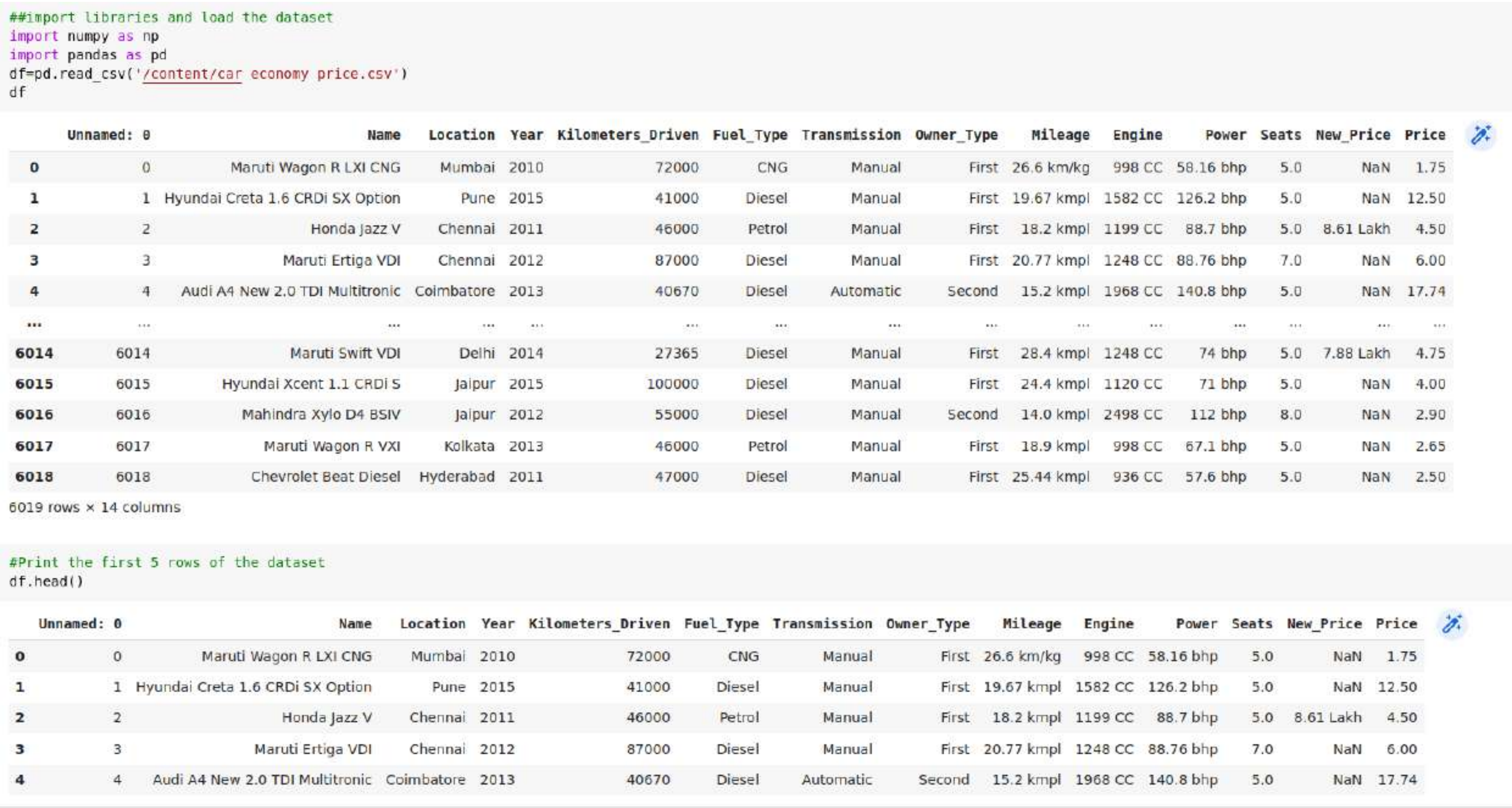


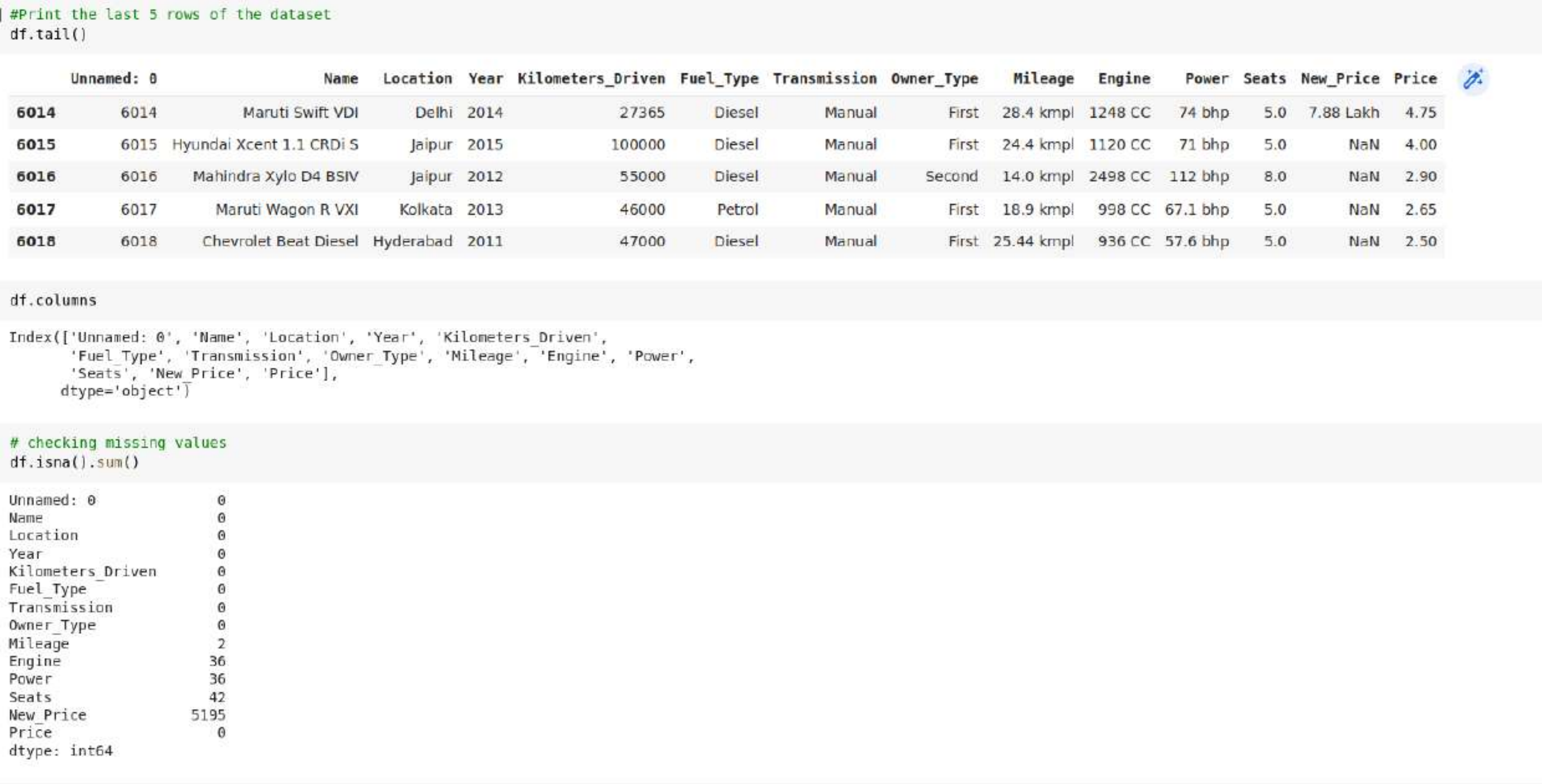
```
##import libraries and load the dataset
import numpy as np
import pandas as pd
df=pd.read_csv('/content/car economy price.csv')
df
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6014	6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.4 kmpl	1248 CC	74 bhp	5.0	7.88 Lakh	4.75
6015	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.4 kmpl	1120 CC	71 bhp	5.0	NaN	4.00
6016	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.0 kmpl	2498 CC	112 bhp	8.0	NaN	2.90
6017	6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First	18.9 kmpl	998 CC	67.1 bhp	5.0	NaN	2.65
6018	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44 kmpl	936 CC	57.6 bhp	5.0	NaN	2.50

6019 rows × 14 columns

```
#Print the first 5 rows of the dataset
df.head()
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74



```
#Print the last 5 rows of the dataset
df.tail()
```

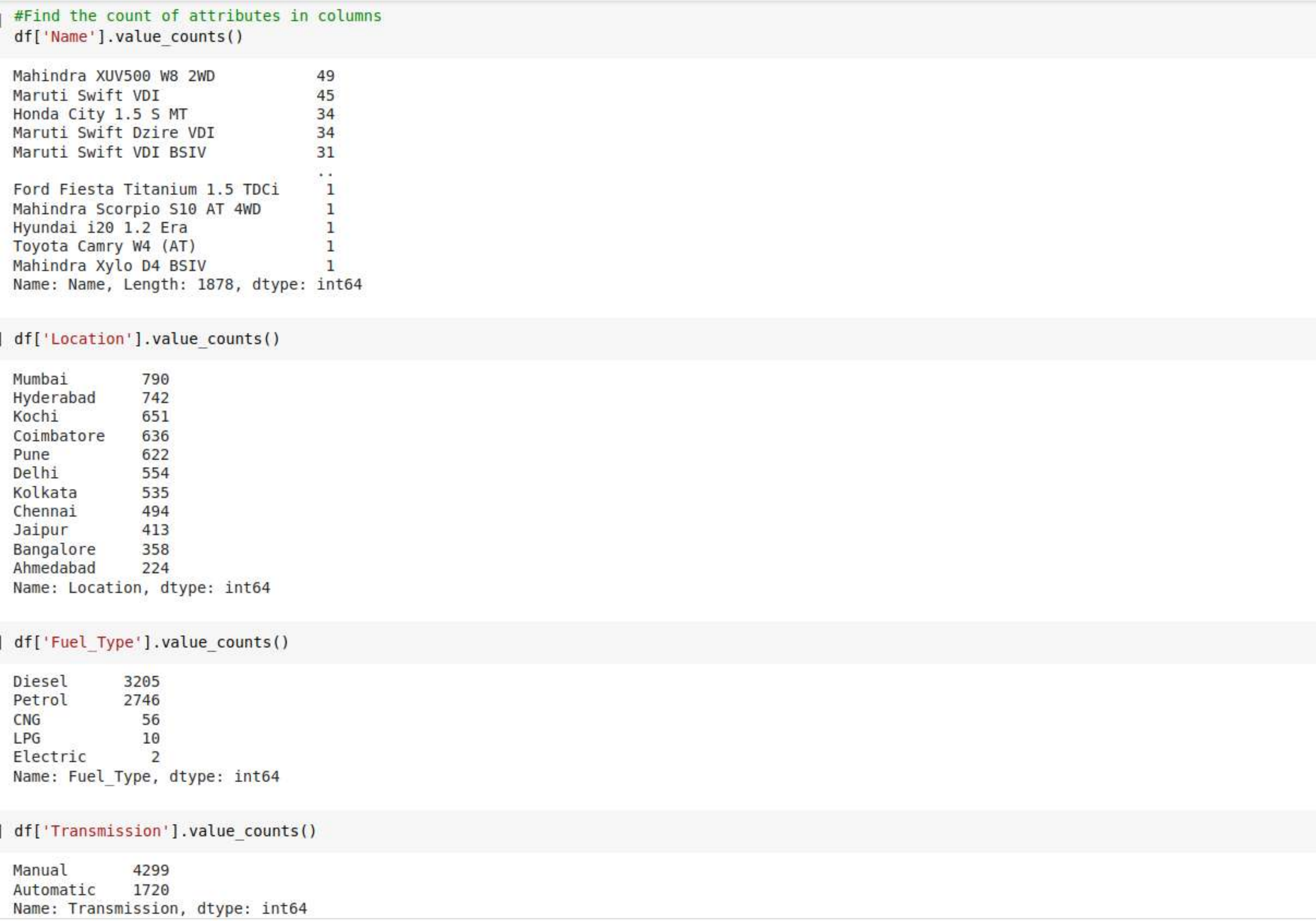
	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
6014	6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.4 kmpl	1248 CC	74 bhp	5.0	7.88 Lakh	4.75
6015	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.4 kmpl	1120 CC	71 bhp	5.0	NaN	4.00
6016	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.0 kmpl	2498 CC	112 bhp	8.0	NaN	2.90
6017	6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First	18.9 kmpl	998 CC	67.1 bhp	5.0	NaN	2.65
6018	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44 kmpl	936 CC	57.6 bhp	5.0	NaN	2.50

```
df.columns
```

