not contain Turbo charger.

EDA

Exploring Fuel Type

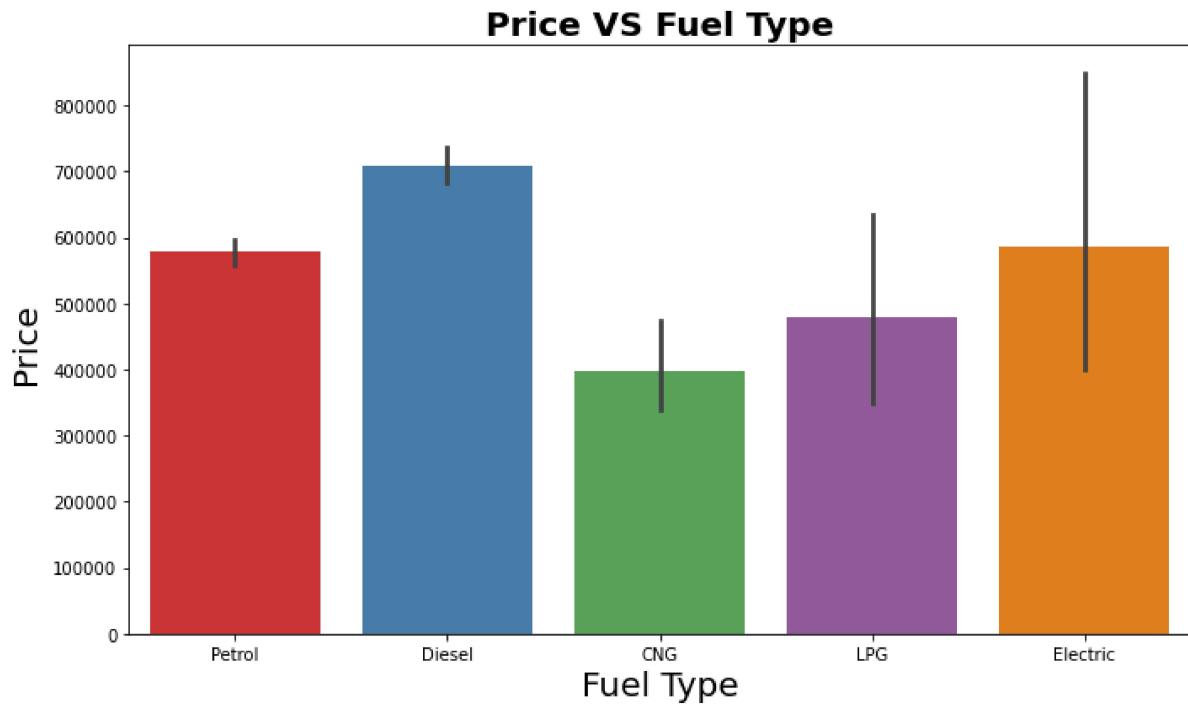
In [234...]

```
# Counting Fuel type
plt.rcParams["figure.autolayout"] = True
sns.set_palette('Set3')
f,ax=plt.subplots(1,2,figsize=(16,6))
df['Fuel Type'].value_counts().plot.pie(explode=[0.05,0.05,0.05,0.05,0.05], autopct=_
                                         textprops =[{'fontsize':13}], ax=ax[0], sha_
                                         ax[0].set_title('Fuel Type Distribution', fontsize=20, fontweight ='bold')
                                         ax[0].set_ylabel(' ')
                                         sns.countplot('Fuel Type',data=df,ax=ax[1])
                                         ax[1].set_title('Fuel Type Distribution', fontsize=18, fontweight ='bold')
                                         ax[1].set_xlabel("Fuel Type", fontsize=18, fontweight ='bold')
                                         ax[1].set_ylabel("No. of Cars", fontsize=18, fontweight ='bold')
                                         plt.xticks(fontsize=18, fontweight ='bold')
                                         plt.tight_layout()
                                         plt.show()
```



- Plot of Fuel Type Vs Price

```
In [19]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('Set1')
plt.figure(figsize =(10,6))
sns.barplot(y=df['Price (Rs.)'],x=df['Fuel Type'])
plt.title("Price VS Fuel Type",fontsize=20,fontweight ='bold')
plt.xlabel('Fuel Type',fontsize = 20)
plt.ylabel('Price ',fontsize = 20)
plt.tight_layout()
```



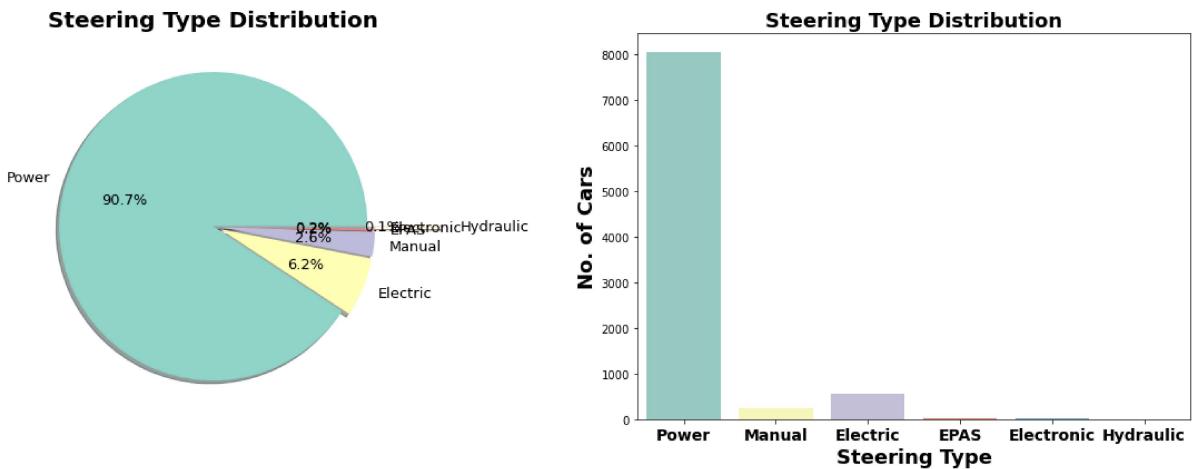
Comment :

- Most of car are Petrol operated followed by Diesel. This may be due to low prices of Petrol car compare to diesel car.
- Very small segment of electric car and also price is quite high compare to petrol based.
- CNG based car are Cheapest compare to others.

Exploring Steering Type

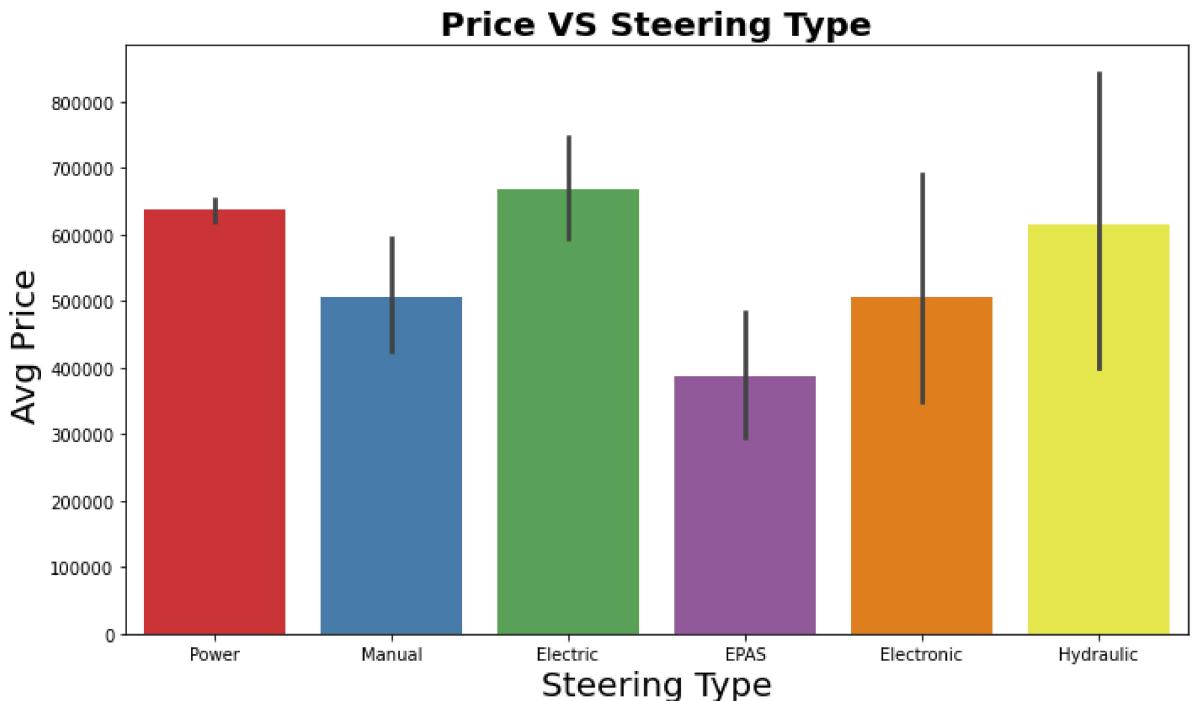
```
In [236...]:
# Exploring Steering Type
plt.rcParams["figure.autolayout"] = True
sns.set_palette('Set3')
f,ax=plt.subplots(1,2,figsize=(16,6))
df['Steering Type'].value_counts().plot.pie(explode=[0.0,0.05,0.05,0.05,0.05,0.5],a
                                              textprops =[{'fontsize':13}, ax=ax[0],sha
                                              ax[0].set_title('Steering Type Distribution', fontsize=20,fontweight ='bold')
                                              ax[0].set_ylabel('')
                                              sns.countplot('Steering Type',data=df,ax=ax[1])
                                              ax[1].set_title('Steering Type Distribution',fontsize=18,fontweight ='bold')
                                              ax[1].set_xlabel("Steering Type",fontsize=18,fontweight ='bold')
```

```
ax[1].set_ylabel("No. of Cars", fontsize=18, fontweight = 'bold')
plt.xticks(fontsize=14, fontweight = 'bold')
plt.show()
```



