

## PREPROCESSING TRAINING DATA: CAR ECONOMY PRICE.CSV FILE

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

Out[1]:

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second
...	...	...	...	...	...	...	...	...
6014	6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First
6015	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First
6016	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second
6017	6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First
6018	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First

6019 rows × 14 columns

```
In [2]: df.shape
```

Out[2]: (6019, 14)

```
In [3]: df.size
```

Out[3]: 84266

```
In [4]: df.head()
```

```
Out[4]:
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	M
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	

```
In [5]: df.tail()
```

```
Out[5]:
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
6014	6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First
6015	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First
6016	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second
6017	6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First
6018	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First

```
In [6]: df.columns
```

```
Out[6]: Index(['Unnamed: 0', 'Name', 'Location', 'Year', 'Kilometers_Driven',  
              'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power',  
              'Seats', 'New_Price', 'Price'],  
             dtype='object')
```

```
In [7]: df.dtypes
```

```
Out[7]: Unnamed: 0      int64
        Name          object
        Location      object
        Year           int64
        Kilometers_Driven int64
        Fuel_Type      object
        Transmission   object
        Owner_Type     object
        Mileage         object
        Engine          object
        Power           object
        Seats          float64
        New_Price       object
        Price          float64
        dtype: object
```

```
In [8]: # MISSING VALUES:
        df.isna().sum()
```

```
Out[8]: Unnamed: 0      0
        Name          0
        Location      0
        Year           0
        Kilometers_Driven 0
        Fuel_Type      0
        Transmission   0
        Owner_Type     0
        Mileage        2
        Engine         36
        Power          36
        Seats         42
        New_Price     5195
        Price          0
        dtype: int64
```

```
In [9]: # COLUMN: NEW_PRICE HAS 5195/6019 MISSING VALUES, SO DROP IT
        # COLUMN: UNNAMED WILL HAVE NO EFFECT ON OUR REGRESSION MODEL TO PREDICT PRICE, SO DROP IT
```

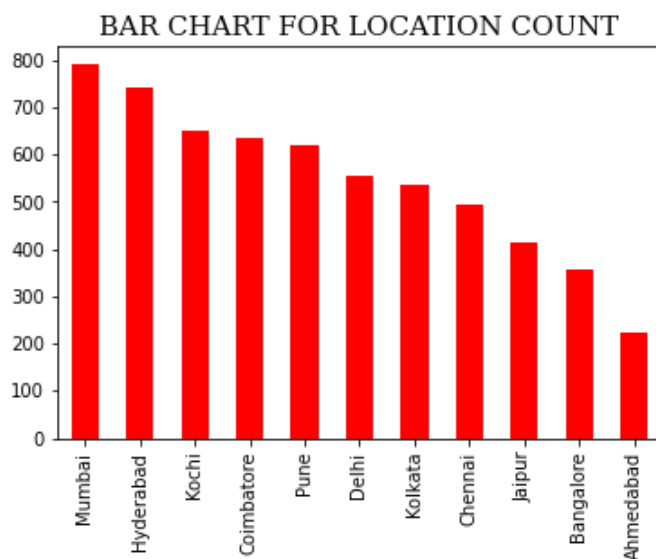
```
In [10]: # COUNT IN COLUMN: NAME
         df['Name'].value_counts()
         # DROP COLUMN NAME AS ALMOST 2000/6000 NAMES ARE DIFFERENT
```

```
Out[10]: Mahindra XUV500 W8 2WD      49
         Maruti Swift VDI             45
         Honda City 1.5 S MT          34
         Maruti Swift Dzire VDI        34
         Maruti Swift VDI BSIV         31
         ..
         Ford Fiesta Titanium 1.5 TDCi 1
         Mahindra Scorpio S10 AT 4WD   1
         Hyundai i20 1.2 Era           1
         Toyota Camry W4 (AT)          1
         Mahindra Xylo D4 BSIV         1
         Name: Name, Length: 1878, dtype: int64
```

```
In [11]: # COUNT IN COLUMN: LOCATION
df['Location'].value_counts()
```

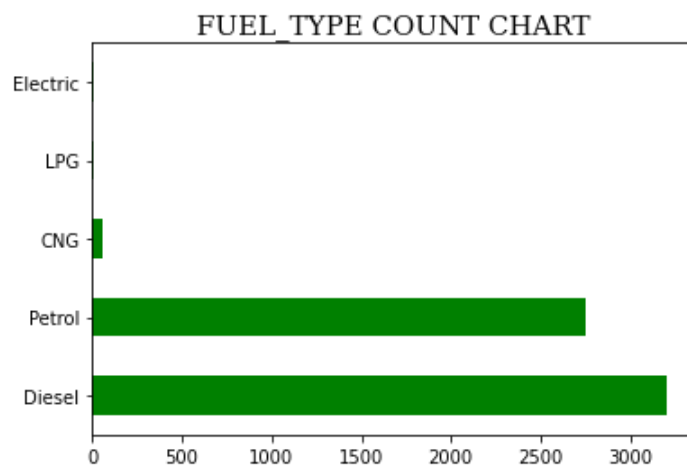
```
Out[11]: Mumbai          790
Hyderabad          742
Kochi              651
Coimbatore         636
Pune               622
Delhi             554
Kolkata           535
Chennai           494
Jaipur            413
Bangalore         358
Ahmedabad         224
Name: Location, dtype: int64
```

```
In [12]: # GRAPHICAL REPRESENTATION:
import matplotlib.pyplot as plt
df['Location'].value_counts().plot(kind='bar',color='red')
font={'family':'serif','color':'black','size':15}
plt.title('BAR CHART FOR LOCATION COUNT',fontdict=font)
plt.show()
```