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

```
# checking missing values
df.isna().sum()
```

```
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
```



```
#Find the count of attributes in columns
df['Name'].value_counts()
```

```
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
```

```
df['Location'].value_counts()
```

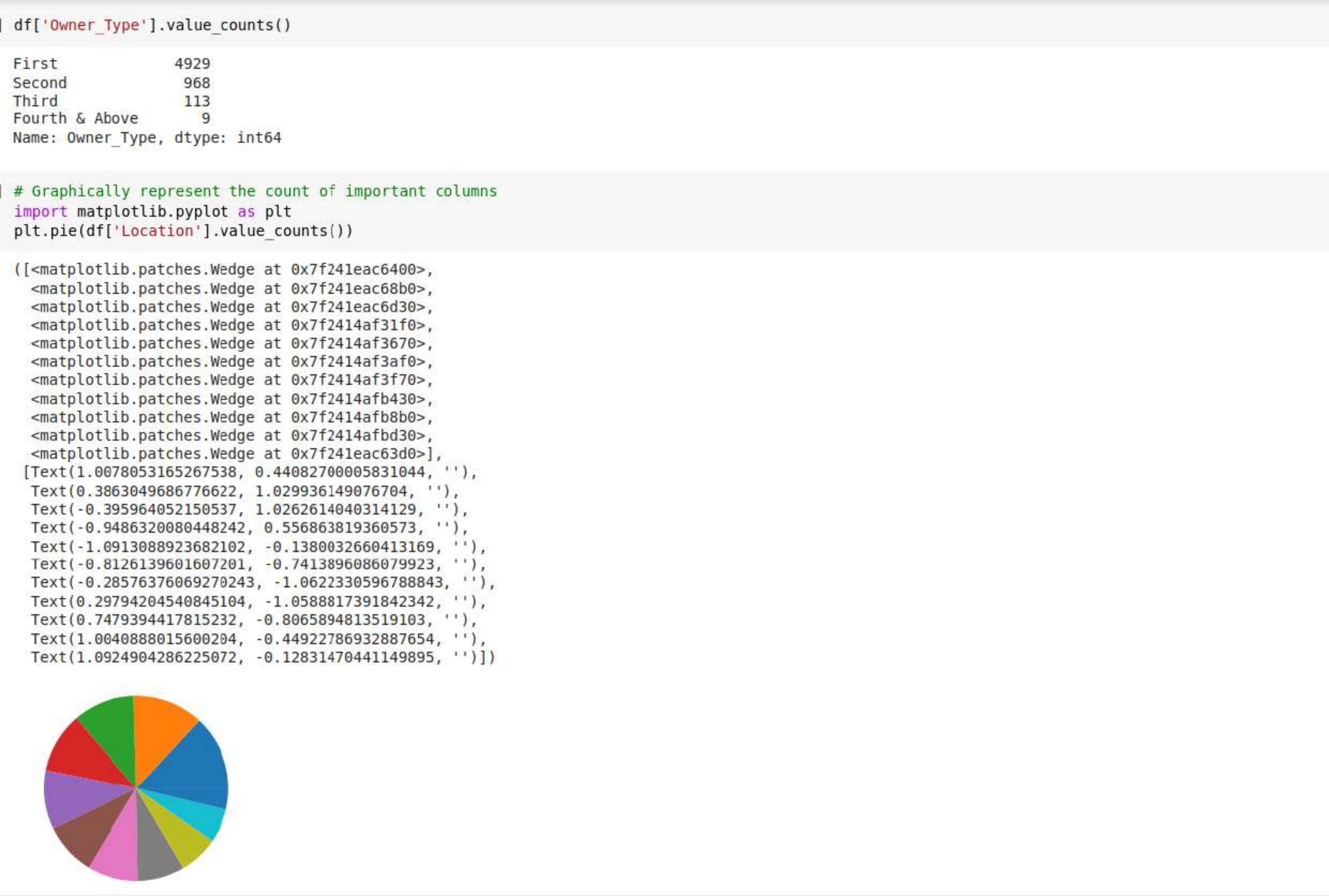
```
Mumbai        790
Hyderabad     742
Kochi         651
Coimbatore    636
Pune          622
Delhi         554
Kolkata       535
Chennai       494
Jaipur        413
Bangalore     358
Ahmedabad     224
Name: Location, dtype: int64
```

```
df['Fuel_Type'].value_counts()
```

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

```
df['Transmission'].value_counts()
```

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

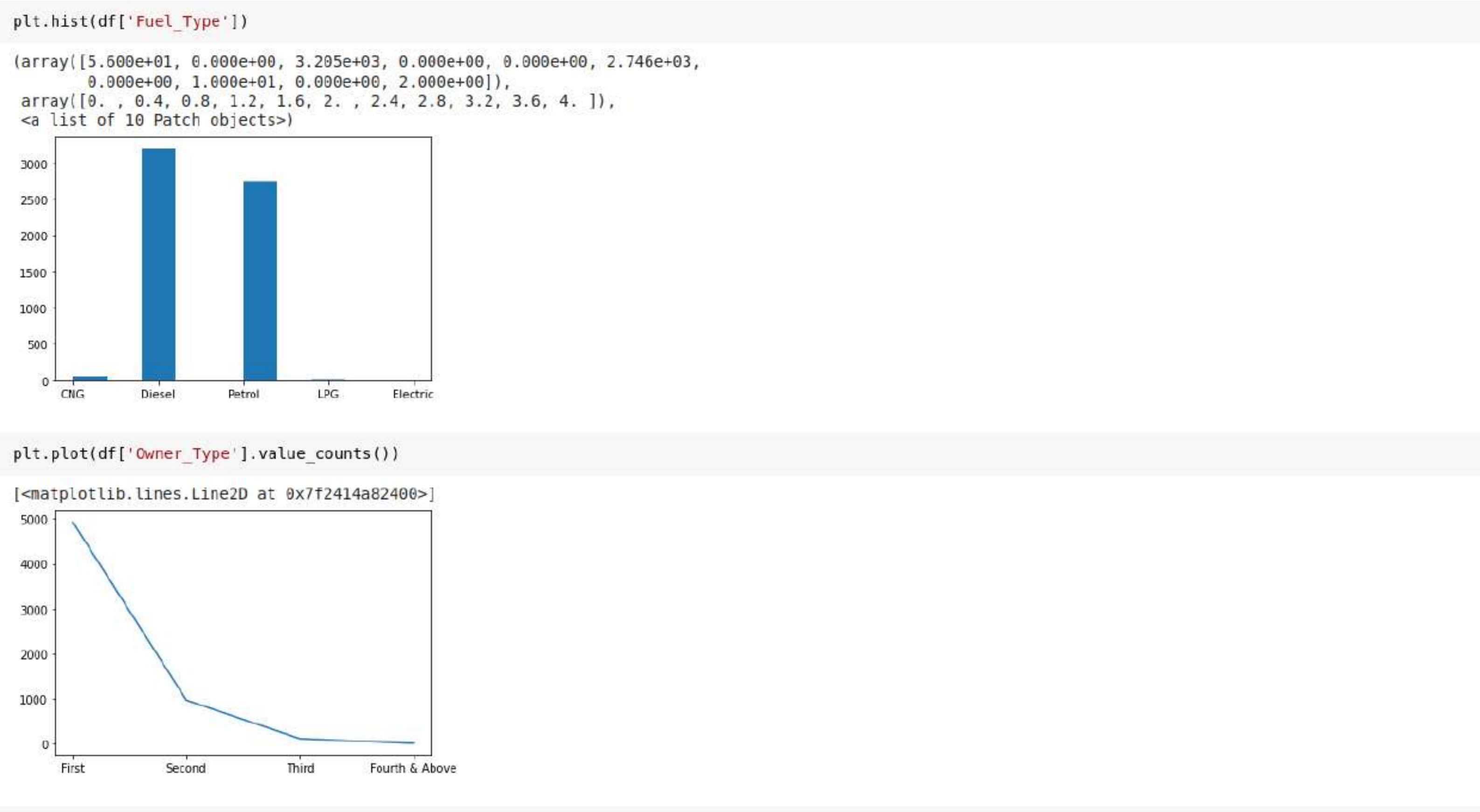


```
df['Owner_Type'].value_counts()
```

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

```
# Graphically represent the count of important columns
import matplotlib.pyplot as plt
plt.pie(df['Location'].value_counts())
```

```
([<matplotlib.patches.Wedge at 0x7f241eac6400>,
  <matplotlib.patches.Wedge at 0x7f241eac68b0>,
  <matplotlib.patches.Wedge at 0x7f241eac6d30>,
  <matplotlib.patches.Wedge at 0x7f2414af31f0>,
  <matplotlib.patches.Wedge at 0x7f2414af3670>,
  <matplotlib.patches.Wedge at 0x7f2414af3af0>,
  <matplotlib.patches.Wedge at 0x7f2414af3f70>,
  <matplotlib.patches.Wedge at 0x7f2414afb430>,
  <matplotlib.patches.Wedge at 0x7f2414afb8b0>,
  <matplotlib.patches.Wedge at 0x7f2414afbd30>,
  <matplotlib.patches.Wedge at 0x7f241eac63d0>],
 [Text(1.0078053165267538, 0.44082700005831044, ''),
  Text(0.3863049686776622, 1.029936149076704, ''),
  Text(-0.395964052150537, 1.0262614040314129, ''),
  Text(-0.9486320080448242, 0.556863819360573, ''),
  Text(-1.0913088923682102, -0.1380032660413169, ''),
  Text(-0.8126139601607201, -0.7413896086079923, ''),
  Text(-0.28576376069270243, -1.0622330596788843, ''),
  Text(0.29794204540845104, -1.0588817391842342, ''),
  Text(0.7479394417815232, -0.8065894813519103, ''),
  Text(1.0040888015600204, -0.44922786932887654, ''),
  Text(1.0924904286225072, -0.12831470441149895, '')])
```