- **Price Vs Steering Type**

```
In [73]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('Set1')
plt.figure(figsize =(10,6))
sns.barplot(y=df['Price (Rs.)'],x=df['Steering Type'])
plt.title("Price VS Steering Type",fontsize=20,fontweight = 'bold')
plt.xlabel('Steering Type',fontsize = 20)
plt.ylabel('Avg Price ',fontsize = 20)
plt.tight_layout()
```



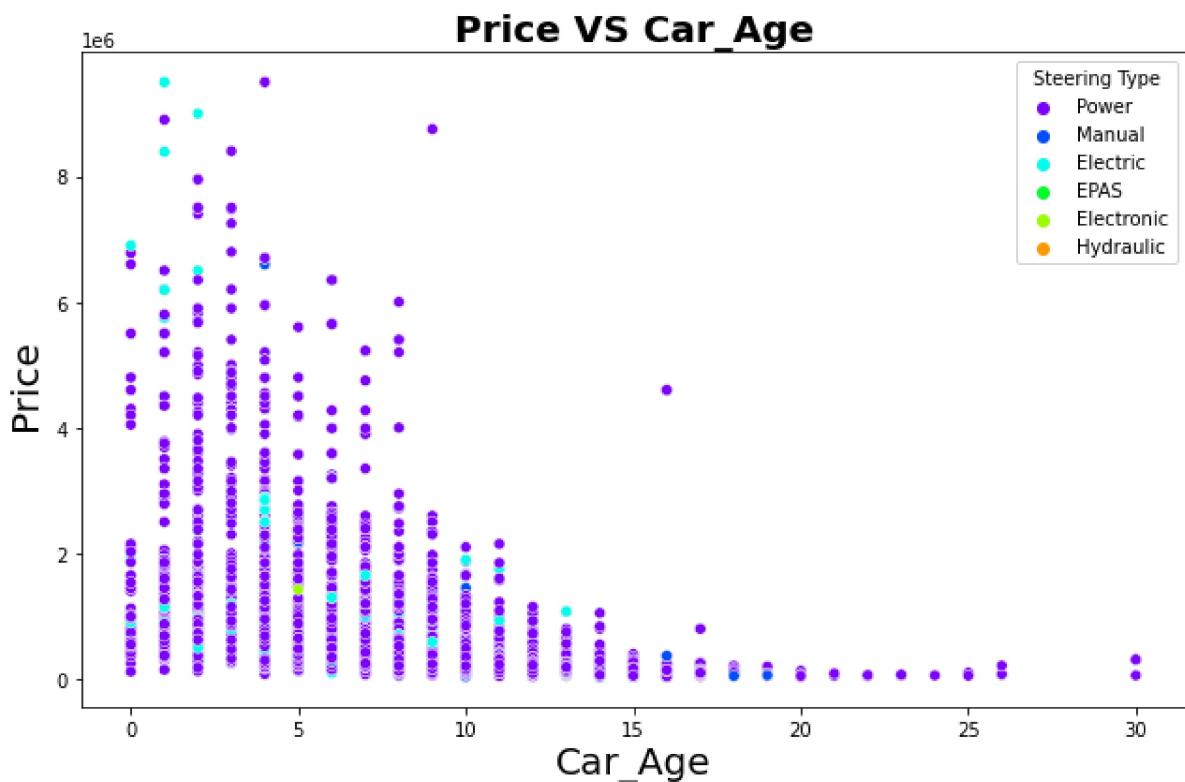
Comment :

- More than 90 % of car users prefer Power steering compare to others.

- 6.2% car based on electric steering, which is costly compare to others.
- Very small section of car still uses Manual Steering, Most probably they belong to old model.

Let check predication in last point by plotting Car age Vs Price based on steering types.

```
In [72]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('gist_rainbow_r')
plt.figure(figsize=(9,6))
sns.scatterplot(y=df['Price (Rs.)'],x=df['Car_Age'], hue=df['Steering Type'])
plt.title("Price VS Car_Age", fontsize=20, fontweight='bold')
plt.xlabel('Car_Age', fontsize=20)
plt.ylabel('Price', fontsize=20)
plt.tight_layout()
plt.show()
```



Here we got confirmation of prediction in previous section,

Almost all manual steering based car atleast 10 year old.

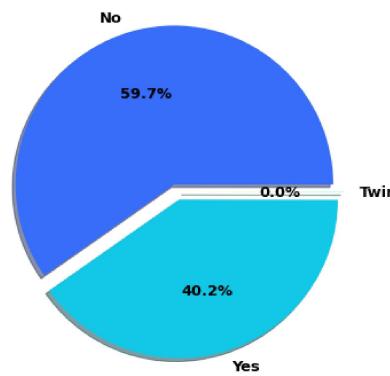
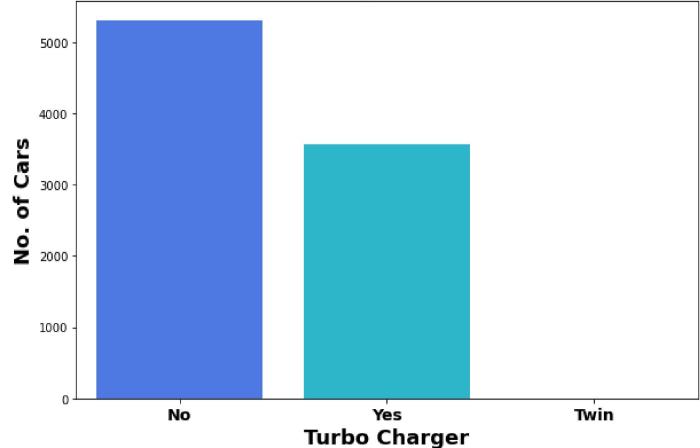
Exploring Steering Type

```
In [74]: # Exploring Steering Type
plt.rcParams["figure.autolayout"] = True
sns.set_palette('rainbow')
f,ax=plt.subplots(1,2,figsize=(16,6))
df['Turbo Charger'].value_counts().plot.pie(explode=[0.05,0.05,0.05], autopct='%2.1f',
                                             textprops = { 'fontweight': 'bold', 'fontsi
```

```

ax[0].set_title('Turbo Charger Distribution', fontsize=20, fontweight = 'bold')
ax[0].set_ylabel('')
sns.countplot('Turbo Charger', data=df, ax=ax[1])
ax[1].set_title('Turbo Charger Distribution', fontsize=18, fontweight = 'bold')
ax[1].set_xlabel("Turbo Charger", fontsize=18, fontweight = 'bold')
ax[1].set_ylabel("No. of Cars", fontsize=18, fontweight = 'bold')
plt.xticks(fontsize=14, fontweight = 'bold')
plt.show()

```

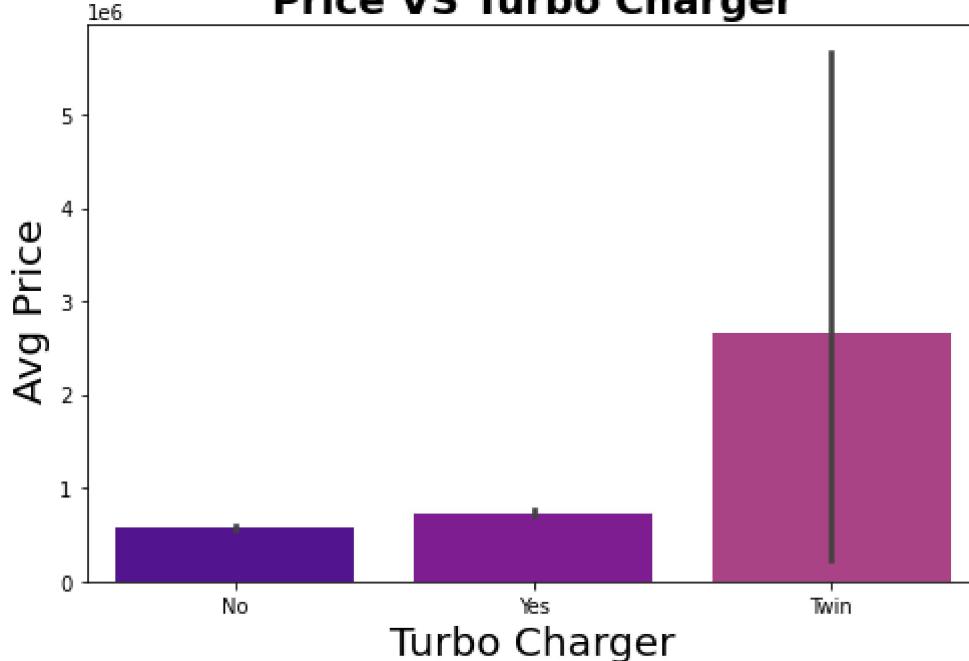
Turbo Charger Distribution**Turbo Charger Distribution**

