

## nmassignment2ipynb

```
[1]: import pandas as pd
import numpy as np
data=pd.read_csv("/content/car data.csv")
```

### 1.EDA

```
[3]: data.head()
```

```
[3]:   Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014         3.35         5.59       27000    Petrol
1    sx4    2013         4.75         9.54       43000    Diesel
2    ciaz    2017         7.25         9.85        6900    Petrol
3  wagon r    2011         2.85         4.15        5200    Petrol
4   swift    2014         4.60         6.87       42450    Diesel
```

```
   Seller_Type  Transmission  Owner
0    Dealer      Manual      0
1    Dealer      Manual      0
2    Dealer      Manual      0
3    Dealer      Manual      0
4    Dealer      Manual      0
```

```
[4]: data.tail()
```

```
[4]:   Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
296   city    2016         9.50         11.6       33988    Diesel
297   brio    2015         4.00         5.9       60000    Petrol
298   city    2009         3.35         11.0       87934    Petrol
299   city    2017        11.50         12.5        9000    Diesel
300   brio    2016         5.30         5.9        5464    Petrol
```

```
   Seller_Type  Transmission  Owner
296    Dealer      Manual      0
297    Dealer      Manual      0
298    Dealer      Manual      0
299    Dealer      Manual      0
300    Dealer      Manual      0
```

```
[5]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Car_Name        301 non-null    object
1   Year            301 non-null    int64
2   Selling_Price   301 non-null    float64
3   Present_Price   301 non-null    float64
4   Kms_Driven      301 non-null    int64
5   Fuel_Type       301 non-null    object
6   Seller_Type     301 non-null    object
7   Transmission    301 non-null    object
8   Owner           301 non-null    int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

```
[6]: data.describe()
```

```
[6]:
```

	Year	Selling_Price	Present_Price	Kms_Driven	Owner
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	2013.627907	4.661296	7.628472	36947.205980	0.043189
std	2.891554	5.082812	8.644115	38886.883882	0.247915
min	2003.000000	0.100000	0.320000	500.000000	0.000000
25%	2012.000000	0.900000	1.200000	15000.000000	0.000000
50%	2014.000000	3.600000	6.400000	32000.000000	0.000000
75%	2016.000000	6.000000	9.900000	48767.000000	0.000000
max	2018.000000	35.000000	92.600000	500000.000000	3.000000

```
[8]: data.shape
```

```
[8]: (301, 9)
```

```
[9]: data.columns
```

```
[9]: Index(['Car_Name', 'Year', 'Selling_Price', 'Present_Price', 'Kms_Driven',
        'Fuel_Type', 'Seller_Type', 'Transmission', 'Owner'],
        dtype='object')
```

```
[10]: data.isna().sum()
```

```
[10]: Car_Name      0
      Year        0
      Selling_Price  0
      Present_Price  0
```

```

Kms_Driven      0
Fuel_Type       0
Seller_Type     0
Transmission    0
Owner           0
dtype: int64

```

```
[11]: data.duplicated()
```

```

[11] : 0      False
      1      False
      2      False
      3      False
      4      False
      ...
     296     False
     297     False
     298     False
     299     False
     300     False
      Length: 301, dtype: bool

```

```
[12]: data.duplicated().sum()
```

```
[12]: 2
```

```
[15]: data.drop_duplicates()
```

```

[15]:
   Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type \
0      ritz  2014           3.35           5.59       27000    Petrol
1       sx4  2013           4.75           9.54       43000    Diesel
2      ciaz  2017           7.25           9.85        6900    Petrol
3  wagon r  2011           2.85           4.15        5200    Petrol
4     swift  2014           4.60           6.87       42450    Diesel
..      ...  ...           ...           ...       ...      ...
296   city  2016           9.50          11.60       33988    Diesel
297   brio  2015           4.00           5.90       60000    Petrol
298   city  2009           3.35          11.00       87934    Petrol
299   city  2017          11.50          12.50        9000    Diesel
300   brio  2016           5.30           5.90        5464    Petrol

      Seller_Type  Transmission  Owner
0         Dealer         Manual      0
1         Dealer         Manual      0
2         Dealer         Manual      0
3         Dealer         Manual      0
4         Dealer         Manual      0

```

```

..      ""      ""      ""
296      Dealer      Manual      0
297      Dealer      Manual      0
298      Dealer      Manual      0
299      Dealer      Manual      0
300      Dealer      Manual      0

```