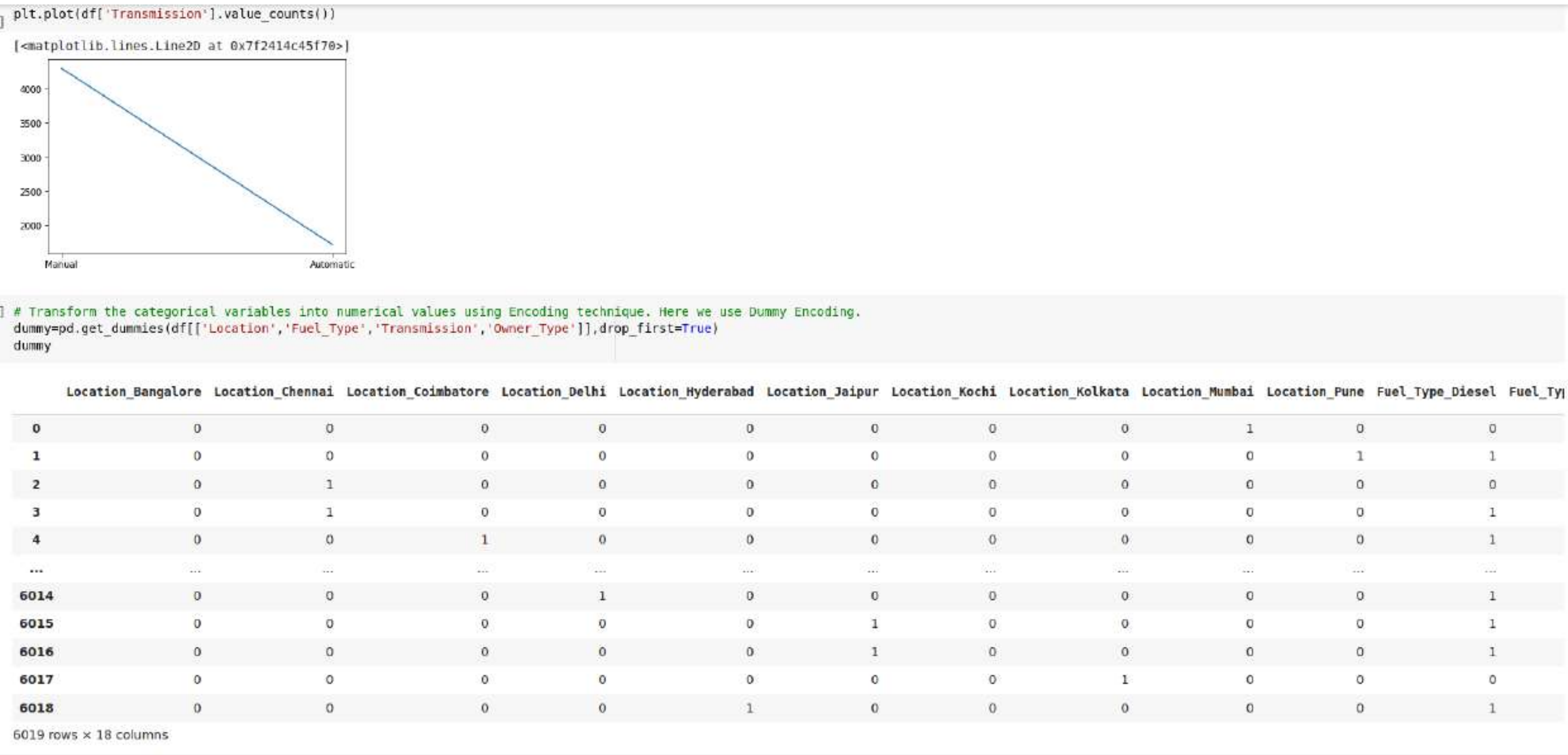


```
plt.hist(df['Fuel_Type'])
```

```
(array([5.600e+01, 0.000e+00, 3.205e+03, 0.000e+00, 0.000e+00, 2.746e+03,
        0.000e+00, 1.000e+01, 0.000e+00, 2.000e+00]),
 array([0. , 0.4, 0.8, 1.2, 1.6, 2. , 2.4, 2.8, 3.2, 3.6, 4. ]),
 <a list of 10 Patch objects>)
```

```
plt.plot(df['Owner_Type'].value_counts())
```

```
[<matplotlib.lines.Line2D at 0x7f2414a82400>]
```



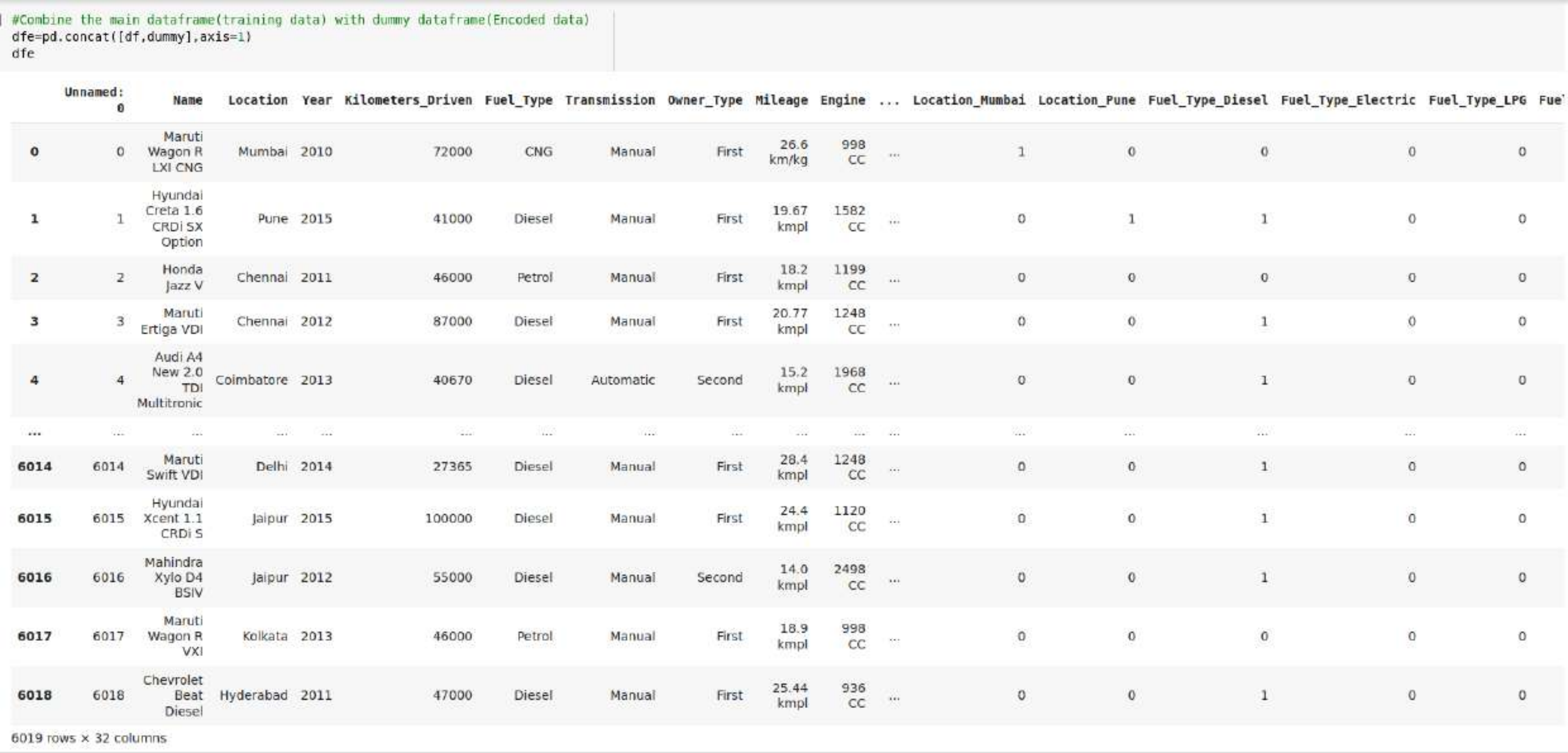
```
plt.plot(df['Transmission'].value_counts())
```

```
[<matplotlib.lines.Line2D at 0x7f2414c45f70>]
```

```
# Transform the categorical variables into numerical values using Encoding technique. Here we use Dummy Encoding.
dummy=pd.get_dummies(df[['Location','Fuel_Type','Transmission','Owner_Type']],drop_first=True)
dummy
```

	Location_Bangalore	Location_Chennai	Location_Coimbatore	Location_Delhi	Location_Hyderabad	Location_Jaipur	Location_Kochi	Location_Kolkata	Location_Mumbai	Location_Pune	Fuel_Type_Diesel	Fuel_Ty
0	0	0	0	0	0	0	0	0	1	0	0	
1	0	0	0	0	0	0	0	0	0	1	1	
2	0	1	0	0	0	0	0	0	0	0	0	
3	0	1	0	0	0	0	0	0	0	0	1	
4	0	0	1	0	0	0	0	0	0	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	
6014	0	0	0	1	0	0	0	0	0	0	1	
6015	0	0	0	0	0	1	0	0	0	0	1	
6016	0	0	0	0	0	1	0	0	0	0	1	
6017	0	0	0	0	0	0	0	1	0	0	0	
6018	0	0	0	0	1	0	0	0	0	0	1	

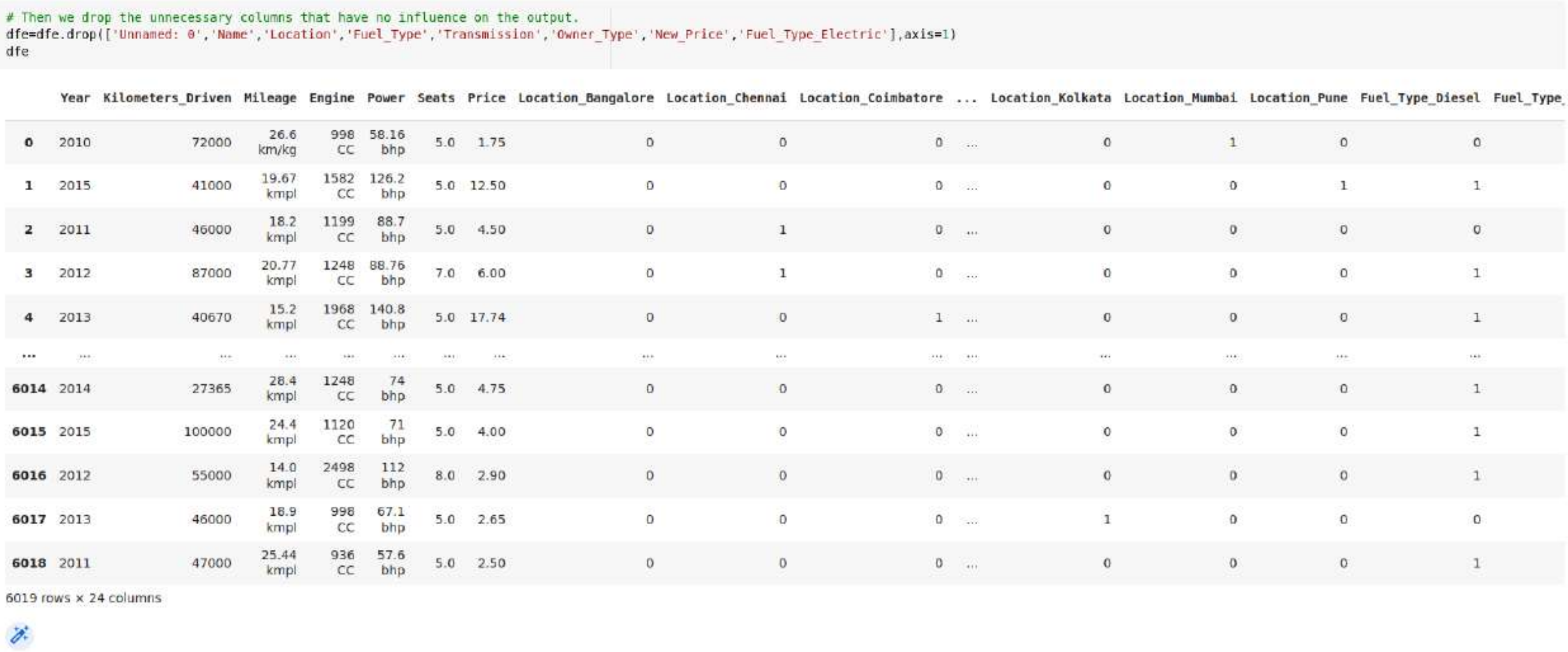
6019 rows × 18 columns



```
#Combine the main dataframe(training data) with dummy dataframe(Encoded data)
dfe=pd.concat([df,dummy],axis=1)
dfe
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	...	Location_Mumbai	Location_Pune	Fuel_Type_Diesel	Fuel_Type_Electric	Fuel_Type_LPG	Fue
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	...	1	0	0	0	0	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	...	0	1	1	0	0	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	...	0	0	0	0	0	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	...	0	0	1	0	0	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	...	0	0	1	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
6014	6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.4 kmpl	1248 CC	...	0	0	1	0	0	
6015	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.4 kmpl	1120 CC	...	0	0	1	0	0	
6016	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.0 kmpl	2498 CC	...	0	0	1	0	0	
6017	6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First	18.9 kmpl	998 CC	...	0	0	0	0	0	
6018	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44 kmpl	936 CC	...	0	0	1	0	0	

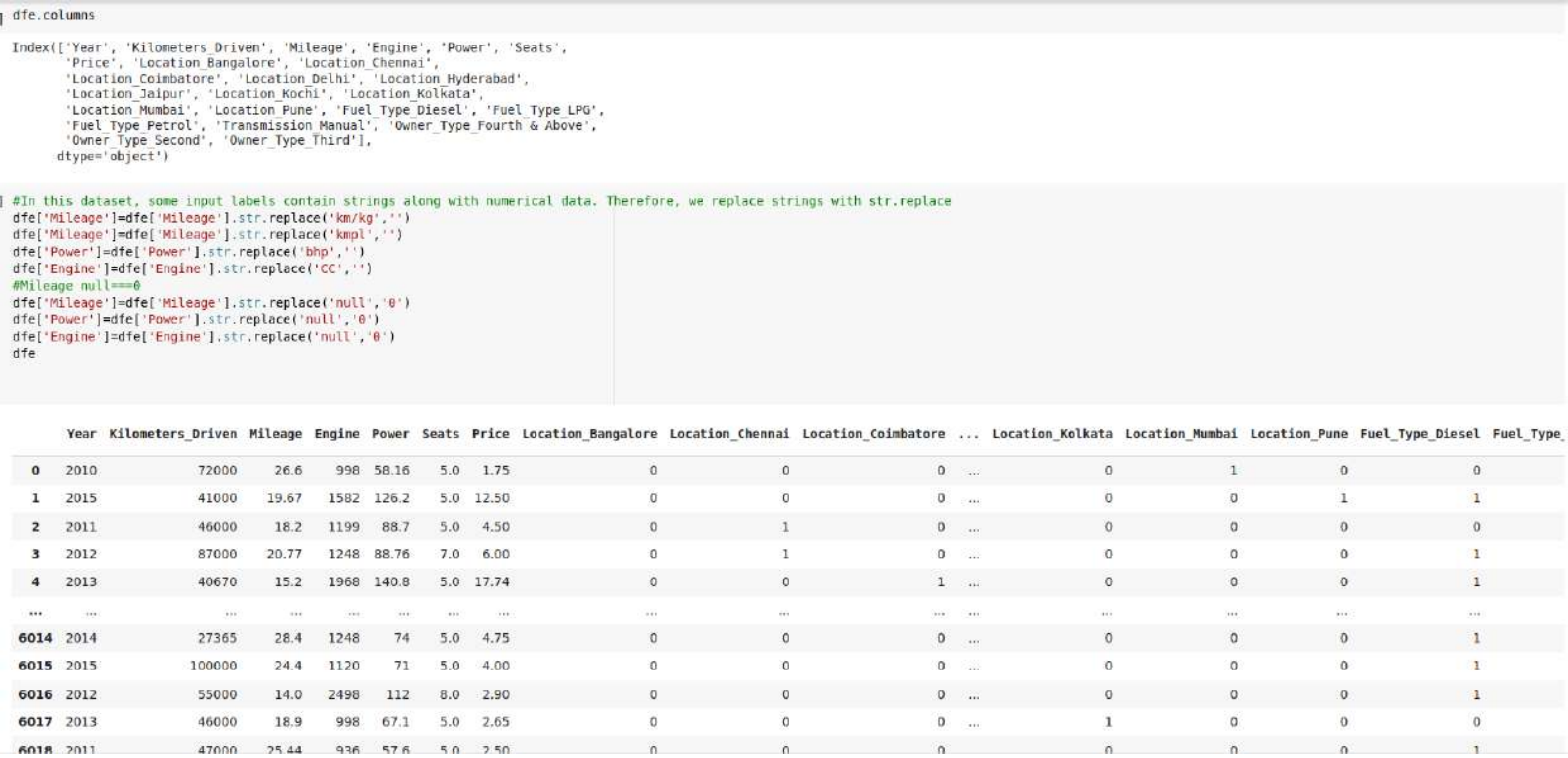
6019 rows × 32 columns