In [75]:

```

plt.rcParams["figure.autolayout"] = True
sns.set_palette('plasma')
plt.figure(figsize =(7,5))
sns.barplot(y=df['Price (Rs.)'],x=df['Turbo Charger'])
plt.title("Price VS Turbo Charger",fontsize=20,fontweight = 'bold')
plt.xlabel('Turbo Charger',fontsize = 20)
plt.ylabel('Avg Price ',fontsize = 20)
plt.tight_layout()

```

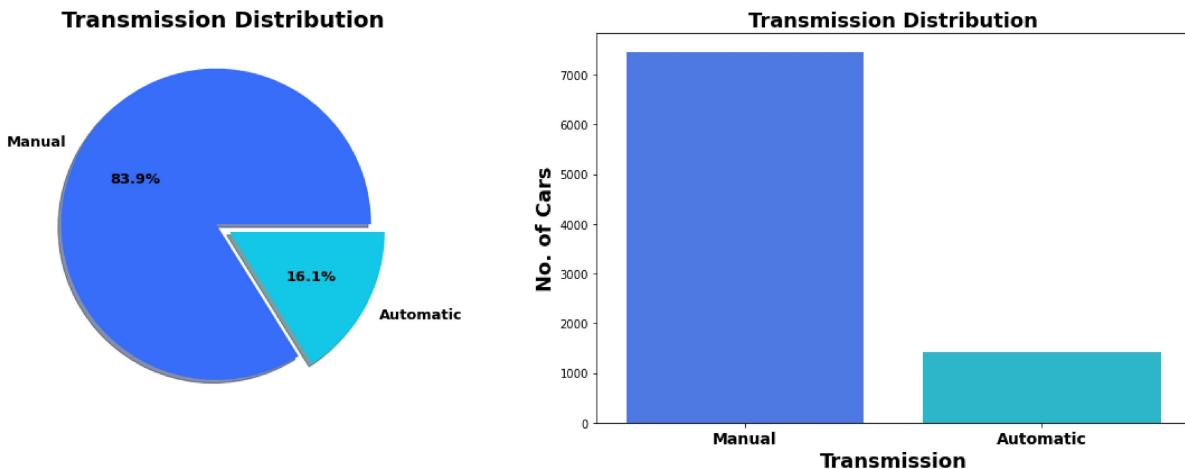
Price VS Turbo Charger**Comment :**

- 40% cars are with turbo charger & almost less than 1 % car with twin facility.
- As expected Max price for car based on Twin engine followed by with turbocharger.

Exploring Transmission

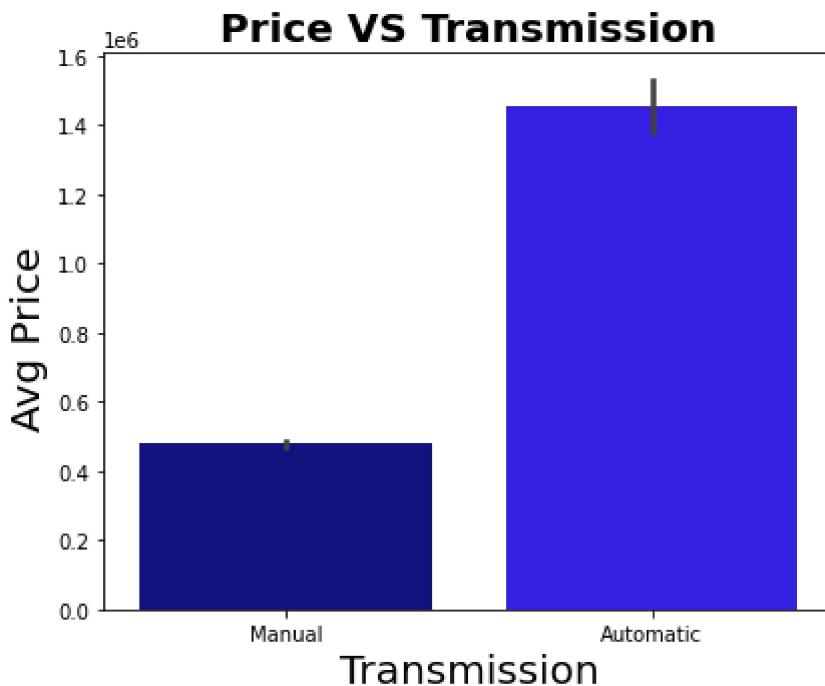
In [237...]

```
# Exploring Transmission
plt.rcParams["figure.autolayout"] = True
sns.set_palette('rainbow')
f,ax=plt.subplots(1,2,figsize=(16,6))
df['Transmission'].value_counts().plot.pie(explode=[0.05,0.05], autopct='%.2.1f%%',
                                             textprops = { 'fontweight': 'bold','fontsize':13}, ax=ax[0])
ax[0].set_title('Transmission Distribution', fontsize=20,fontweight ='bold')
ax[0].set_ylabel('')
sns.countplot('Transmission',data=df,ax=ax[1])
ax[1].set_title('Transmission Distribution',fontsize=18,fontweight ='bold')
ax[1].set_xlabel("Transmission",fontsize=18,fontweight ='bold')
ax[1].set_ylabel("No. of Cars",fontsize=18,fontweight ='bold')
plt.xticks(fontsize=14,fontweight ='bold')
plt.show()
```



In [76]:

```
plt.rcParams["figure.autolayout"] = True
sns.set_palette('gnuplot2')
plt.figure(figsize =(6,5))
sns.barplot(y=df['Price (Rs.)'],x=df['Transmission'])
plt.title("Price VS Transmission",fontsize=20,fontweight ='bold')
plt.xlabel('Transmission',fontsize = 20)
plt.ylabel('Avg Price ',fontsize = 20)
plt.tight_layout()
```

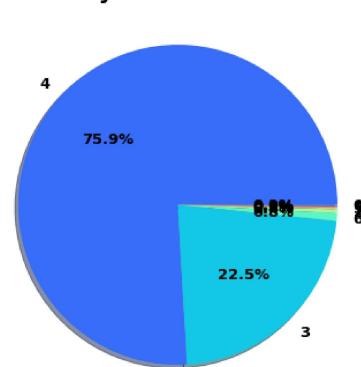
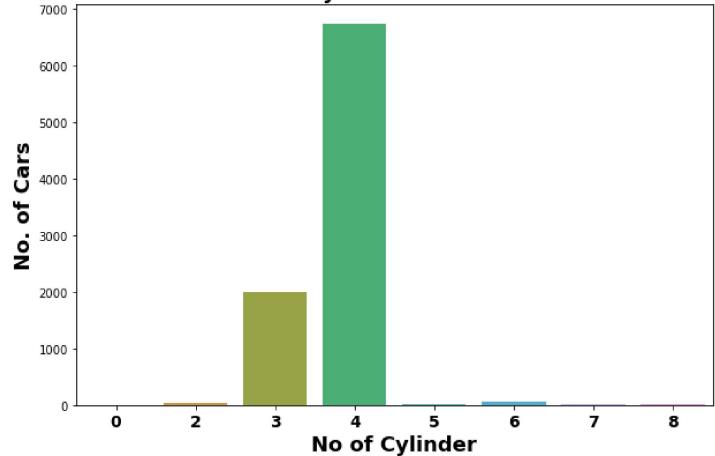


Comment :