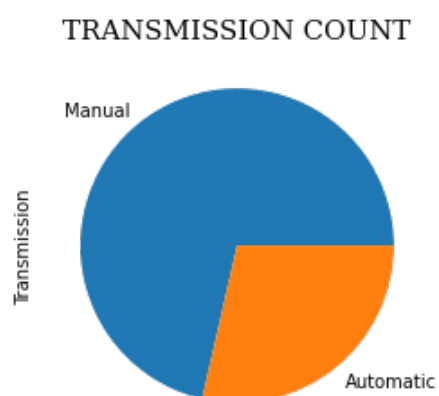
```
In [13]: # COUNT IN COLUMN: FUEL_TYPE
fuel=df['Fuel_Type'].value_counts()
print(fuel)
fuel.plot(kind='barh',color='green')
plt.title("FUEL_TYPE COUNT CHART",fontdict=font)
plt.show()
```

```
Diesel      3205
Petrol      2746
CNG         56
LPG         10
Electric     2
Name: Fuel_Type, dtype: int64
```



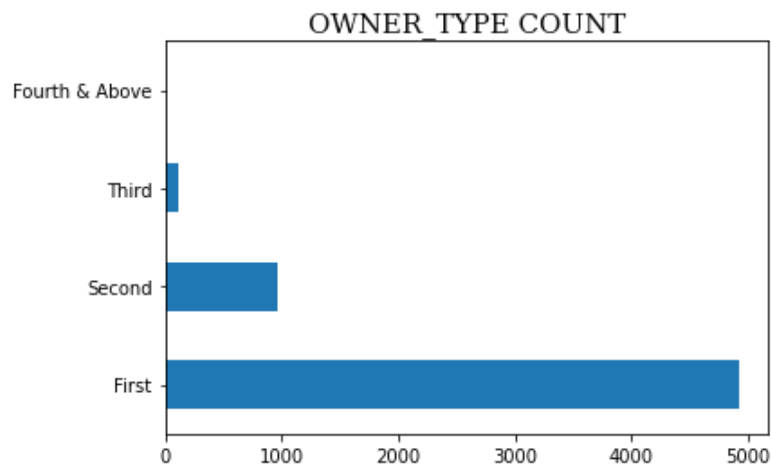
```
In [14]: # COUNT IN COLUMN: TRANSMISSION
trans=df['Transmission'].value_counts()
print(trans)
trans.plot(kind='pie')
plt.title('TRANSMISSION COUNT',fontdict=font)
plt.show()
```

```
Manual      4299
Automatic   1720
Name: Transmission, dtype: int64
```



```
In [15]: # COUNT IN COLUMN: OWNER_TYPE
own=df['Owner_Type'].value_counts()
print(own)
own.plot(kind='barh')
plt.title('OWNER_TYPE COUNT',fontdict=font)
plt.show()
```

```
First          4929
Second         968
Third          113
Fourth & Above    9
Name: Owner_Type, dtype: int64
```



```
In [16]: # CONVERT STRING TO NUMERICAL: GET DUMMY ENCODING
dummy=pd.get_dummies(df[['Location','Fuel_Type','Transmission','Owner_Type']],drop_first=True)
dummy
```

Out[16]:

	Location_Bangalore	Location_Chennai	Location_Coimbatore	Location_Delhi	Location_Hyderabad
0	0	0	0	0	0
1	0	0	0	0	0
2	0	1	0	0	0
3	0	1	0	0	0
4	0	0	1	0	0
...	...	...	...	...	...
6014	0	0	0	1	0
6015	0	0	0	0	0
6016	0	0	0	0	0
6017	0	0	0	0	0
6018	0	0	0	0	1

6019 rows × 18 columns

```
In [17]: # CONCAT: TO COMBINE THE OLD DF AND DUUMY TO GET ALL THE REQUIRED COLUMNS
dfe=pd.concat([df,dummy],axis=1)
dfe
```

Out[17]:

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second
...	...	...	...	...	...	...	...	...
6014	6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First
6015	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First
6016	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second
6017	6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First
6018	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First