```
# Then we drop the unnecessary columns that have no influence on the output.
dfe=dfe.drop(['Unnamed: 0','Name','Location','Fuel_Type','Transmission','Owner_Type','New_Price','Fuel_Type_Electric'],axis=1)
dfe
```

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

6019 rows × 24 columns

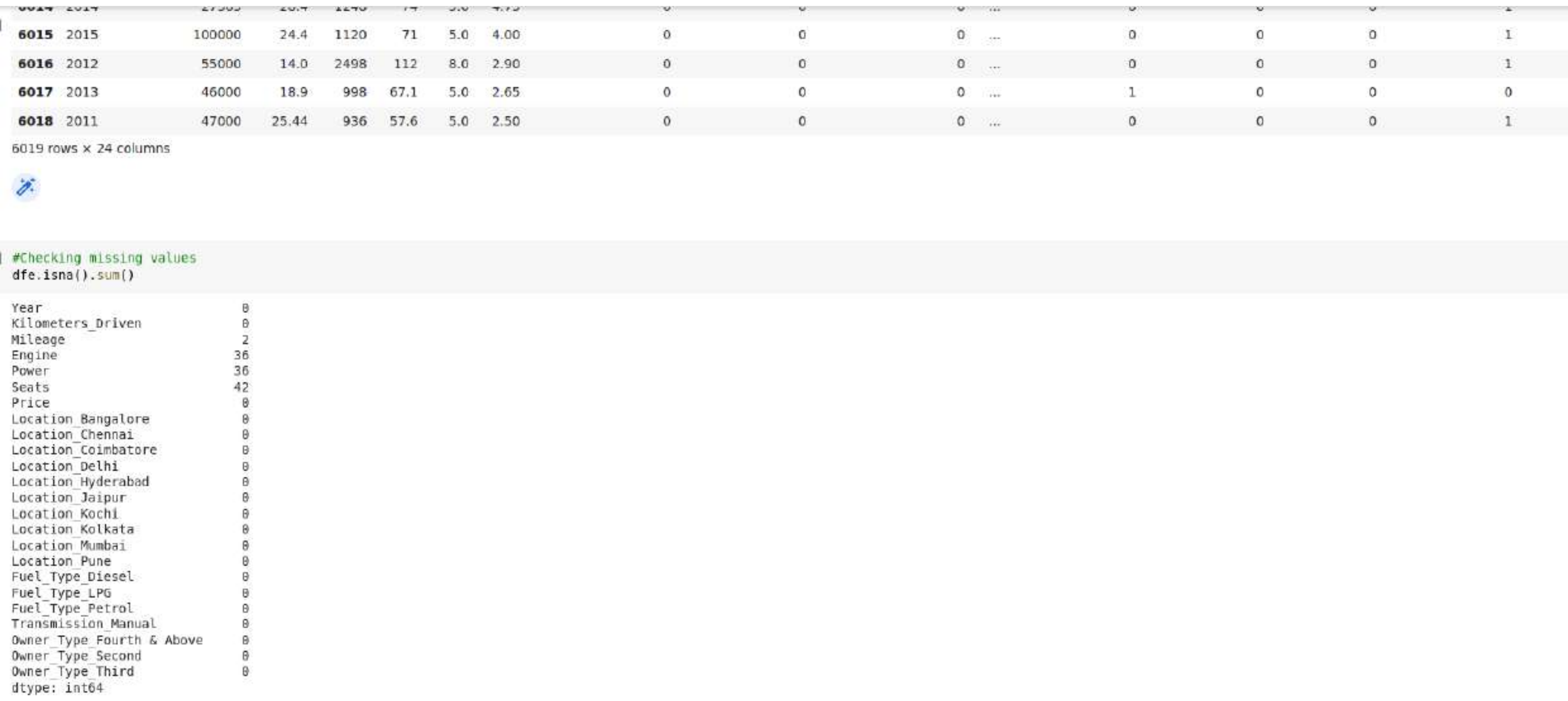


```
dfe.columns
```

```
Index(['Year', 'Kilometers_Driven', 'Mileage', 'Engine', 'Power', 'Seats',
       'Price', '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 this dataset, some input labels contain strings along with numerical data. Therefore, we replace strings with str.replace
dfe['Mileage']=dfe['Mileage'].str.replace('km/kg','')
dfe['Mileage']=dfe['Mileage'].str.replace('kmpl','')
dfe['Power']=dfe['Power'].str.replace('bhp','')
dfe['Engine']=dfe['Engine'].str.replace('CC','')
#Mileage null===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
```

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

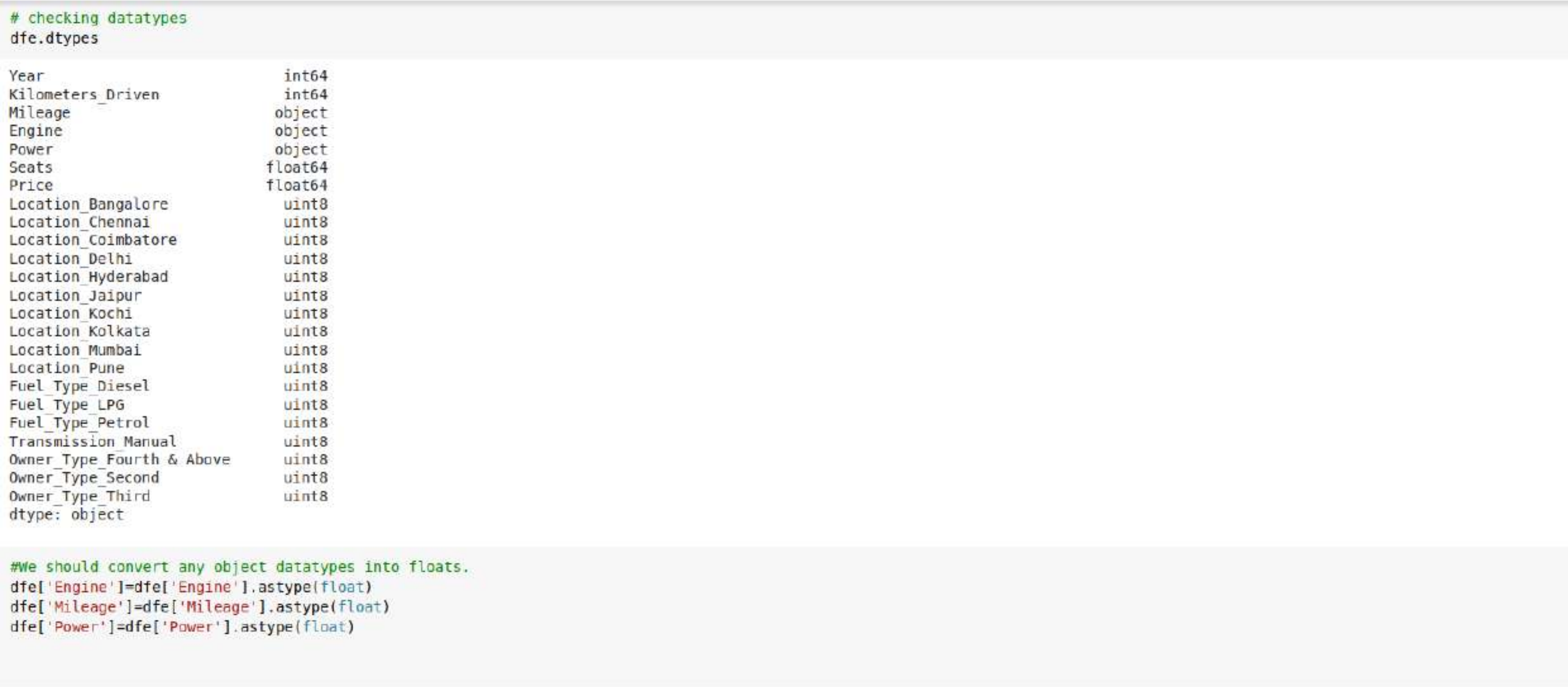


6014	2014	27365	28.4	1248	74	5.0	4.75	0	0	0	...	0	0	0	1
6015	2015	100000	24.4	1120	71	5.0	4.00	0	0	0	...	0	0	0	1
6016	2012	55000	14.0	2498	112	8.0	2.90	0	0	0	...	0	0	0	1
6017	2013	46000	18.9	998	67.1	5.0	2.65	0	0	0	...	1	0	0	0
6018	2011	47000	25.44	936	57.6	5.0	2.50	0	0	0	...	0	0	0	1

6019 rows × 24 columns