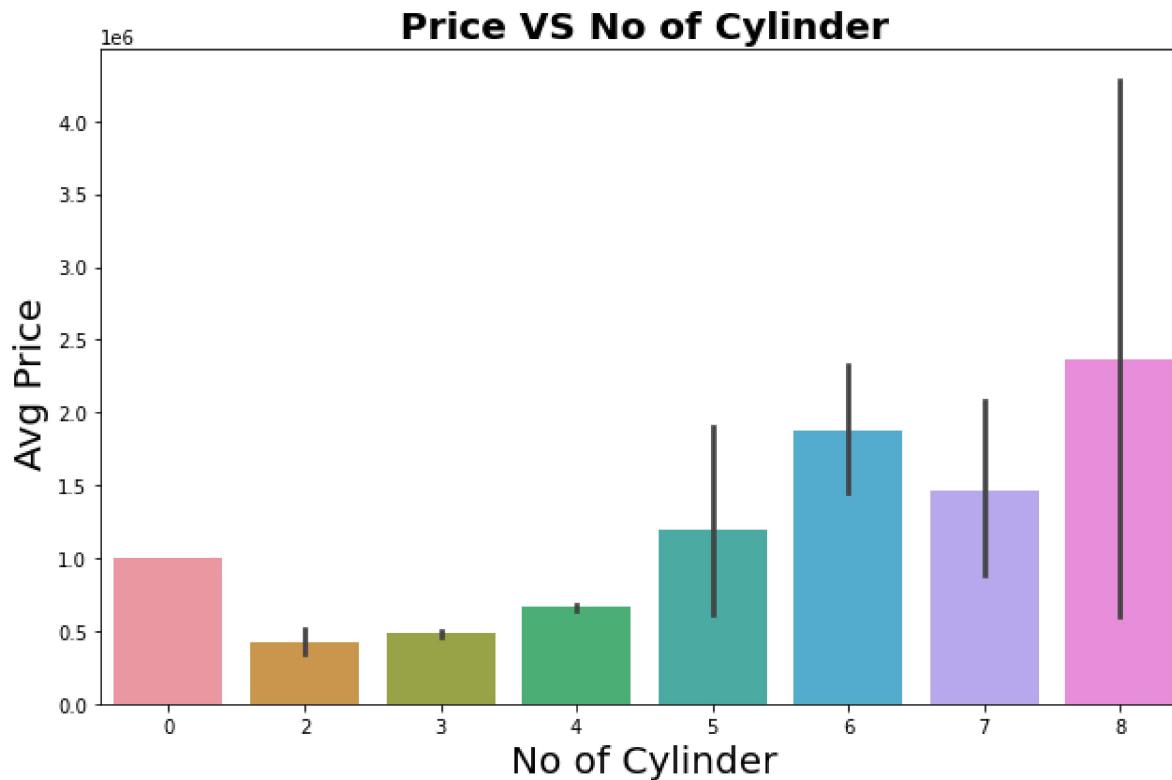
- Most of car are with mannal transmission.
- The price of Automatic transmission is much greater than manual transmission.

Exploring Price Vs No of Cylinder

```
In [78]: # Exploring Transmission
plt.rcParams["figure.autolayout"] = True
sns.set_palette('rainbow')
f,ax=plt.subplots(1,2,figsize=(16,6))
df['No of Cylinder'].value_counts().plot.pie(autopct='%2.1f%%',
                                             textprops =[ 'fontweight': 'bold','fontsize':13], ax=ax[0])
ax[0].set_title('No of Cylinder Distribution', fontsize=20,fontweight ='bold')
ax[0].set_ylabel('')
sns.countplot('No of Cylinder',data=df,ax=ax[1])
ax[1].set_title('No of Cylinder Distribution',fontsize=18,fontweight ='bold')
ax[1].set_xlabel("No of Cylinder",fontsize=18,fontweight ='bold')
ax[1].set_ylabel("No. of Cars",fontsize=18,fontweight ='bold')
plt.xticks(fontsize=14,fontweight ='bold')
plt.show()
```

No of Cylinder Distribution**No of Cylinder Distribution**

```
In [77]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('gnuplot2')
plt.figure(figsize =(9,6))
sns.barplot(y=df['Price (Rs.)'],x=df['No of Cylinder'])
plt.title("Price VS No of Cylinder",fontsize=20,fontweight ='bold')
plt.xlabel('No of Cylinder',fontsize = 20)
plt.ylabel('Avg Price ',fontsize = 20)
plt.tight_layout()
```

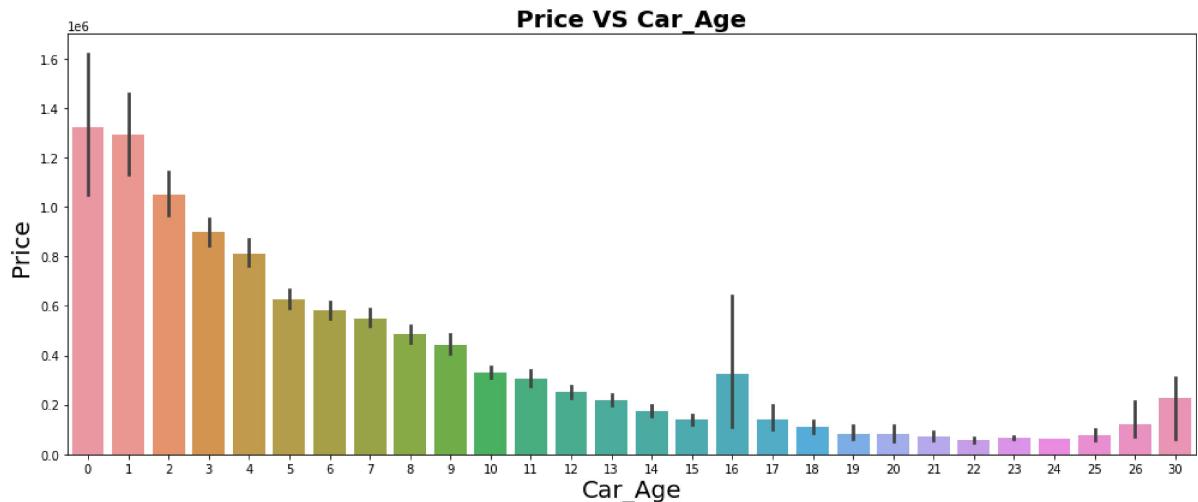


Comment :

- Most of Cars with are 4 cylinder engines followed by 3 cylinder engines.
- In terms of Avg Price as number of cylinder increases the average price increases.

Car Age Vs Average Price

```
In [35]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('gnuplot2')
plt.figure(figsize =(14,6))
sns.barplot(y=df['Price (Rs.)'],x=df['Car_Age'])
plt.title("Price VS Car_Age",fontsize=20,fontweight ='bold')
plt.xlabel('Car_Age',fontsize = 20)
plt.ylabel('Price ',fontsize = 20)
plt.tight_layout()
```

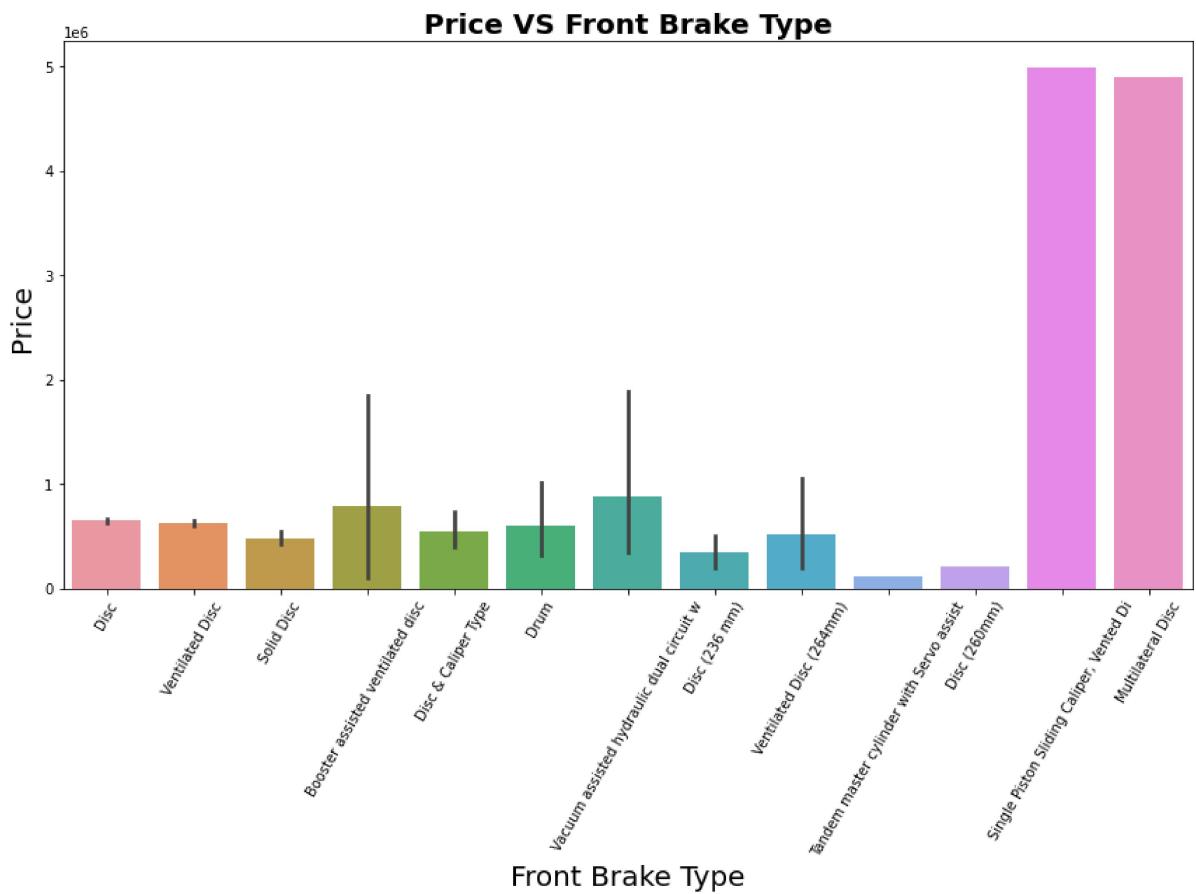


Comment :