```
[299 rows x 9 columns]
```

```
[17]: data.value_counts("Car_Name")
```

```

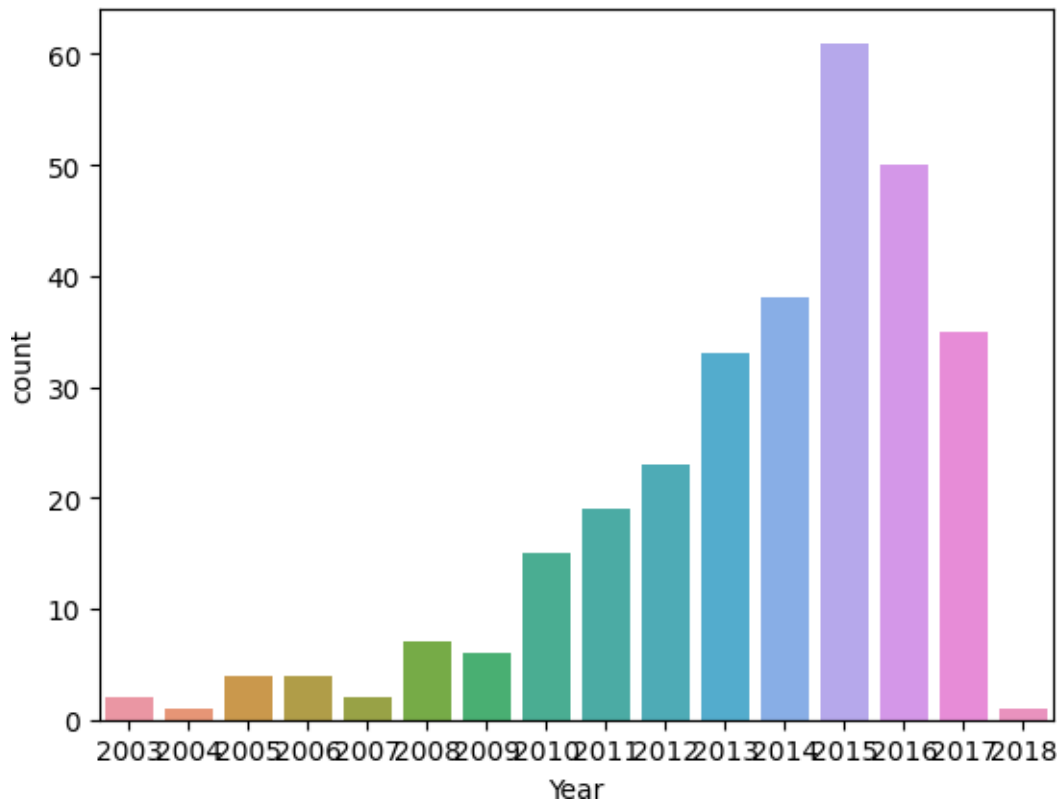
[17]: Car_Name
city                26
corolla altis       16
verna               14
fortuner            11
brio                10
..
Hero Passion X pro   1
Hero Hunk            1
camry               1
Hero Honda Passion Pro 1
800                 1
Length: 98, dtype: int64

```

```
[18]: import seaborn as sn
```

```
[19]: sn.countplot(x="Year", data=data)
```

```
[19]: <Axes: xlabel='Year', ylabel='count'>
```



## 2.PREPROCESSING

```
[20]: from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
z=data[["Kms_Driven"]]
x=scaler.fit_transform(z)
data[["Kms_Driven"]]=x
```

```
[24]: data['Year']=2023-data['Year']
t=data[["Year"]]
s=scaler.fit_transform(t)
data[["Year"]]=s
```

```
[22]: z=data[["Selling_Price"]]
x=scaler.fit_transform(z)
data[["Selling_Price"]]=x
```

```
[25]: data.drop('Car_Name',axis=1)
```

```
[25]:
```

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	\
0	0.733333	0.093123	5.59	0.053053	Petrol	Dealer	

1	0.666667	0.133238	9.54	0.085085	Diesel	Dealer
2	0.933333	0.204871	9.85	0.012813	Petrol	Dealer
3	0.533333	0.078797	4.15	0.009409	Petrol	Dealer
4	0.733333	0.128940	6.87	0.083984	Diesel	Dealer
..	...	...	...	...	...	...
296	0.866667	0.269341	11.60	0.067043	Diesel	Dealer
297	0.800000	0.111748	5.90	0.119119	Petrol	Dealer
298	0.400000	0.093123	11.00	0.175043	Petrol	Dealer
299	0.933333	0.326648	12.50	0.017017	Diesel	Dealer
300	0.866667	0.148997	5.90	0.009938	Petrol	Dealer

	Transmission	Owner
0	Manual	0
1	Manual	0
2	Manual	0
3	Manual	0
4	Manual	0
..	...	...
296	Manual	0
297	Manual	0
298	Manual	0
299	Manual	0
300	Manual	0

[301 rows x 8 columns]

### 3. TRAIN, TEST, SPLIT

```
[30]: from sklearn.model_selection import train_test_split
x=data.drop('Selling_Price',axis=1)
y=data["Selling_Price"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
↪4,random_state=20)
```

```
[31]: x_train.shape
```

```
[31]: (180, 8)
```

```
[32]: y_train.shape
```

```
[32]: (180,)
```

```
[33]: x_test.shape
```

```
[33]: (121, 8)
```