```
#Checking missing values
dfe.isna().sum()
```

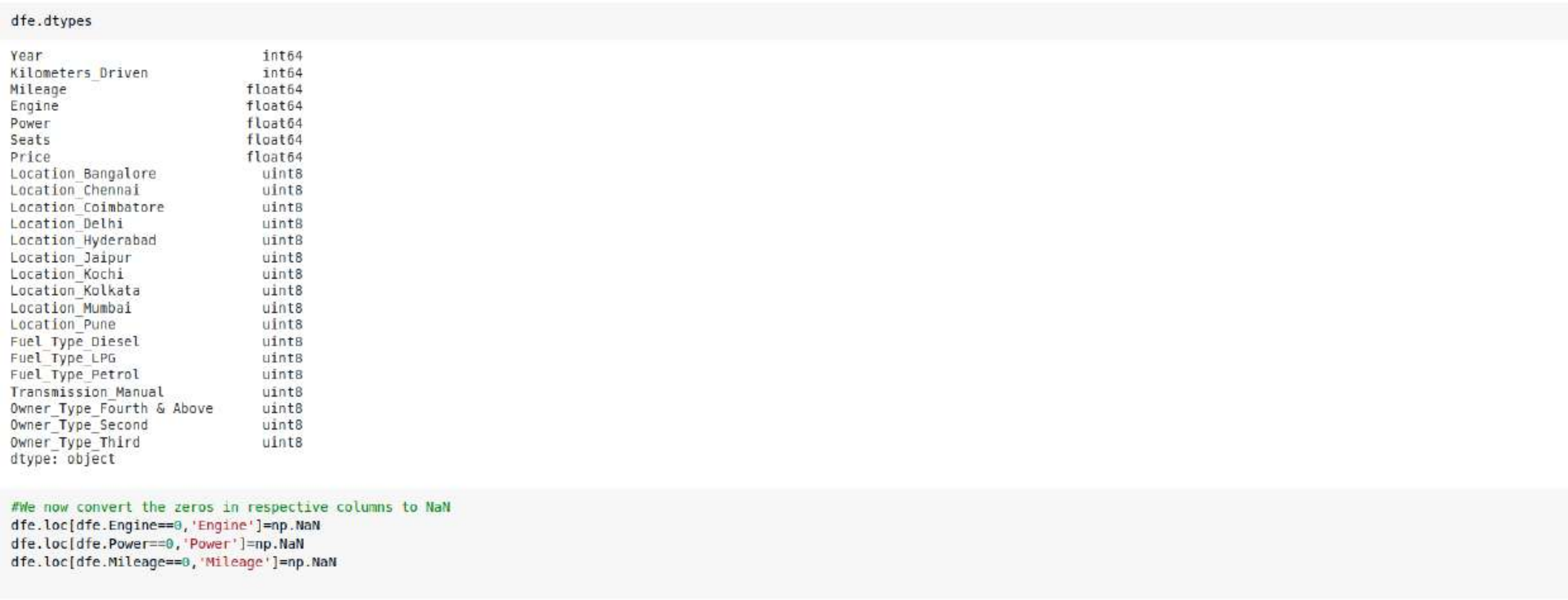
```
Year                         0
Kilometers_Driven            0
Mileage                      2
Engine                      36
Power                       36
Seats                       42
Price                        0
Location_Bangalore           0
Location_Chennai             0
Location_Coimbatore          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_LPG                0
Fuel_Type_Petrol             0
Transmission_Manual          0
Owner_Type_Fourth & Above    0
Owner_Type_Second            0
Owner_Type_Third             0
dtype: int64
```



```
# checking datatypes
dfe.dtypes
```

```
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_LPG                  uint8
Fuel_Type_Petrol               uint8
Transmission_Manual            uint8
Owner_Type_Fourth & Above      uint8
Owner_Type_Second              uint8
Owner_Type_Third               uint8
dtype: object
```

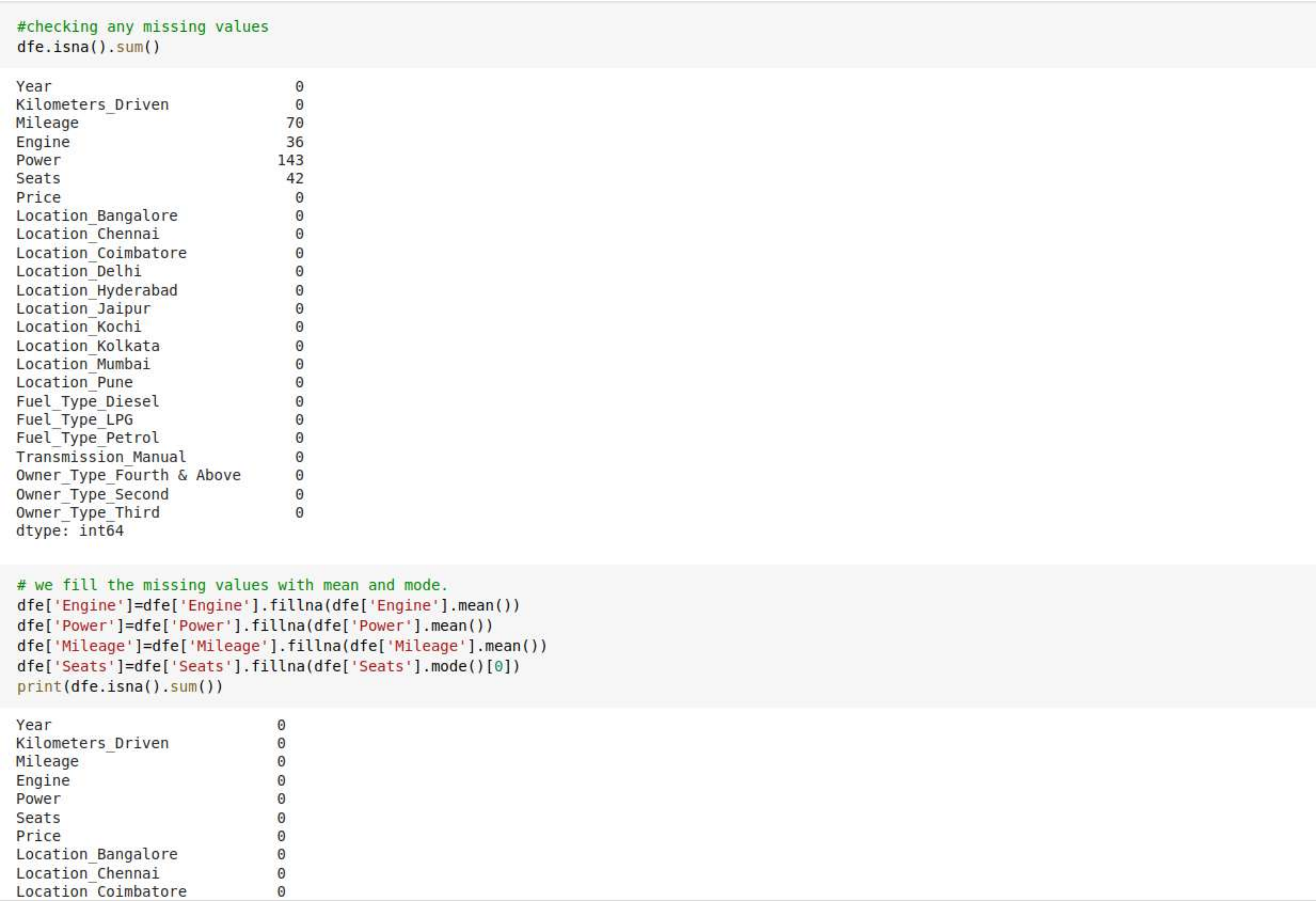
```
#We should convert any object datatypes into floats.
dfe['Engine']=dfe['Engine'].astype(float)
dfe['Mileage']=dfe['Mileage'].astype(float)
dfe['Power']=dfe['Power'].astype(float)
```



```
dfe.dtypes
```

```
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_LPG                 uint8
Fuel_Type_Petrol              uint8
Transmission_Manual           uint8
Owner_Type_Fourth & Above     uint8
Owner_Type_Second             uint8
Owner_Type_Third              uint8
dtype: object
```

```
#We now convert the zeros in respective columns to 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
```

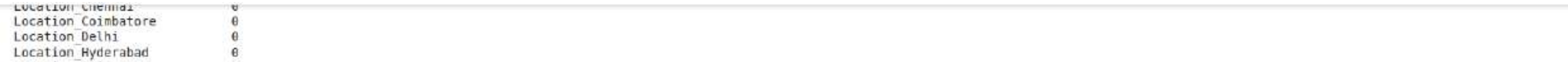


```
#checking any missing values
dfe.isna().sum()
```

```
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_LPG                  0
Fuel_Type_Petrol               0
Transmission_Manual            0
Owner_Type_Fourth & Above      0
Owner_Type_Second              0
Owner_Type_Third               0
dtype: int64
```

```
# we fill the missing values with mean and mode.
dfe['Engine']=dfe['Engine'].fillna(dfe['Engine'].mean())
dfe['Power']=dfe['Power'].fillna(dfe['Power'].mean())
dfe['Mileage']=dfe['Mileage'].fillna(dfe['Mileage'].mean())
dfe['Seats']=dfe['Seats'].fillna(dfe['Seats'].mode()[0])
print(dfe.isna().sum())
```

```
Year                         0
Kilometers_Driven            0
Mileage                      0
Engine                       0
Power                        0
Seats                        0
Price                        0
Location_Bangalore           0
Location_Chennai             0
Location_Coimbatore          0
```

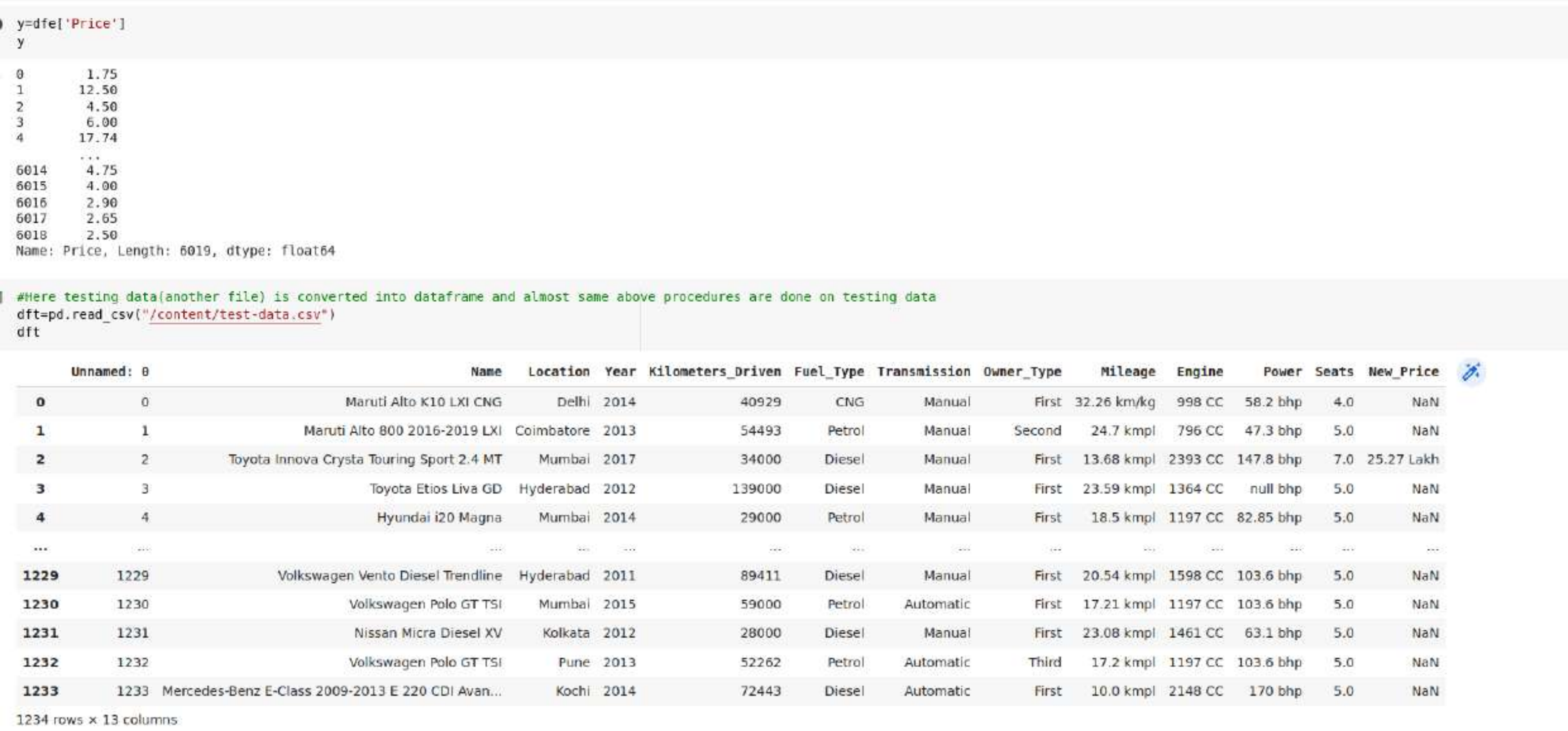