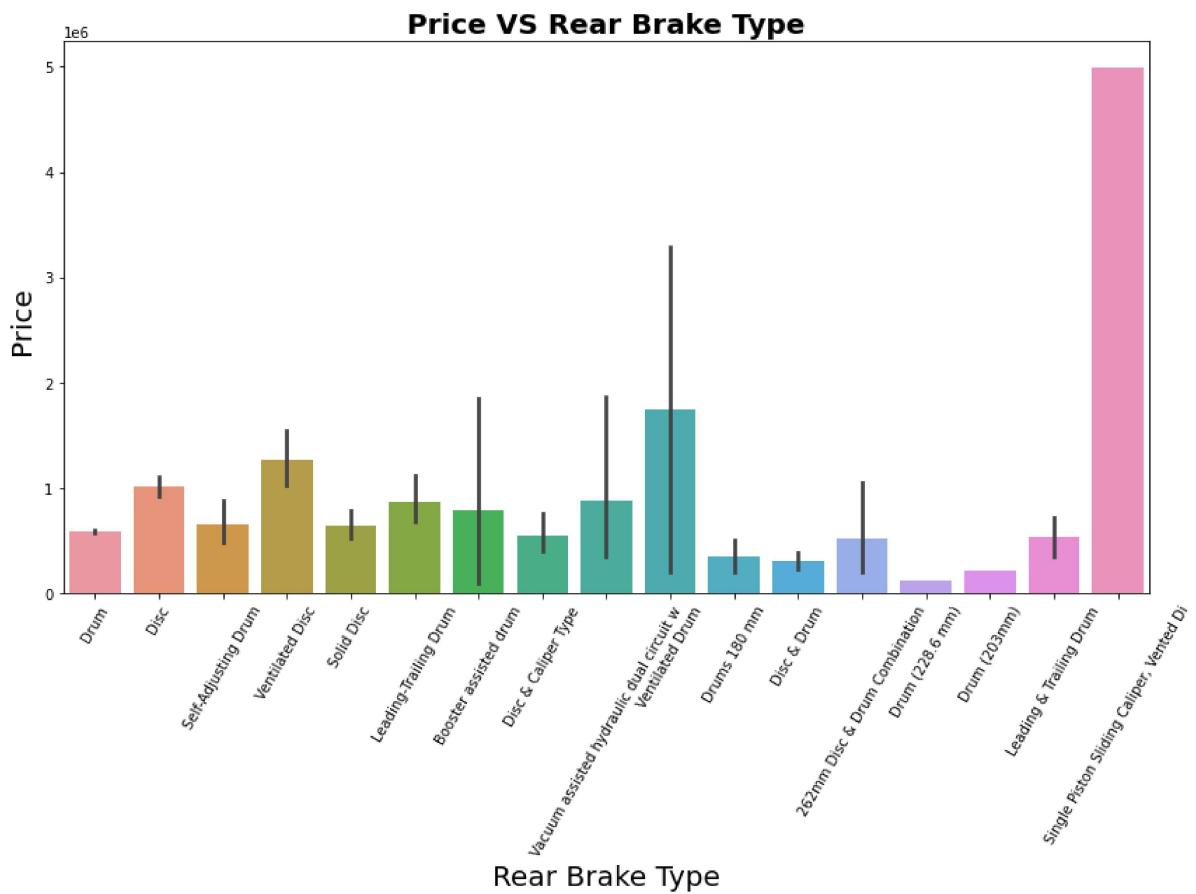
As car get older, price of car depreciates.

Exploration of brakes vs price

```
In [83]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('tab10_r')
plt.figure(figsize =(12,9))
sns.barplot(y=df['Price (Rs.)'],x=df['Front Brake Type'])
plt.title("Price VS Front Brake Type",fontsize=20,fontweight ='bold')
plt.xlabel('Front Brake Type',fontsize = 20)
plt.xticks(rotation=60)
plt.ylabel('Price ',fontsize = 20)
plt.tight_layout()
plt.show()
```

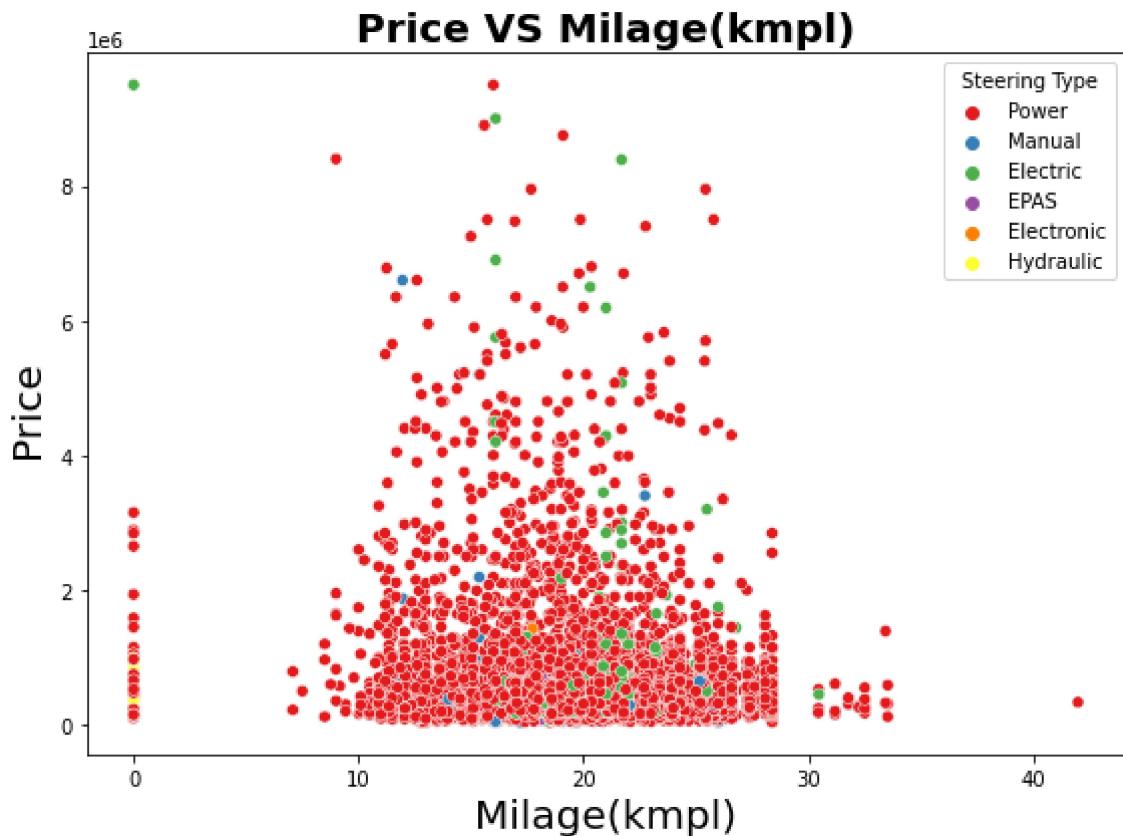


```
In [82]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('summer_r')
plt.figure(figsize =(12,9))
sns.barplot(y=df['Price (Rs.)'],x=df['Rear Brake Type'])
plt.title("Price VS Rear Brake Type",fontsize=20,fontweight ='bold')
plt.xlabel('Rear Brake Type',fontsize = 20)
plt.xticks(rotation=60)
plt.ylabel('Price ',fontsize = 20)
plt.tight_layout()
plt.show()
```



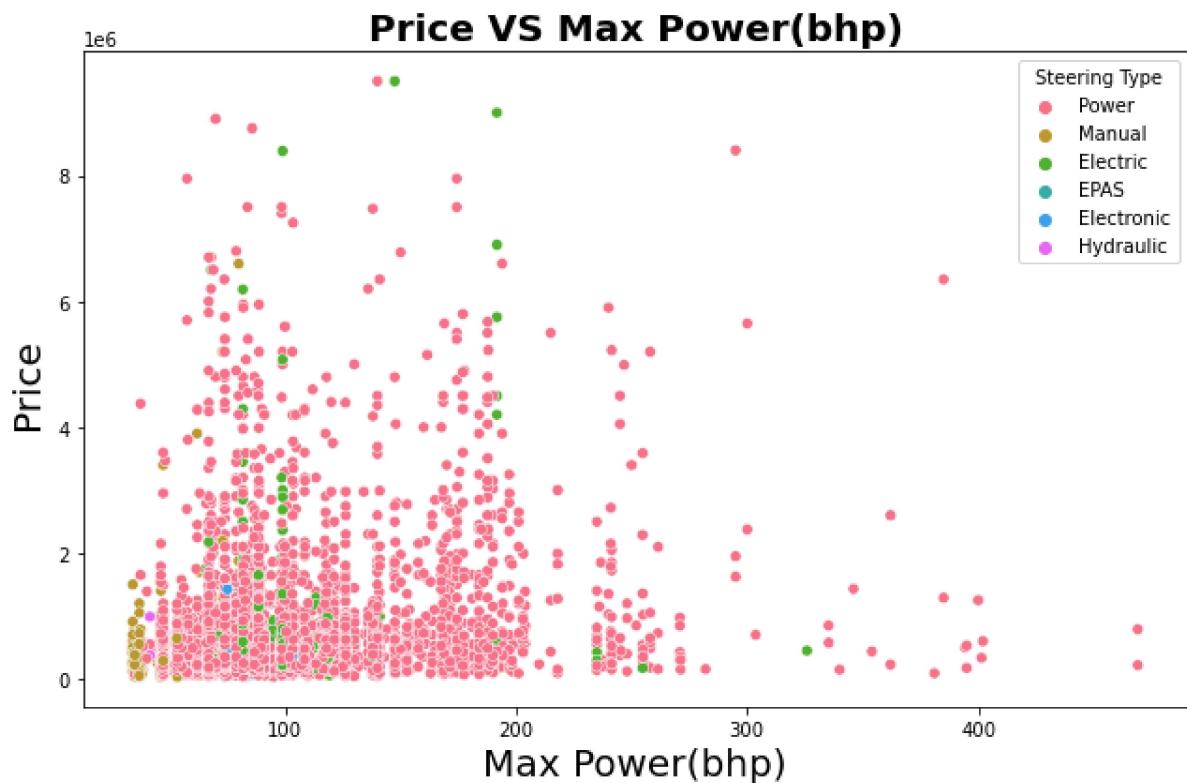
Price VS Milage

```
In [65]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('Set1')
plt.figure(figsize =(8,6))
sns.scatterplot(y=df['Price (Rs.)'],x=df[ 'Milage(kmpl)'],hue =df['Steering Type'])
plt.title("Price VS Milage(kmpl)",fontsize=20,fontweight ='bold')
plt.xlabel('Milage(kmpl)',fontsize = 20)
plt.ylabel('Price ',fontsize = 20)
plt.tight_layout()
plt.show()
```