6019 rows × 32 columns

```
In [18]: # NOW DROP UNWANTED COLUMNS
dfe=dfe.drop(['Unnamed: 0', 'Name', 'New_Price', 'Location', 'Fuel_Type', 'Transmission', 'Owner_Type'],axis=1)
dfe
```

Out[18]:

	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Price	Location_Bangalore	Location_Ch
0	2010	72000	26.6 km/kg	998 CC	58.16 bhp	5.0	1.75	0	
1	2015	41000	19.67 kmpl	1582 CC	126.2 bhp	5.0	12.50	0	
2	2011	46000	18.2 kmpl	1199 CC	88.7 bhp	5.0	4.50	0	
3	2012	87000	20.77 kmpl	1248 CC	88.76 bhp	7.0	6.00	0	
4	2013	40670	15.2 kmpl	1968 CC	140.8 bhp	5.0	17.74	0	
...	...	...	...	...	...	...	...	...	
6014	2014	27365	28.4 kmpl	1248 CC	74 bhp	5.0	4.75	0	
6015	2015	100000	24.4 kmpl	1120 CC	71 bhp	5.0	4.00	0	
6016	2012	55000	14.0 kmpl	2498 CC	112 bhp	8.0	2.90	0	
6017	2013	46000	18.9 kmpl	998 CC	67.1 bhp	5.0	2.65	0	
6018	2011	47000	25.44 kmpl	936 CC	57.6 bhp	5.0	2.50	0	

6019 rows × 25 columns

```
In [19]: dfe.dtypes
```

```
Out[19]: Year                int64
Kilometers_Driven          int64
Mileage                    object
Engine                    object
Power                    object
Seats                    float64
Price                    float64
Location_Bangalore         uint8
Location_Chennai           uint8
Location_Coimbatore        uint8
Location_Delhi             uint8
Location_Hyderabad         uint8
Location_Jaipur            uint8
Location_Kochi             uint8
Location_Kolkata           uint8
Location_Mumbai            uint8
Location_Pune              uint8
Fuel_Type_Diesel           uint8
Fuel_Type_Electric         uint8
Fuel_Type_LPG              uint8
Fuel_Type_Petrol           uint8
Transmission_Manual        uint8
Owner_Type_Fourth & Above  uint8
Owner_Type_Second          uint8
Owner_Type_Third           uint8
dtype: object
```



```
In [20]: # REMOVE STRINGS IN EACH COLUMN
# 1.MILEAGE
dfe['Mileage']=dfe['Mileage'].str.replace('km/kg','')
dfe['Mileage']=dfe['Mileage'].str.replace('kmpl','')
dfe['Engine']=dfe['Engine'].str.replace('CC','')
dfe['Power']=dfe['Power'].str.replace('bhp','')
# REPLACE STRING NULL WITH 0
dfe['Mileage']=dfe['Mileage'].str.replace('null','0')
dfe['Power']=dfe['Power'].str.replace('null','0')
dfe['Engine']=dfe['Engine'].str.replace('null','0')
dfe
```

```
Out[20]:
```

	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Price	Location_Bangalore	Location_Ch
0	2010	72000	26.6	998	58.16	5.0	1.75	0	
1	2015	41000	19.67	1582	126.2	5.0	12.50	0	
2	2011	46000	18.2	1199	88.7	5.0	4.50	0	
3	2012	87000	20.77	1248	88.76	7.0	6.00	0	
4	2013	40670	15.2	1968	140.8	5.0	17.74	0	
...	...	...	...	...	...	...	...	...	...
6014	2014	27365	28.4	1248	74	5.0	4.75	0	
6015	2015	100000	24.4	1120	71	5.0	4.00	0	
6016	2012	55000	14.0	2498	112	8.0	2.90	0	
6017	2013	46000	18.9	998	67.1	5.0	2.65	0	
6018	2011	47000	25.44	936	57.6	5.0	2.50	0	