```
Location_Chennai             0
Location_Coimbatore          0
Location_Delhi               0
Location_Hyderabad           0
Location_Jaipur              0
Location_Kochi               0
Location_Kolkata             0
Location_Mumbai              0
Location_Pune                0
Fuel_Type_Diesel             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
```

```
# Now we asssign the input(x) and output(y) datas
x=dfe.drop(['Price'],axis=1)
x
```

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

6019 rows × 23 columns



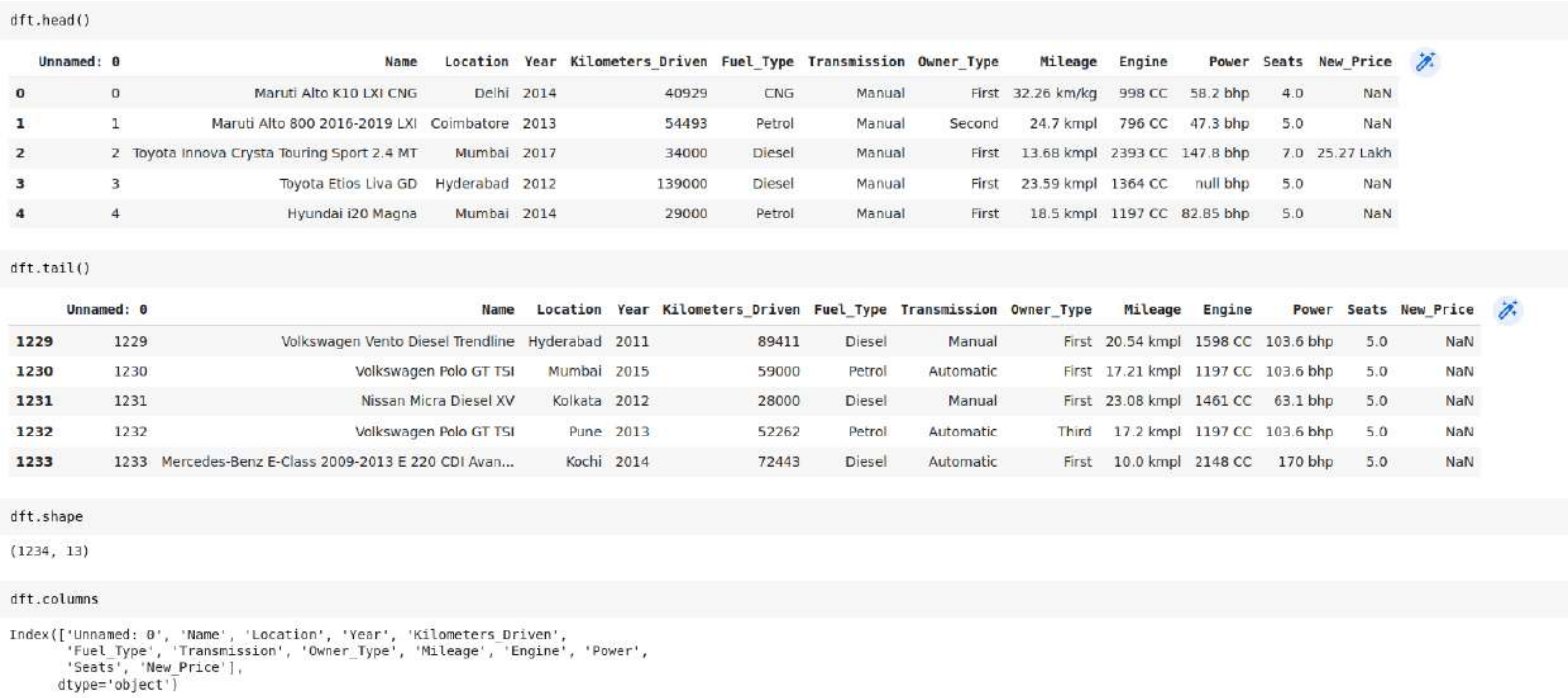
```
y=dfe['Price']
y
```

```
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
```

```
#Here testing data(another file) is converted into dataframe and almost same above procedures are done on testing data
dft=pd.read_csv("/content/test-data.csv")
dft
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	First	32.26 km/kg	998 CC	58.2 bhp	4.0	NaN
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Second	24.7 kmpl	796 CC	47.3 bhp	5.0	NaN
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	First	13.68 kmpl	2393 CC	147.8 bhp	7.0	25.27 Lakh
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	First	23.59 kmpl	1364 CC	null bhp	5.0	NaN
4	4	Hyundai i20 Magna	Mumbai	2014	29000	Petrol	Manual	First	18.5 kmpl	1197 CC	82.85 bhp	5.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1229	1229	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN
1230	1230	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	17.21 kmpl	1197 CC	103.6 bhp	5.0	NaN
1231	1231	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN
1232	1232	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	103.6 bhp	5.0	NaN
1233	1233	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	First	10.0 kmpl	2148 CC	170 bhp	5.0	NaN

1234 rows × 13 columns



```
dft.head()
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	First	32.26 km/kg	998 CC	58.2 bhp	4.0	NaN
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Second	24.7 kmpl	796 CC	47.3 bhp	5.0	NaN
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	First	13.68 kmpl	2393 CC	147.8 bhp	7.0	25.27 Lakh
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	First	23.59 kmpl	1364 CC	null bhp	5.0	NaN
4	4	Hyundai i20 Magna	Mumbai	2014	29000	Petrol	Manual	First	18.5 kmpl	1197 CC	82.85 bhp	5.0	NaN

```
dft.tail()
```

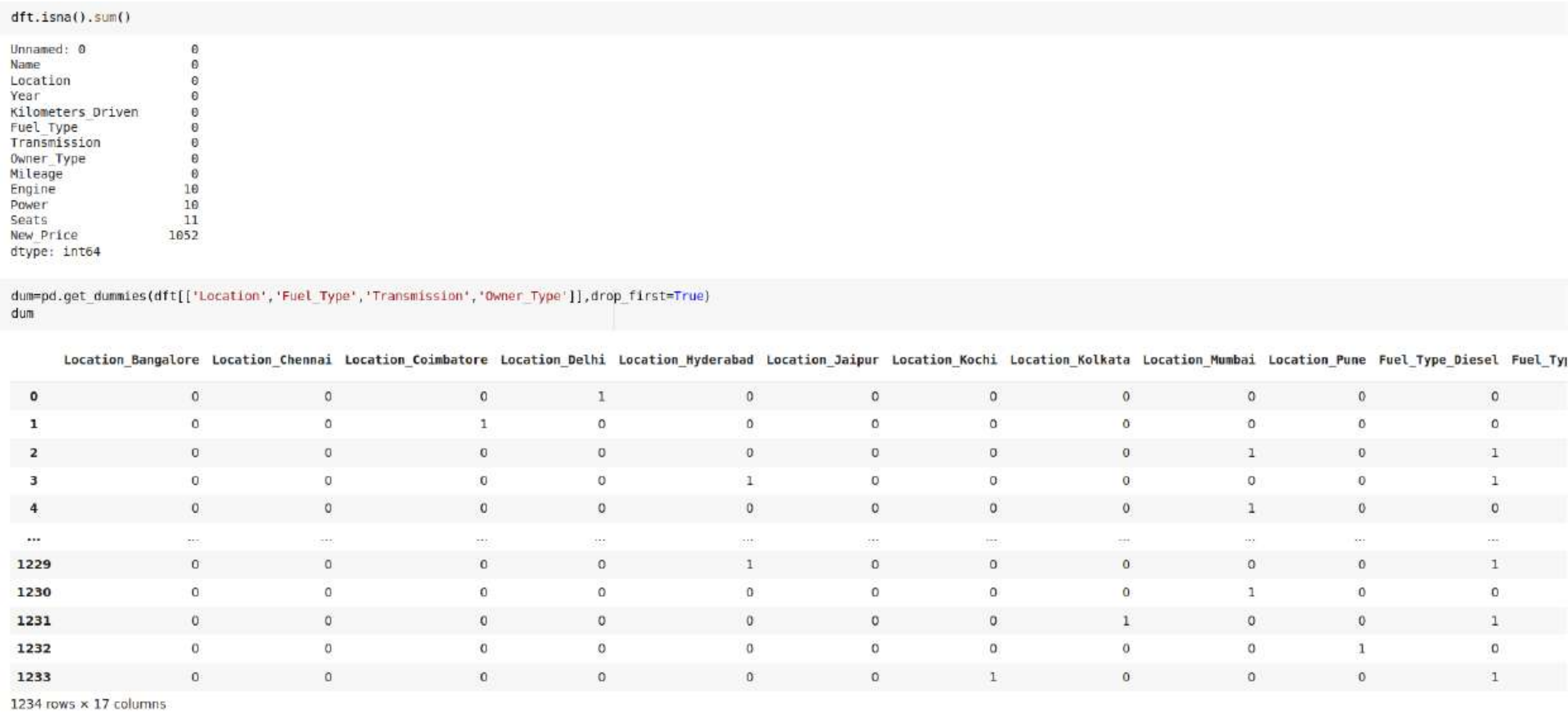
	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
1229	1229	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN
1230	1230	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	17.21 kmpl	1197 CC	103.6 bhp	5.0	NaN
1231	1231	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN
1232	1232	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	103.6 bhp	5.0	NaN
1233	1233	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	First	10.0 kmpl	2148 CC	170 bhp	5.0	NaN

```
dft.shape
```

```
(1234, 13)
```

```
dft.columns
```

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



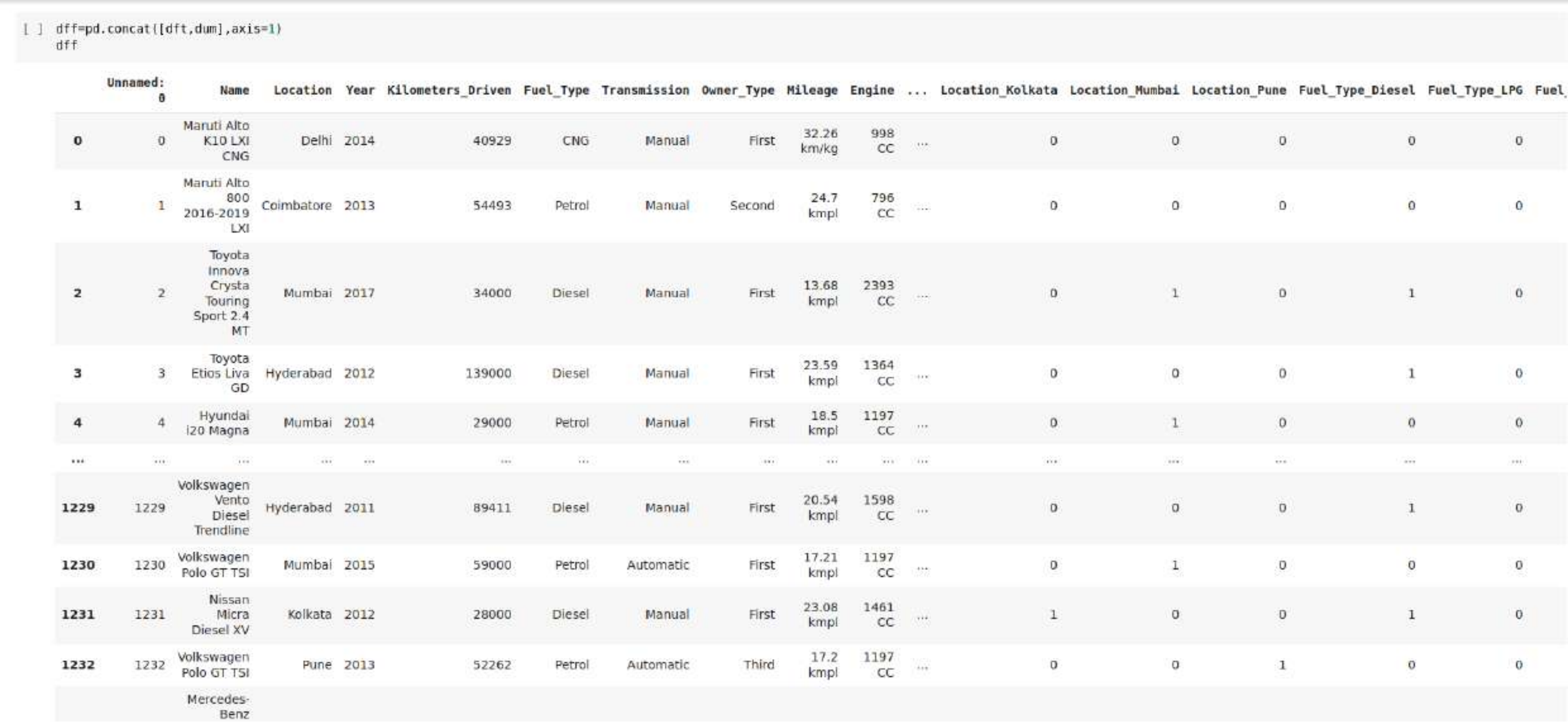
```
dft.isna().sum()
```

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