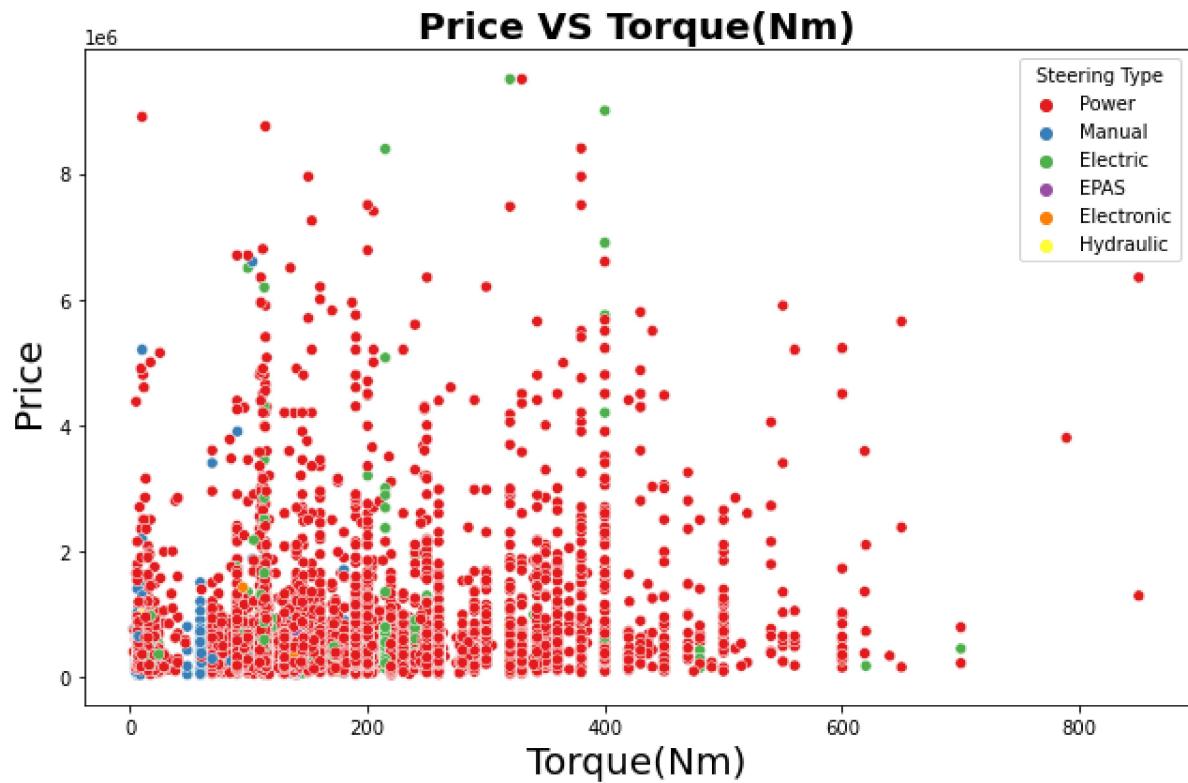
Milage (kmpl) varies in between 10 to 25 kmpl for most of cars. For Majority cars price is below 0.5e6. We didn't get any other significant relation between price and steering types.

```
In [60]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('husl')
plt.figure(figsize =(9,6))
sns.scatterplot(y=df['Price (Rs.)'],x=df['Max Power(bhp)'], hue =df['Steering Type'])
plt.title("Price VS Max Power(bhp)",fontsize=20,fontweight ='bold')
plt.xlabel('Max Power(bhp)',fontsize = 20)
plt.ylabel('Price ',fontsize = 20)
plt.tight_layout()
plt.show()
```



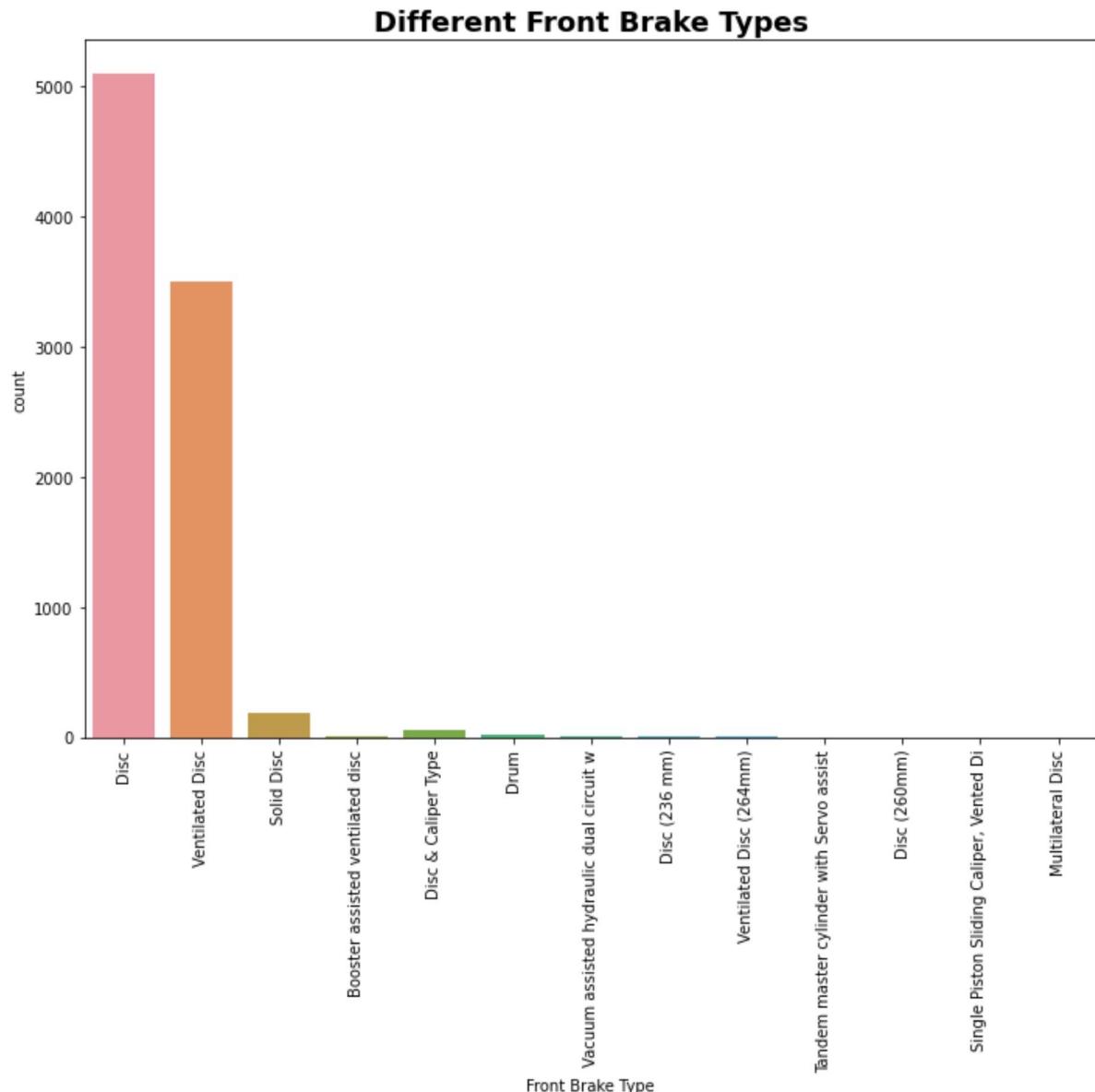
Majority of cars have max power below 200 bhp.

```
In [67]: plt.rcParams["figure.autolayout"] = True
sns.set_palette('Set1')
plt.figure(figsize =(9,6))
sns.scatterplot(y=df['Price (Rs.)'],x=df['Torque(Nm)'], hue =df['Steering Type'])
plt.title("Price VS Torque(Nm)",fontsize=20,fontweight ='bold')
plt.xlabel('Torque(Nm)',fontsize = 20)
plt.ylabel('Price ',fontsize = 20)
plt.tight_layout()
plt.show()
```



In [241...]

```
#Count plot for front_brake_type column
plt.figure(figsize=[10,10])
sns.countplot(df['Front Brake Type'])
plt.title('Different Front Brake Types', fontsize=18, fontweight = 'bold')
plt.xticks(rotation=90 );
```



Most of car used Disc brakes of various different types.