6019 rows × 25 columns

```
In [21]: dfe.isna().sum()
```

```
Out[21]: Year                                0
Kilometers_Driven                          0
Mileage                                    2
Engine                                    36
Power                                    36
Seats                                    42
Price                                      0
Location_Bangalore                        0
Location_Chennai                        0
Location_Coimbatore                      0
Location_Delhi                          0
Location_Hyderabad                      0
Location_Jaipur                         0
Location_Kochi                         0
Location_Kolkata                        0
Location_Mumbai                        0
Location_Pune                          0
Fuel_Type_Diesel                        0
Fuel_Type_Electric                     0
Fuel_Type_LPG                          0
Fuel_Type_Petrol                       0
Transmission_Manual                    0
Owner_Type_Fourth & Above                0
Owner_Type_Second                      0
Owner_Type_Third                       0
dtype: int64
```

```
In [22]: dfe.dtypes
```

```
Out[22]: Year                                int64
Kilometers_Driven                          int64
Mileage                                    object
Engine                                    object
Power                                    object
Seats                                    float64
Price                                    float64
Location_Bangalore                        uint8
Location_Chennai                        uint8
Location_Coimbatore                      uint8
Location_Delhi                          uint8
Location_Hyderabad                      uint8
Location_Jaipur                         uint8
Location_Kochi                         uint8
Location_Kolkata                       uint8
Location_Mumbai                        uint8
Location_Pune                          uint8
Fuel_Type_Diesel                        uint8
Fuel_Type_Electric                     uint8
Fuel_Type_LPG                          uint8
Fuel_Type_Petrol                       uint8
Transmission_Manual                    uint8
Owner_Type_Fourth & Above               uint8
Owner_Type_Second                      uint8
Owner_Type_Third                      uint8
dtype: object
```

```
In [23]: # CONVERT OBJECT DTYPE TO FLOAT DTYPE
dfe['Mileage']=dfe['Mileage'].astype(float)
dfe['Power']=dfe['Power'].astype(float)
dfe['Engine']=dfe['Engine'].astype(float)
dfe.dtypes
```

```
Out[23]: Year                                int64
Kilometers_Driven                          int64
Mileage                                    float64
Engine                                    float64
Power                                    float64
Seats                                    float64
Price                                    float64
Location_Bangalore                        uint8
Location_Chennai                        uint8
Location_Coimbatore                      uint8
Location_Delhi                          uint8
Location_Hyderabad                      uint8
Location_Jaipur                         uint8
Location_Kochi                         uint8
Location_Kolkata                       uint8
Location_Mumbai                        uint8
Location_Pune                          uint8
Fuel_Type_Diesel                        uint8
Fuel_Type_Electric                     uint8
Fuel_Type_LPG                          uint8
Fuel_Type_Petrol                       uint8
Transmission_Manual                    uint8
Owner_Type_Fourth & Above               uint8
Owner_Type_Second                      uint8
Owner_Type_Third                      uint8
dtype: object
```

```
In [24]: # REPLACE 0 IN ENGINE POWER AND MILEAGE TO NOT A NUMBER(NaN)
dfe.loc[dfe.Engine==0, 'Engine']=np.NaN
dfe.loc[dfe.Power==0, 'Power']=np.NaN
dfe.loc[dfe.Mileage==0, 'Mileage']=np.NaN
```

```
In [25]: dfe.isna().sum()
```

```
Out[25]: Year                                0
Kilometers_Driven                          0
Mileage                                    70
Engine                                     36
Power                                    143
Seats                                     42
Price                                      0
Location_Bangalore                        0
Location_Chennai                         0
Location_Coimbatore                      0
Location_Delhi                           0
Location_Hyderabad                       0
Location_Jaipur                          0
Location_Kochi                           0
Location_Kolkata                         0
Location_Mumbai                          0
Location_Pune                             0
Fuel_Type_Diesel                         0
Fuel_Type_Electric                       0
Fuel_Type_LPG                           0
Fuel_Type_Petrol                         0
Transmission_Manual                     0
Owner_Type_Fourth & Above                 0
Owner_Type_Second                        0
Owner_Type_Third                         0
dtype: int64
```

```
In [26]: # REPLACE MISSING VALUE
dfe['Engine']=dfe['Engine'].fillna(dfe['Engine'].mean())
dfe['Mileage']=dfe['Mileage'].fillna(dfe['Mileage'].mean())
dfe['Power']=dfe['Power'].fillna(dfe['Power'].mean())
dfe['Seats']=dfe['Seats'].fillna(dfe['Seats'].mode()[0])
```

```
In [27]: dfe.isna().sum()
```

```
Out[27]: Year                                0
Kilometers_Driven                          0
Mileage                                     0
Engine                                      0
Power                                       0
Seats                                       0
Price                                       0
Location_Bangalore                        0
Location_Chennai                         0
Location_Coimbatore                      0
Location_Delhi                           0
Location_Hyderabad                       0
Location_Jaipur                          0
Location_Kochi                           0
Location_Kolkata                         0
Location_Mumbai                          0
Location_Pune                            0
Fuel_Type_Diesel                         0
Fuel_Type_Electric                       0
Fuel_Type_LPG                           0
Fuel_Type_Petrol                         0
Transmission_Manual                     0
Owner_Type_Fourth & Above                0
Owner_Type_Second                        0
Owner_Type_Third                         0
dtype: int64
```

```
In [28]: # SEPERATING X AND Y FROM THE PURE DATA
x=dfe.drop(['Price'],axis=1)
x
```

```
Out[28]:
```

	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Location_Bangalore	Location_Chennai
0	2010	72000	26.60	998.0	58.16	5.0	0	0
1	2015	41000	19.67	1582.0	126.20	5.0	0	0
2	2011	46000	18.20	1199.0	88.70	5.0	0	1
3	2012	87000	20.77	1248.0	88.76	7.0	0	1
4	2013	40670	15.20	1968.0	140.80	5.0	0	0
...	...	...	...	...	...	...	...	...
6014	2014	27365	28.40	1248.0	74.00	5.0	0	0
6015	2015	100000	24.40	1120.0	71.00	5.0	0	0
6016	2012	55000	14.00	2498.0	112.00	8.0	0	0
6017	2013	46000	18.90	998.0	67.10	5.0	0	0
6018	2011	47000	25.44	936.0	57.60	5.0	0	0