```
dum=pd.get_dummies(dft[['Location','Fuel_Type','Transmission','Owner_Type']],drop_first=True)
dum
```

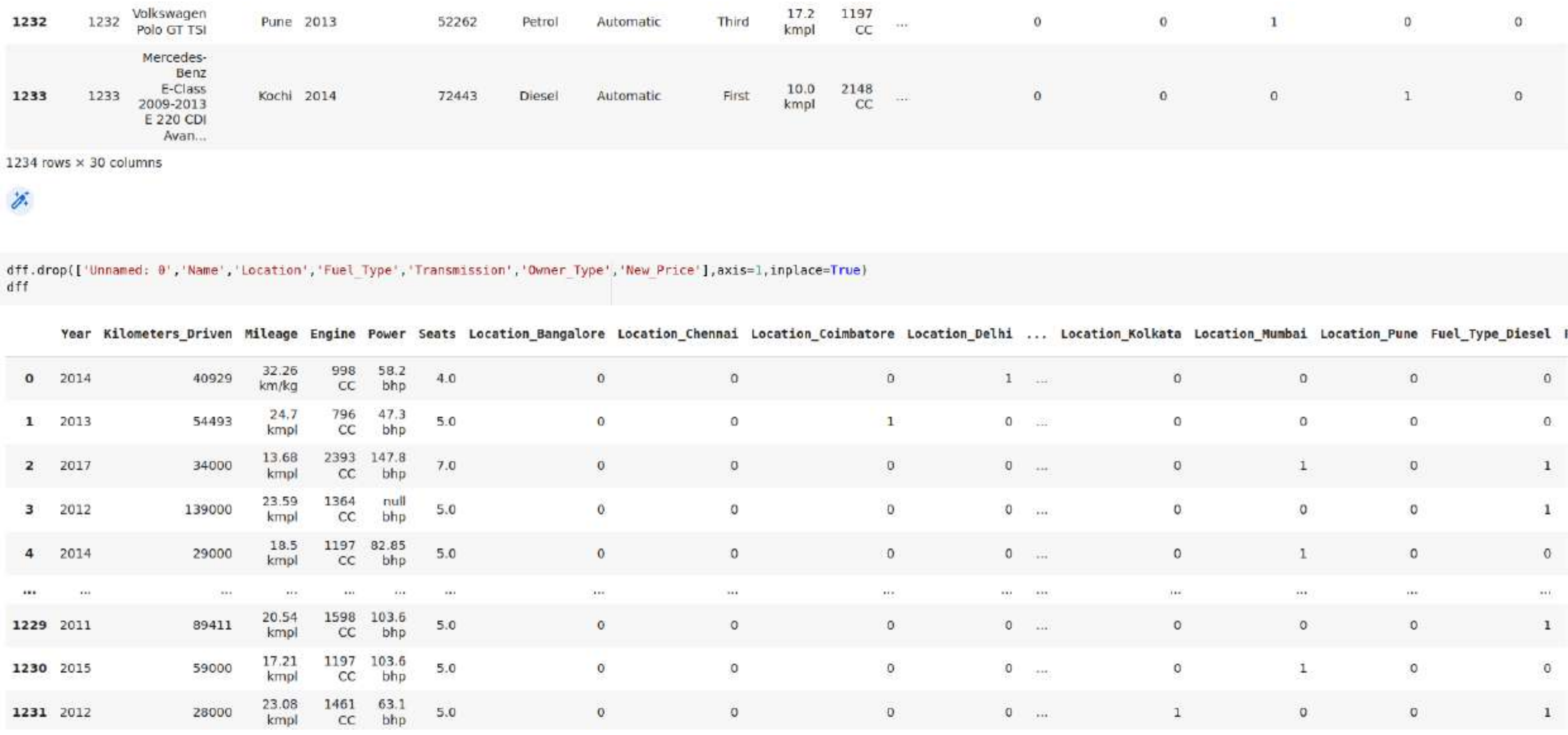
	Location_Bangalore	Location_Chennai	Location_Coimbatore	Location_Delhi	Location_Hyderabad	Location_Jaipur	Location_Kochi	Location_Kolkata	Location_Mumbai	Location_Pune	Fuel_Type_Diesel	Fuel_Typ
0	0	0	0	1	0	0	0	0	0	0	0	
1	0	0	1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	1	0	1	
3	0	0	0	0	1	0	0	0	0	0	1	
4	0	0	0	0	0	0	0	0	1	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	
1229	0	0	0	0	1	0	0	0	0	0	1	
1230	0	0	0	0	0	0	0	0	1	0	0	
1231	0	0	0	0	0	0	0	1	0	0	1	
1232	0	0	0	0	0	0	0	0	0	1	0	
1233	0	0	0	0	0	0	1	0	0	0	1	

1234 rows × 17 columns



```
[ ] dff=pd.concat([dft,dum],axis=1)
    dff
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	...	Location_Kolkata	Location_Mumbai	Location_Pune	Fuel_Type_Diesel	Fuel_Type_LPG	Fuel_
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	First	32.26 km/kg	998 CC	...	0	0	0	0	0	
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Second	24.7 kmpl	796 CC	...	0	0	0	0	0	
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	First	13.68 kmpl	2393 CC	...	0	1	0	1	0	
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	First	23.59 kmpl	1364 CC	...	0	0	0	1	0	
4	4	Hyundai i20 Magna	Mumbai	2014	29000	Petrol	Manual	First	18.5 kmpl	1197 CC	...	0	1	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1229	1229	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	20.54 kmpl	1598 CC	...	0	0	0	1	0	
1230	1230	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	17.21 kmpl	1197 CC	...	0	1	0	0	0	
1231	1231	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	23.08 kmpl	1461 CC	...	1	0	0	1	0	
1232	1232	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	...	0	0	1	0	0	
		Mercedes-Benz															

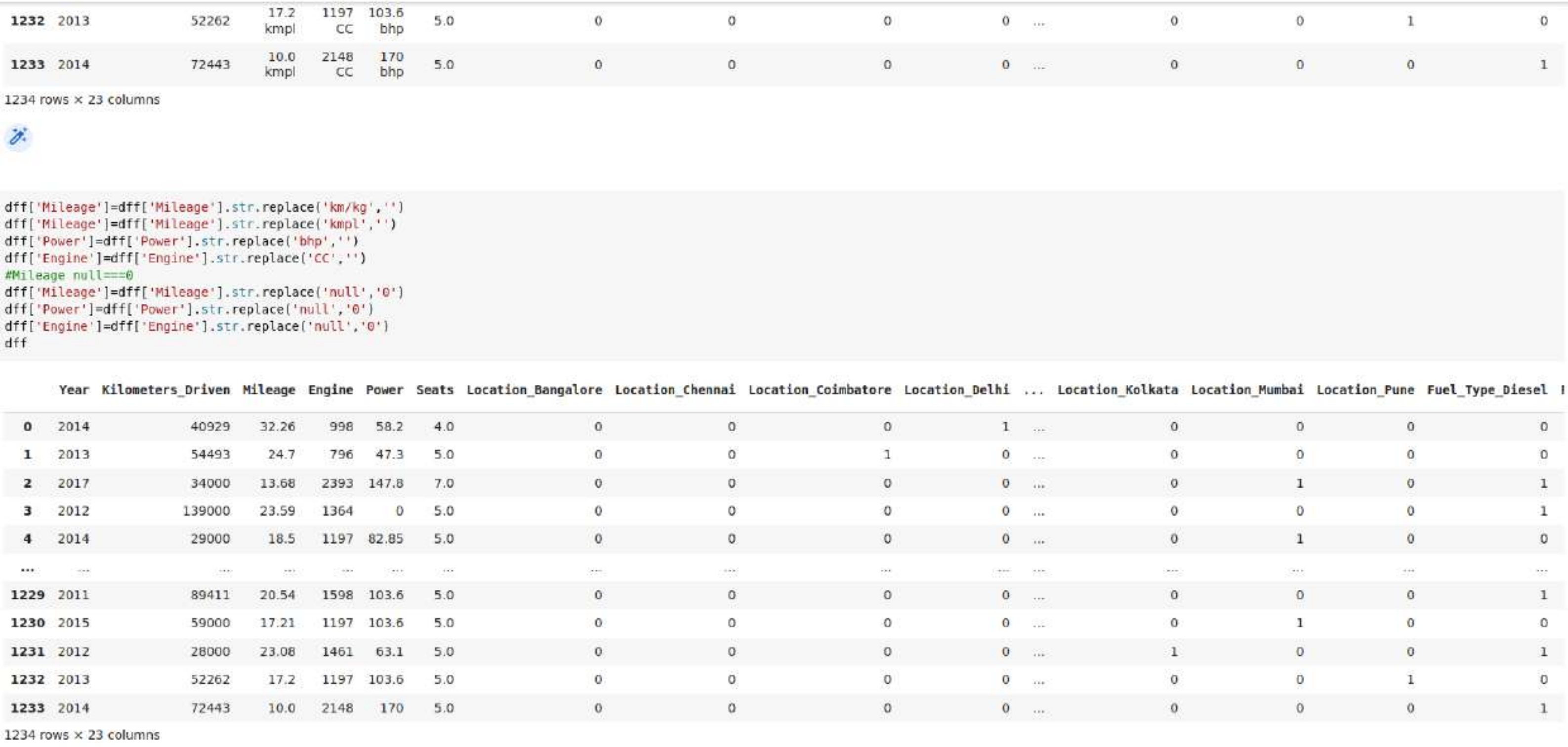


1232	1232	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	...	0	0	1	0	0
1233	1233	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	First	10.0 kmpl	2148 CC	...	0	0	0	1	0

1234 rows × 30 columns

```
dff.drop(['Unnamed: 0','Name','Location','Fuel_Type','Transmission','Owner_Type','New_Price'],axis=1,inplace=True)
dff
```

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



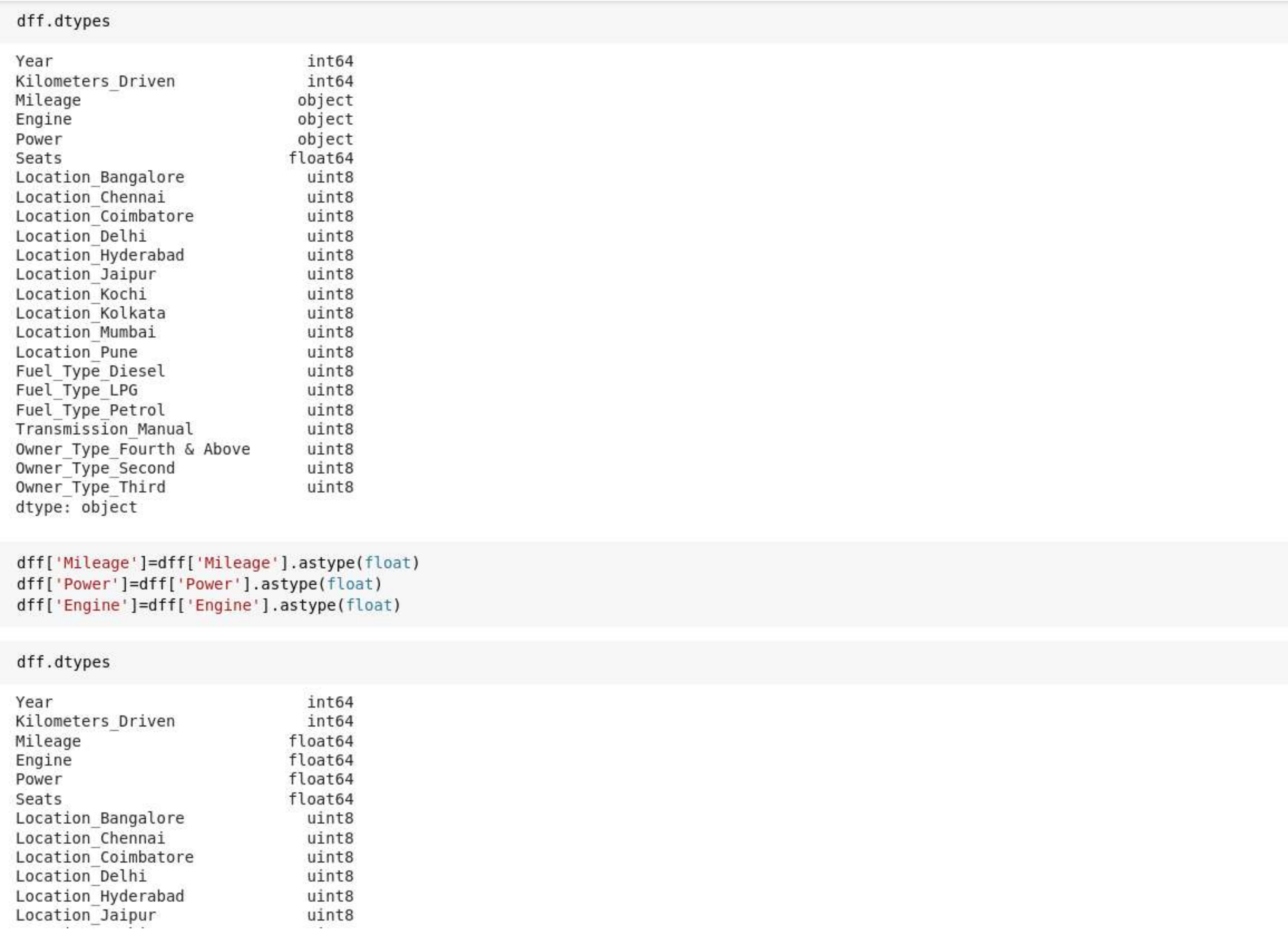
1232	2013	52262	17.2 kmpl	1197 CC	103.6 bhp	5.0	0	0	0	0	...	0	0	1	0
1233	2014	72443	10.0 kmpl	2148 CC	170 bhp	5.0	0	0	0	0	...	0	0	0	1

1234 rows × 23 columns

```
dff['Mileage']=dff['Mileage'].str.replace('km/kg','')
dff['Mileage']=dff['Mileage'].str.replace('kmpl','')
dff['Power']=dff['Power'].str.replace('bhp','')
dff['Engine']=dff['Engine'].str.replace('CC','')
#Mileage null===0
dff['Mileage']=dff['Mileage'].str.replace('null','0')
dff['Power']=dff['Power'].str.replace('null','0')
dff['Engine']=dff['Engine'].str.replace('null','0')
dff
```

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

1234 rows × 23 columns



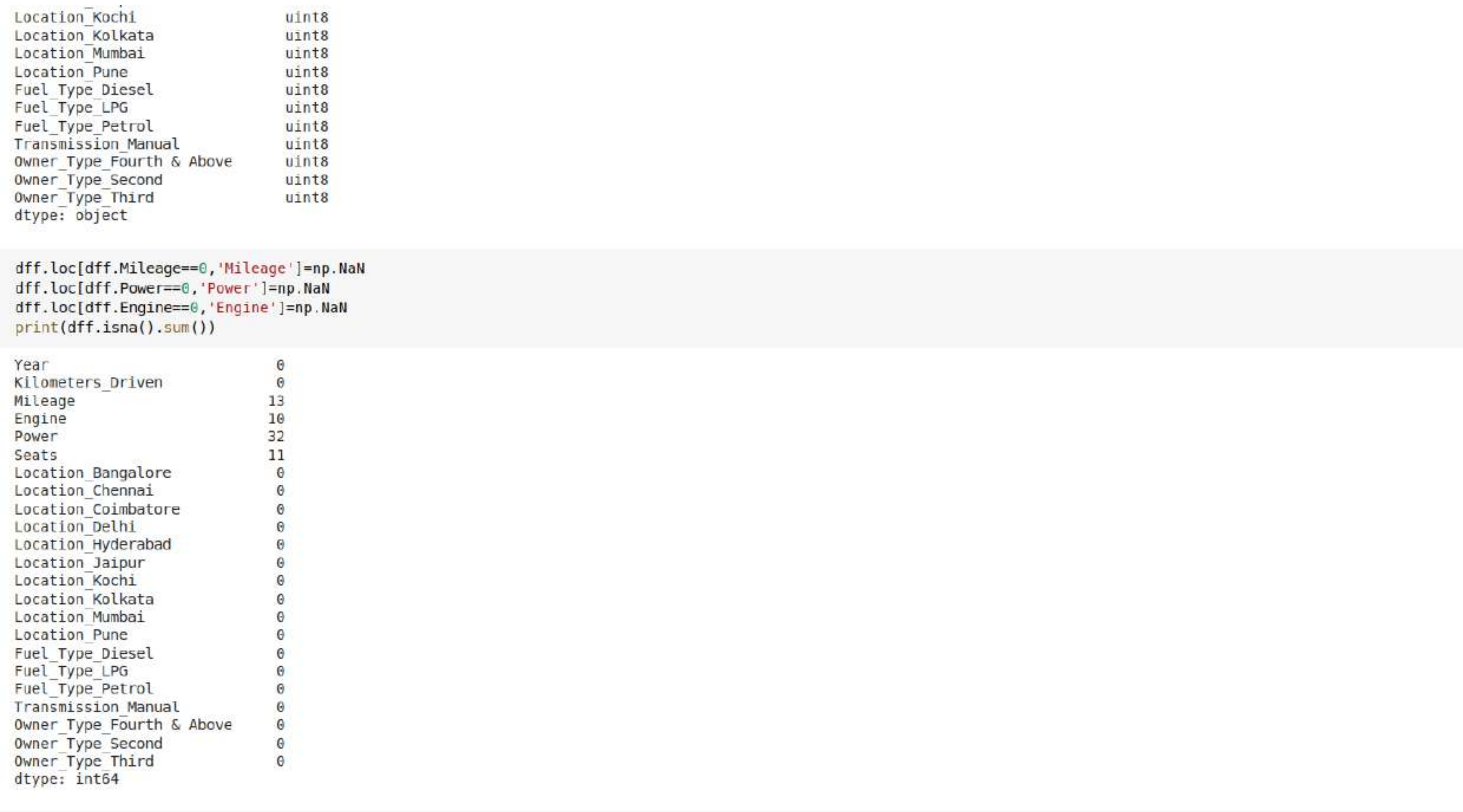
```
dff.dtypes
```

```
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
```

```
dff['Mileage']=dff['Mileage'].astype(float)
dff['Power']=dff['Power'].astype(float)
dff['Engine']=dff['Engine'].astype(float)
```

```
dff.dtypes
```

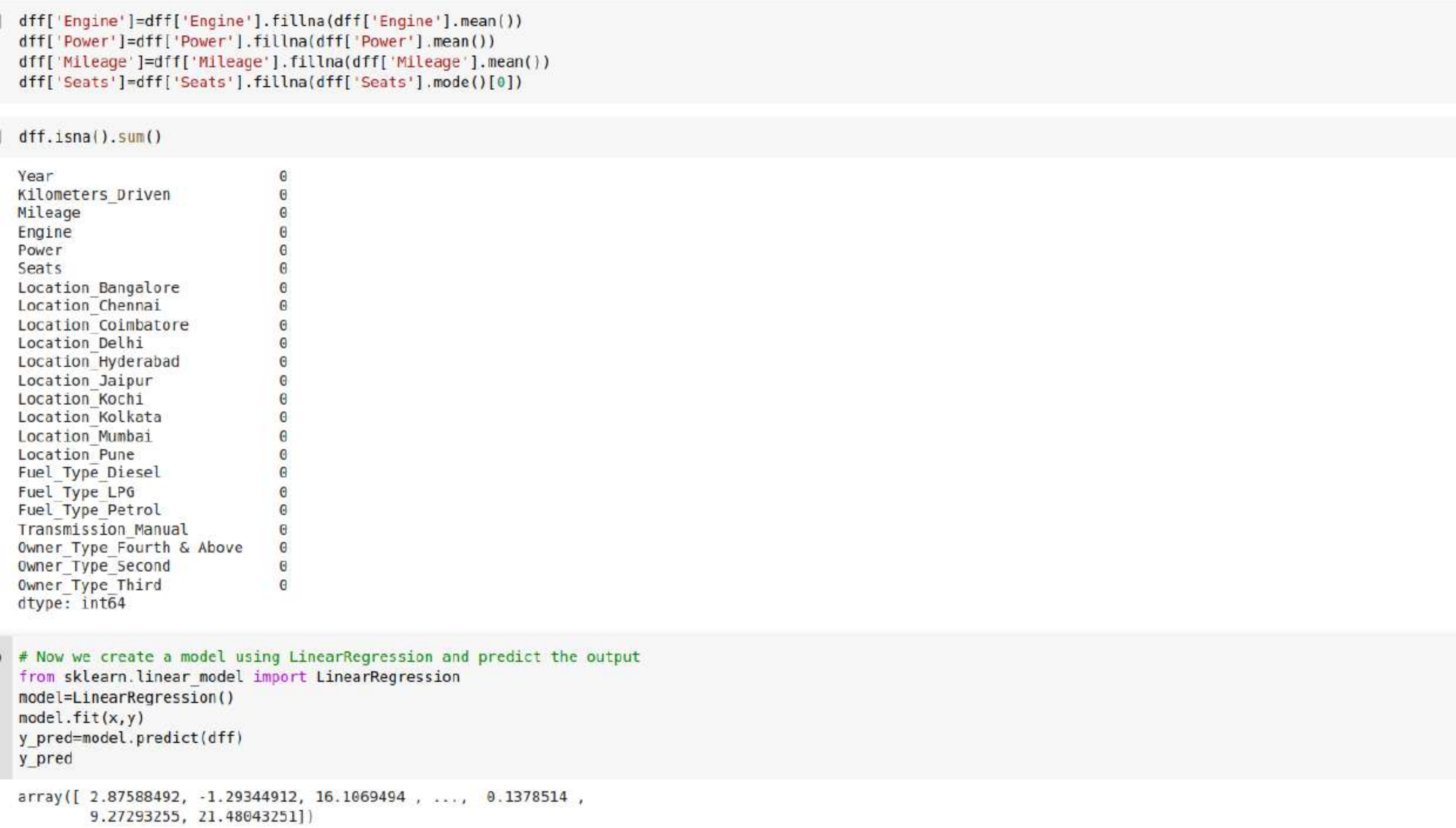
```
Year                           int64
Kilometers_Driven              int64
Mileage                      float64
Engine                       float64
Power                        float64
Seats                        float64
Location_Bangalore             uint8
Location_Chennai               uint8
Location_Coimbatore            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
```

```
dff.loc[dff.Mileage==0,'Mileage']=np.NaN
dff.loc[dff.Power==0,'Power']=np.NaN
dff.loc[dff.Engine==0,'Engine']=np.NaN
print(dff.isna().sum())
```

```
Year                          0
Kilometers_Driven             0
Mileage                      13
Engine                       10
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
```



```
dff['Engine']=dff['Engine'].fillna(dff['Engine'].mean())
dff['Power']=dff['Power'].fillna(dff['Power'].mean())
dff['Mileage']=dff['Mileage'].fillna(dff['Mileage'].mean())
dff['Seats']=dff['Seats'].fillna(dff['Seats'].mode()[0])
```

```
dff.isna().sum()
```

```
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
```

```
# Now we create a model using LinearRegression and predict the output
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x,y)
y_pred=model.predict(dff)
y_pred
```

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