Feature Engineering

1. Encoding Categorical Data

```
In [186...]: # Splitting data in Numeric and categorical Variable.
Numerical = ['KMs driven', 'Engine Displacement(CC)', 'Car_Age','Milage(kmpl)', 'Ma
              'Seating Capacity', 'No of Cylinder', 'Length(mm)', 'Width(mm)', 'Heig
Categorical = ['Fuel Type', 'Transmission', 'Color', 'Gear Box', 'Steering Type', '
               'Rear Brake Type', 'Tyre Volume', 'Turbo Charger', 'Super Charger']
```

```
In [187...]: # Using Label encoder for transforming Categorical data
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

```
for i in Categorical:
    df[i] = le.fit_transform(df[i])
df.head()
```

Out[187]:

	Fuel Type	KMs driven	Engine Displacement(CC)	Transmission	Milage(kmpl)	Max Power(bhp)	Torque(Nm)	Seatin Capacit
0	4	18600	1497	1	17.40	117.3	145.0	
1	1	15000	1956	1	17.10	170.0	350.0	
2	1	115000	2499	1	14.80	80.0	19.0	
3	4	80000	1497	1	16.80	116.4	146.0	
4	4	35000	1197	1	20.36	78.9	111.8	

In [189...]: df['Engine Type'] = le.fit_transform(df['Engine Type'])

In [192...]: df['Car_Brand'] = le.fit_transform(df['Car_Brand'])
df['Car_Model'] = le.fit_transform(df['Car_Model'])

In [194...]: df.head()

Out[194]:

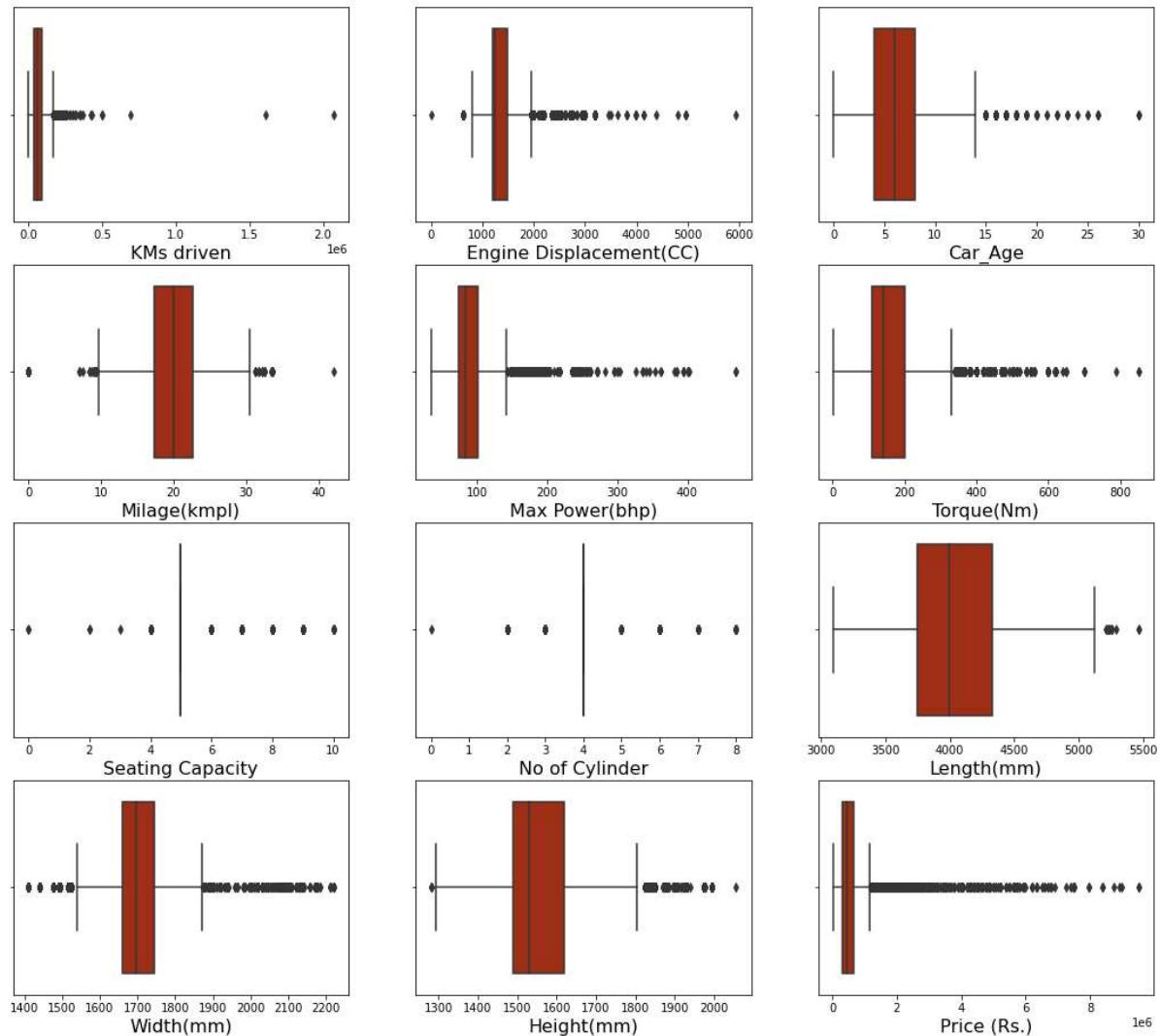
	Fuel Type	KMs driven	Engine Displacement(CC)	Transmission	Milage(kmpl)	Max Power(bhp)	Torque(Nm)	Seatin Capacit
0	4	18600	1497	1	17.40	117.3	145.0	
1	1	15000	1956	1	17.10	170.0	350.0	
2	1	115000	2499	1	14.80	80.0	19.0	
3	4	80000	1497	1	16.80	116.4	146.0	
4	4	35000	1197	1	20.36	78.9	111.8	

In [193...]: df.shape

Out[193]: (8884, 25)

2. Outliers Detection & Removal