6019 rows × 24 columns

```
In [29]: y=dfe['Price']  
y
```

```
Out[29]: 0          1.75  
         1         12.50  
         2          4.50  
         3          6.00  
         4         17.74  
         ...  
        6014         4.75  
        6015         4.00  
        6016         2.90  
        6017         2.65  
        6018         2.50  
        Name: Price, Length: 6019, dtype: float64
```

## PREPROCESSING ON TESTING DATA: TEST-DATA.CSV FILE

```
In [30]: dft=pd.read_csv("/content/test-data.csv")
dft
```

Out[30]:

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	Fir
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Secor
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	Fir
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	Fir
4	4	Hyundai i20 Magna	Mumbai	2014	29000	Petrol	Manual	Fir
...	...	...	...	...	...	...	...	...
1229	1229	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	Fir
1230	1230	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	Fir
1231	1231	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	Fir
1232	1232	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Thi
1233	1233	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	Fir

1234 rows x 13 columns

```
In [31]: dft.shape
```

Out[31]: (1234, 13)

```
In [32]: dft.size
```

Out[32]: 16042

```
In [33]: dft.head()
```

Out[33]:

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	First	15.5
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Second	14.5
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	First	15.5
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	First	14.5
4	4	Hyundai i20 Magna	Mumbai	2014	29000	Petrol	Manual	First	15.5

```
In [34]: dft.tail()
```

Out[34]:

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage
1229	1229	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	14.5
1230	1230	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	15.5
1231	1231	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	14.5
1232	1232	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	15.5
1233	1233	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avantgarde	Kochi	2014	72443	Diesel	Automatic	First	15.5

```
In [35]: dft.columns
```

Out[35]: Index(['Unnamed: 0', 'Name', 'Location', 'Year', 'Kilometers\_Driven', 'Fuel\_Type', 'Transmission', 'Owner\_Type', 'Mileage', 'Engine', 'Power', 'Seats', 'New\_Price'], dtype='object')

```
In [36]: dft.dtypes
```

```
Out[36]: Unnamed: 0      int64
         Name          object
         Location      object
         Year           int64
         Kilometers_Driven int64
         Fuel_Type      object
         Transmission   object
         Owner_Type     object
         Mileage        object
         Engine         object
         Power          object
         Seats          float64
         New_Price      object
         dtype: object
```

```
In [37]: dft.isna().sum()
```

```
Out[37]: Unnamed: 0      0
         Name          0
         Location      0
         Year           0
         Kilometers_Driven int64
         Fuel_Type      0
         Transmission   0
         Owner_Type     0
         Mileage        0
         Engine         10
         Power          10
         Seats          11
         New_Price      1052
         dtype: int64
```

```
In [38]: # COUNT AND VISUALIZATION IN EACH STRING COLUMN
         name=dft['Name'].value_counts()
         print(name)
```

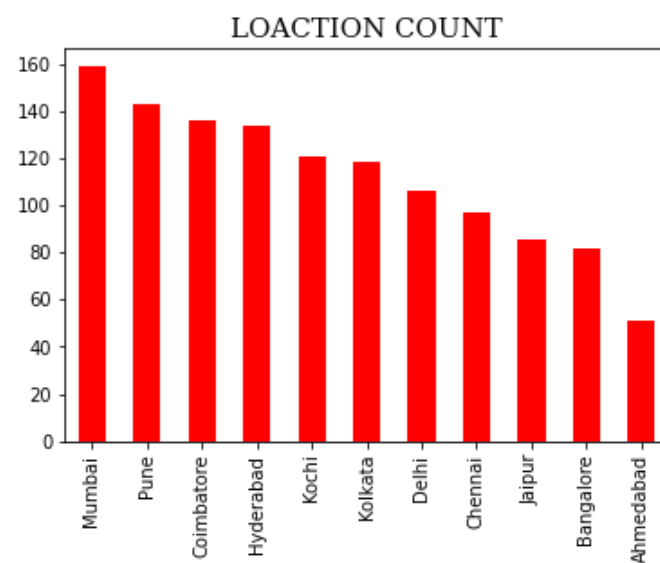
```
Maruti Alto LXi          9
Honda City 1.5 V MT      8
Maruti Swift Dzire VDI   8
Volkswagen Polo 1.2 MPI Highline 8
Hyundai i10 Magna        7
..
Hyundai Santro GLS I - Euro II 1
Honda City i DTec VX Option BL 1
Land Rover Discovery 4 SDV6 SE 1
Hyundai Verna CRDi 1.6 SX Option 1
Mercedes-Benz E-Class 2009-2013 E 220 CDI Avantgarde 1
Name: Name, Length: 769, dtype: int64
```



```
In [39]: # COLUMN:LOCATION
loc=df['Location'].value_counts()
print(loc)
loc.plot(kind='bar',color='red')
plt.title('LOACTION COUNT',fontdict=font)
plt.show()
```

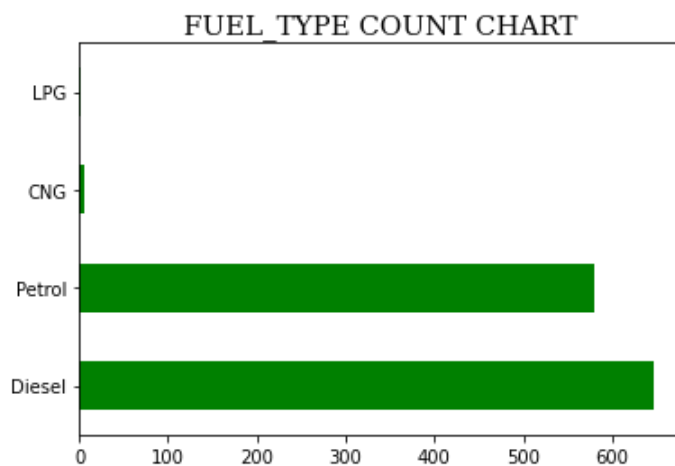
Mumbai	159
Pune	143
Coimbatore	136
Hyderabad	134
Kochi	121
Kolkata	119
Delhi	106
Chennai	97
Jaipur	86
Bangalore	82
Ahmedabad	51

Name: Location, dtype: int64



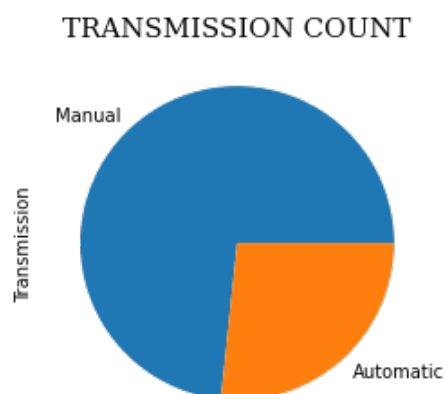
```
In [40]: # COUNT IN COLUMN: FUEL_TYPE
fuel1=df['Fuel_Type'].value_counts()
print(fuel1)
fuel1.plot(kind='barh',color='green')
plt.title("FUEL_TYPE COUNT CHART",fontdict=font)
plt.show()
```

```
Diesel      3205
Petrol      2746
CNG         56
LPG         10
Electric     2
Name: Fuel_Type, dtype: int64
```



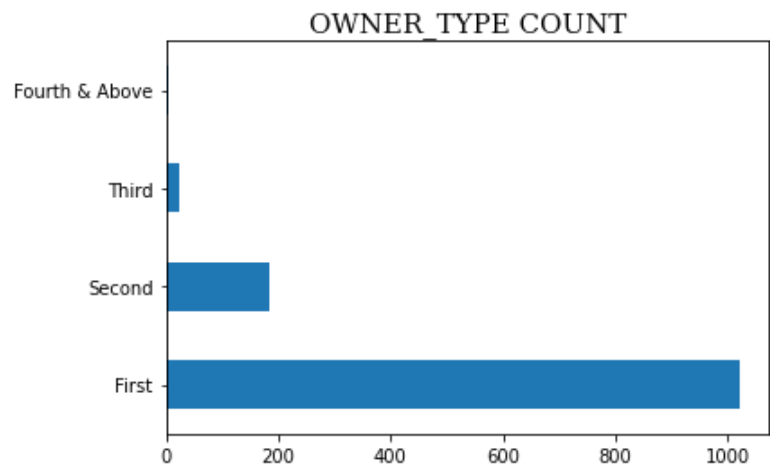
```
In [41]: # COUNT IN COLUMN: TRANSMISSION
trans1=df['Transmission'].value_counts()
print(trans1)
trans1.plot(kind='pie')
plt.title('TRANSMISSION COUNT',fontdict=font)
plt.show()
```

```
Manual      4299
Automatic   1720
Name: Transmission, dtype: int64
```



```
In [42]: # COUNT IN COLUMN: OWNER_TYPE
own1=dft['Owner_Type'].value_counts()
print(own)
own1.plot(kind='barh')
plt.title('OWNER_TYPE COUNT',fontdict=font)
plt.show()
```

First 4929  
Second 968  
Third 113  
Fourth & Above 9  
Name: Owner\_Type, dtype: int64



```
In [43]: # GET DUMMY ENCODING
dummy1=pd.get_dummies(dft[['Location','Fuel_Type','Transmission','Owner_Type']],drop_first=True)
dummy1
```

Out[43]:

	Location_Bangalore	Location_Chennai	Location_Coimbatore	Location_Delhi	Location_Hyderabad
0	0	0	0	1	0
1	0	0	1	0	0
2	0	0	0	0	0
3	0	0	0	0	1
4	0	0	0	0	0
...	...	...	...	...	...
1229	0	0	0	0	1
1230	0	0	0	0	0
1231	0	0	0	0	0
1232	0	0	0	0	0
1233	0	0	0	0	0

1234 rows x 17 columns

```
In [44]: # CONCATENATE
dfe2=pd.concat([dft,dummy1],axis=1)
dfe2
```

Out[44]:

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Typ
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	Fir
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Secor
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	Fir
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	Fir
4	4	Hyundai i20 Magna	Mumbai	2014	29000	Petrol	Manual	Fir
...	...	...	...	...	...	...	...	...
1229	1229	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	Fir
1230	1230	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	Fir
1231	1231	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	Fir
1232	1232	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Thii
1233	1233	Mercedes- Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	Fir

1234 rows x 30 columns

```
In [45]: # DROP UNWANTED COLUMNS
dfe2=dfe2.drop(['Unnamed: 0', 'Name', 'New_Price', 'Location', 'Fuel_Type', 'Transmission', 'Owner_Type'],axis=1)
dfe2
```

Out[45]:

	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Location_Bangalore	Location_Chennai
0	2014	40929	32.26 km/kg	998 CC	58.2 bhp	4.0	0	0
1	2013	54493	24.7 kmpl	796 CC	47.3 bhp	5.0	0	0
2	2017	34000	13.68 kmpl	2393 CC	147.8 bhp	7.0	0	0
3	2012	139000	23.59 kmpl	1364 CC	null bhp	5.0	0	0
4	2014	29000	18.5 kmpl	1197 CC	82.85 bhp	5.0	0	0
...	...	...	...	...	...	...	...	...
1229	2011	89411	20.54 kmpl	1598 CC	103.6 bhp	5.0	0	0
1230	2015	59000	17.21 kmpl	1197 CC	103.6 bhp	5.0	0	0
1231	2012	28000	23.08 kmpl	1461 CC	63.1 bhp	5.0	0	0
1232	2013	52262	17.2 kmpl	1197 CC	103.6 bhp	5.0	0	0
1233	2014	72443	10.0 kmpl	2148 CC	170 bhp	5.0	0	0

1234 rows × 23 columns

```
In [46]: dfe2.dtypes
```

```
Out[46]: Year                int64
Kilometers_Driven          int64
Mileage                    object
Engine                     object
Power                      object
Seats                     float64
Location_Bangalore         uint8
Location_Chennai           uint8
Location_Coimbatore        uint8
Location_Delhi             uint8
Location_Hyderabad         uint8
Location_Jaipur            uint8
Location_Kochi             uint8
Location_Kolkata           uint8
Location_Mumbai            uint8
Location_Pune              uint8
Fuel_Type_Diesel           uint8
Fuel_Type_LPG              uint8
Fuel_Type_Petrol           uint8
Transmission_Manual        uint8
Owner_Type_Fourth & Above  uint8
Owner_Type_Second          uint8
Owner_Type_Third           uint8
dtype: object
```

```
In [47]: # REPLACE COLUMNS WITH DATATYPE OBJECT WITH FLOAT
dfe2['Mileage']=dfe2['Mileage'].str.replace('km/kg','')
dfe2['Mileage']=dfe2['Mileage'].str.replace('kmpl','')
dfe2['Power']=dfe2['Power'].str.replace('bhp','')
dfe2['Engine']=dfe2['Engine'].str.replace('CC','')
dfe2['Mileage']=dfe2['Mileage'].str.replace('null','0')
dfe2['Power']=dfe2['Power'].str.replace('null','0')
dfe2['Engine']=dfe2['Engine'].str.replace('null','0')
```

```
In [48]: dfe2['Mileage']=dfe2['Mileage'].astype(float)
dfe2['Power']=dfe2['Power'].astype(float)
dfe2['Engine']=dfe2['Engine'].astype(float)
```

```
In [49]: dfe2.dtypes
```

```
Out[49]: Year                int64
Kilometers_Driven          int64
Mileage                    float64
Engine                    float64
Power                     float64
Seats                     float64
Location_Bangalore         uint8
Location_Chennai           uint8
Location_Coimbatore        uint8
Location_Delhi             uint8
Location_Hyderabad         uint8
Location_Jaipur            uint8
Location_Kochi             uint8
Location_Kolkata           uint8
Location_Mumbai            uint8
Location_Pune              uint8
Fuel_Type_Diesel           uint8
Fuel_Type_LPG              uint8
Fuel_Type_Petrol           uint8
Transmission_Manual        uint8
Owner_Type_Fourth & Above  uint8
Owner_Type_Second          uint8
Owner_Type_Third           uint8
dtype: object
```

```
In [50]: # REPLACE 0 WITH NAN
dfe2.loc[dfe2.Mileage==0, 'Engine']=np.NaN
dfe2.loc[dfe2.Power==0, 'Power']=np.NaN
dfe2.loc[dfe2.Engine==0, 'Engine']=np.NaN
```

```
In [51]: dfe2.isna().sum()
```

```
Out[51]: Year                                0
Kilometers_Driven                          0
Mileage                                    0
Engine                                    22
Power                                    32
Seats                                    11
Location_Bangalore                        0
Location_Chennai                        0
Location_Coimbatore                      0
Location_Delhi                          0
Location_Hyderabad                      0
Location_Jaipur                          0
Location_Kochi                          0
Location_Kolkata                        0
Location_Mumbai                         0
Location_Pune                           0
Fuel_Type_Diesel                        0
Fuel_Type_LPG                          0
Fuel_Type_Petrol                       0
Transmission_Manual                    0
Owner_Type_Fourth & Above               0
Owner_Type_Second                      0
Owner_Type_Third                       0
dtype: int64
```

```
In [52]: # REPLACE MISSING VALUE WITH MEAN AND MODE
dfe2['Power']=dfe2['Power'].fillna(dfe2['Power'].mean())
dfe2['Engine']=dfe2['Engine'].fillna(dfe2['Engine'].mean())
dfe2['Seats']=dfe2['Seats'].fillna(dfe2['Seats'].mode()[0])
```

```
In [53]: dfe2.isna().sum()
```

```
Out[53]: Year                                0
Kilometers_Driven                          0
Mileage                                    0
Engine                                    0
Power                                    0
Seats                                    0
Location_Bangalore                        0
Location_Chennai                        0
Location_Coimbatore                      0
Location_Delhi                          0
Location_Hyderabad                      0
Location_Jaipur                          0
Location_Kochi                          0
Location_Kolkata                        0
Location_Mumbai                         0
Location_Pune                           0
Fuel_Type_Diesel                        0
Fuel_Type_LPG                          0
Fuel_Type_Petrol                       0
Transmission_Manual                    0
Owner_Type_Fourth & Above               0
Owner_Type_Second                      0
Owner_Type_Third                       0
dtype: int64
```

```
In [54]: # COLUMNS IN X_TRAIN
x.columns
```

```
Out[54]: Index(['Year', 'Kilometers_Driven', 'Mileage', 'Engine', 'Power', 'Seats',
               'Location_Bangalore', 'Location_Chennai', 'Location_Coimbatore',
               'Location_Delhi', 'Location_Hyderabad', 'Location_Jaipur',
               'Location_Kochi', 'Location_Kolkata', 'Location_Mumbai',
               'Location_Pune', 'Fuel_Type_Diesel', 'Fuel_Type_Electric',
               'Fuel_Type_LPG', 'Fuel_Type_Petrol', 'Transmission_Manual',
               'Owner_Type_Fourth & Above', 'Owner_Type_Second', 'Owner_Type_Third'
               ],
              dtype='object')
```

```
In [58]: # COLUMNS IN X_TEST
dfe2.columns
```

```
Out[58]: Index(['Year', 'Kilometers_Driven', 'Mileage', 'Engine', 'Power', 'Seats',
               'Location_Bangalore', 'Location_Chennai', 'Location_Coimbatore',
               'Location_Delhi', 'Location_Hyderabad', 'Location_Jaipur',
               'Location_Kochi', 'Location_Kolkata', 'Location_Mumbai',
               'Location_Pune', 'Fuel_Type_Diesel', 'Fuel_Type_LPG',
               'Fuel_Type_Petrol', 'Transmission_Manual', 'Owner_Type_Fourth & Above',
               'Owner_Type_Second', 'Owner_Type_Third'],
              dtype='object')
```

```
In [56]: # DROP EXTRA COLUMN: 'Fuel_Type_Electric' FORM X_TRAIN:X DATAFRAME
x=x.drop(['Fuel_Type_Electric'],axis=1)
x.columns
```

```
Out[56]: Index(['Year', 'Kilometers_Driven', 'Mileage', 'Engine', 'Power', 'Seats',
               'Location_Bangalore', 'Location_Chennai', 'Location_Coimbatore',
               'Location_Delhi', 'Location_Hyderabad', 'Location_Jaipur',
               'Location_Kochi', 'Location_Kolkata', 'Location_Mumbai',
               'Location_Pune', 'Fuel_Type_Diesel', 'Fuel_Type_LPG',
               'Fuel_Type_Petrol', 'Transmission_Manual', 'Owner_Type_Fourth & Above',
               'Owner_Type_Second', 'Owner_Type_Third'],
              dtype='object')
```

```
In [59]: # MODEL
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x,y)
output_pred=model.predict(dfe2)
output_pred
```

```
Out[59]: array([ 2.87588492, -1.29344912, 16.1069494 , ...,  0.1378514 ,
                  9.27293255, 21.48043251])
```