```
In [10]: plt.figure(figsize=(18,16), facecolor = 'white')
plt_num = 1
for i in Numerical:
    if plt_num <= 12:
        ax = plt.subplot(4,3,plt_num)
        sns.boxplot(df[i], palette='gnuplot')
        plt.xlabel(i, fontsize= 16)
    plt_num += 1
plt.show()
```



From Boxplot we can see presence of outliers.

Since Data is realistic and data is error-free, we will build model with data we already have without removing outliers. Outliers are realistic data.

3. Correlation

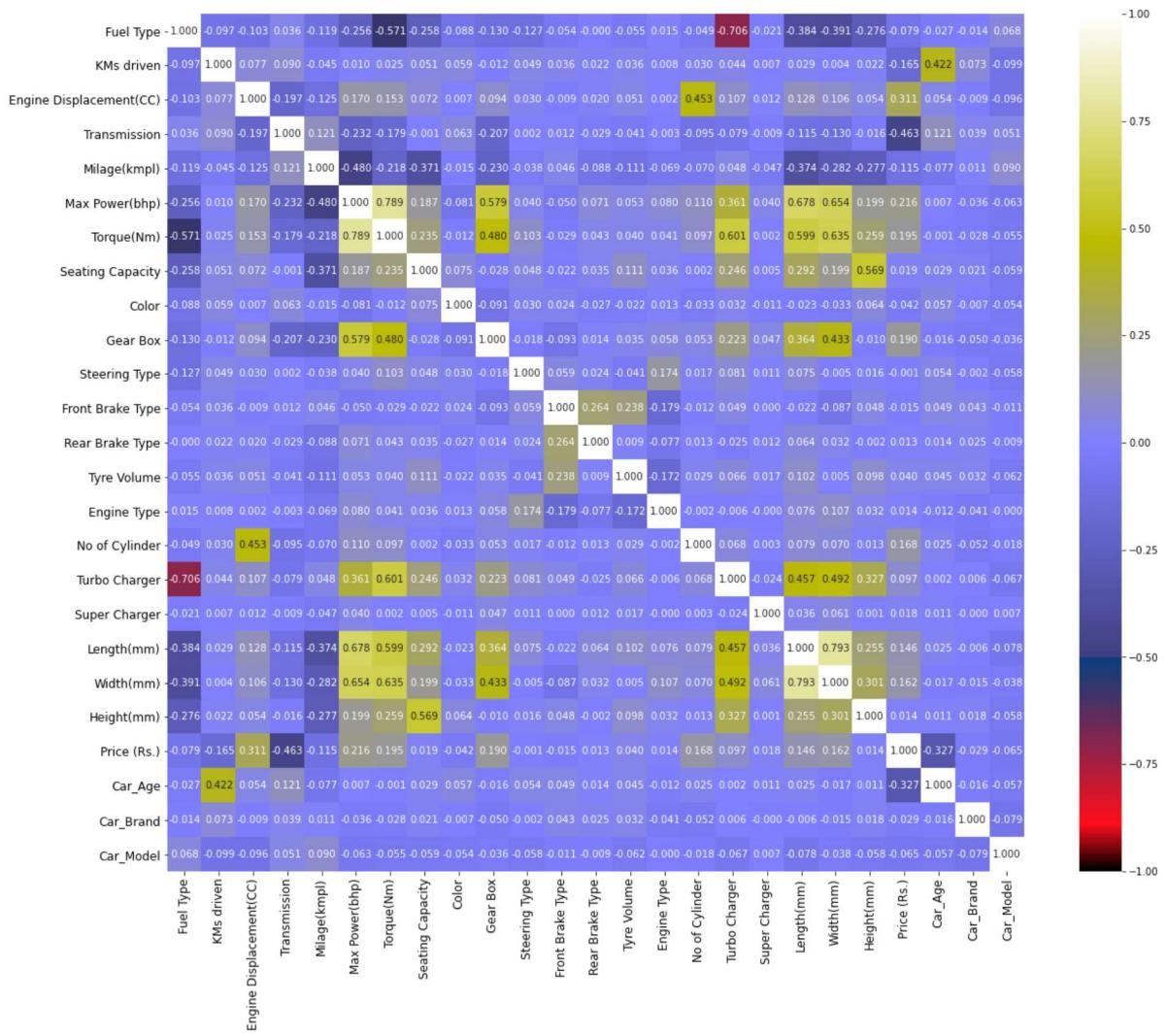
```
In [195... df.corr()
```

Out[195]:

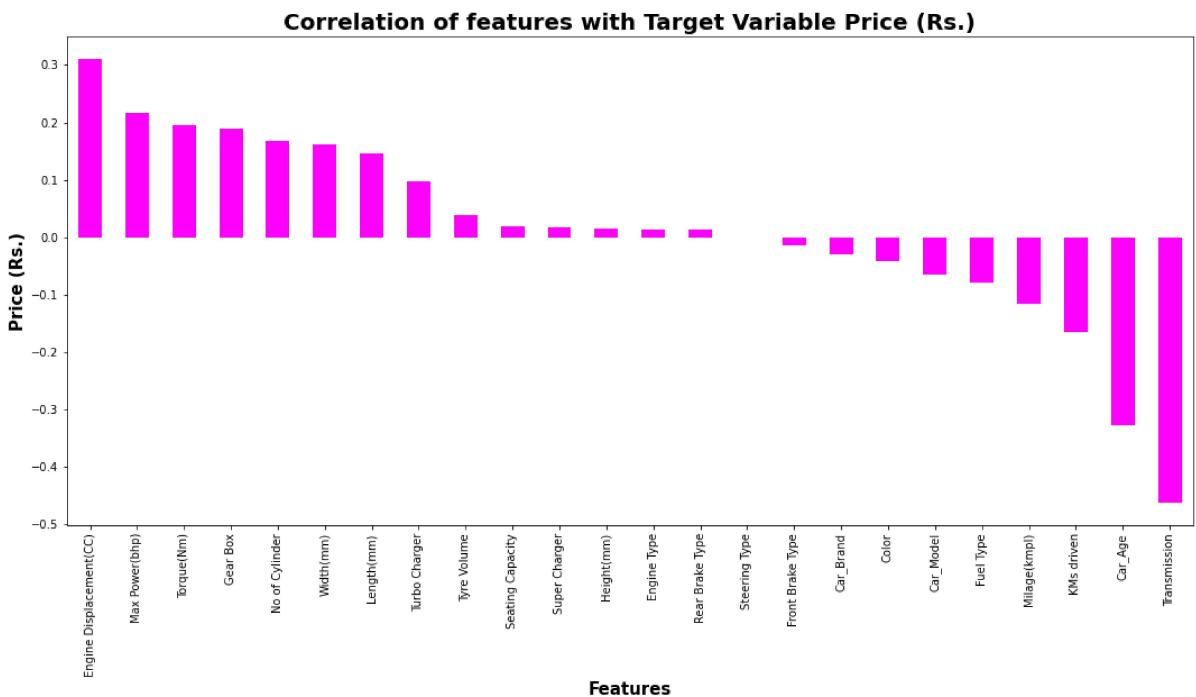
	Fuel Type	KMs driven	Engine Displacement(CC)	Transmission	Milage(kmpl)	Max Power(bhp)
Fuel Type	1.000000	-0.096669	-0.103263	0.036299	-0.119378	-0.255754
KMs driven	-0.096669	1.000000	0.077127	0.089567	-0.044611	0.009638
Engine Displacement(CC)	-0.103263	0.077127	1.000000	-0.197157	-0.125241	0.169724
Transmission	0.036299	0.089567	-0.197157	1.000000	0.121133	-0.231979
Milage(kmpl)	-0.119378	-0.044611	-0.125241	0.121133	1.000000	-0.479573
Max Power(bhp)	-0.255754	0.009638	0.169724	-0.231979	-0.479573	1.000000
Torque(Nm)	-0.570999	0.025004	0.153158	-0.178822	-0.218453	0.788804
Seating Capacity	-0.257936	0.050685	0.071639	-0.000538	-0.370555	0.186852
Color	-0.088287	0.059157	0.007224	0.063239	-0.014775	-0.081254
Gear Box	-0.130240	-0.011959	0.093947	-0.206584	-0.230308	0.579099
Steering Type	-0.127099	0.048709	0.030277	0.002166	-0.037738	0.040120
Front Brake Type	-0.053774	0.035885	-0.009279	0.011637	0.045736	-0.050114
Rear Brake Type	-0.000013	0.022178	0.019866	-0.028710	-0.088282	0.070916
Tyre Volume	-0.054525	0.035672	0.050597	-0.041162	-0.110698	0.052936
Engine Type	0.015317	0.008317	0.001771	-0.002874	-0.069008	0.080080
No of Cylinder	-0.049173	0.029526	0.453062	-0.094681	-0.069725	0.109990
Turbo Charger	-0.705556	0.044174	0.107254	-0.078553	0.047614	0.360693
Super Charger	-0.021321	0.007300	0.012034	-0.008924	-0.047316	0.039714
Length(mm)	-0.384157	0.029410	0.128330	-0.115205	-0.374392	0.678239
Width(mm)	-0.391022	0.003664	0.105600	-0.129999	-0.281511	0.654269
Height(mm)	-0.275566	0.022279	0.054442	-0.016392	-0.276793	0.199255
Price (Rs.)	-0.079260	-0.165029	0.310738	-0.462612	-0.114927	0.215779
Car_Age	-0.026821	0.421579	0.054418	0.120936	-0.077048	0.006523
Car_Brand	-0.013892	0.073370	-0.009376	0.039043	0.010856	-0.035675
Car_Model	0.067776	-0.099206	-0.095602	0.050768	0.089835	-0.062735

In [196...]

```
plt.figure(figsize=(20,16))
sns.heatmap(df.corr(),linecolor='black', vmin=-1, vmax=1, annot=True, square=True,
            annot_kws={'size':10}, cmap='gist_stern')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```



```
In [197...]: plt.figure(figsize = (18,8))
df.corr()['Price (Rs.)'].drop(['Price (Rs.)']).sort_values(ascending=False).plot(kind='bar')
plt.xlabel('Features', fontsize=15, fontweight='bold')
plt.ylabel('Price (Rs.)', fontsize=15, fontweight='bold')
plt.title('Correlation of features with Target Variable Price (Rs.)', fontsize = 20,
plt.show()
```



Observation :

- Maximum correlation observe in Length & Width followed by torque & max power.
- Most of features are moderately & poorly correlated with each other.
- Steering type very very poorly correlated with each other.

4. Checking Multicollinearity between features using variance_inflation_factor

In [200]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = pd.DataFrame()
vif["VIF values"] = [variance_inflation_factor(X_scale,i) for i in range(len(X.columns))]
vif["Features"] = X.columns
vif
```

Out[200]:

	VIF values	Features
0	2.595599	Fuel Type
1	1.254564	KMs driven
2	1.341150	Engine Displacement(CC)
3	1.130918	Transmission
4	1.917950	Milage(kmpl)
5	5.284063	Max Power(bhp)
6	4.834256	Torque(Nm)
7	1.782760	Seating Capacity
8	1.045537	Color
9	1.643369	Gear Box
10	1.098677	Steering Type
11	1.220471	Front Brake Type
12	1.108979	Rear Brake Type
13	1.144690	Tyre Volume
14	1.115098	Engine Type
15	1.270149	No of Cylinder
16	2.500721	Turbo Charger
17	1.017195	Super Charger
18	3.565548	Length(mm)
19	3.508860	Width(mm)
20	1.703933	Height(mm)
21	1.247364	Car_Age
22	1.024425	Car_Brand
23	1.045084	Car_Model

We can see that no multicollinearity exist between features.

In [209...]

```
### Importing Final data final for ML Learning after Feature Engineering
df.to_excel('Final ML Data.xlsx', index = False)
```

In []: