In [1]:

```
import pandas as pd
```

In [4]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score
```

In [5]:

```
car_df=pd.read_csv("CarData.csv")
```

In [6]:

```
car_df
```

Out[6]:

| | name | year | selling_price | km_driven | fuel | seller_type | transmission | owner | mileage | en |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Maruti Swift Dzire VDI | 2014 | 450000 | 145500 | Diesel | Individual | Manual | First Owner | 23.4 kmpl | |
| **1** | Skoda Rapid 1.5 TDI Ambition | 2014 | 370000 | 120000 | Diesel | Individual | Manual | Second Owner | 21.14 kmpl | |
| **2** | Honda City 2017-2020 EXi | 2006 | 158000 | 140000 | Petrol | Individual | Manual | Third Owner | 17.7 kmpl | |
| **3** | Hyundai i20 Sportz Diesel | 2010 | 225000 | 127000 | Diesel | Individual | Manual | First Owner | 23.0 kmpl | |
| **4** | Maruti Swift VXI BSIII | 2007 | 130000 | 120000 | Petrol | Individual | Manual | First Owner | 16.1 kmpl | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8123** | Hyundai i20 Magna | 2013 | 320000 | 110000 | Petrol | Individual | Manual | First Owner | 18.5 kmpl | |
| **8124** | Hyundai Verna CRDi SX | 2007 | 135000 | 119000 | Diesel | Individual | Manual | Fourth & Above Owner | 16.8 kmpl | |
| **8125** | Maruti Swift Dzire ZDi | 2009 | 382000 | 120000 | Diesel | Individual | Manual | First Owner | 19.3 kmpl | |
| **8126** | Tata Indigo CR4 | 2013 | 290000 | 25000 | Diesel | Individual | Manual | First Owner | 23.57 kmpl | |
| **8127** | Tata Indigo CR4 | 2013 | 290000 | 25000 | Diesel | Individual | Manual | First Owner | 23.57 kmpl | |

8128 rows × 13 columns

In [7]:

```
car_df.columns
```

Out[7]:

```
Index(['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
       'transmission', 'owner', 'mileage', 'engine', 'max_power', 'torque',
       'seats'],
      dtype='object')
```

In [8]:

```
car_df.isnull().sum()
```

Out[8]:

```
name               0
year               0
selling_price      0
km_driven          0
fuel               0
seller_type        0
transmission       0
owner              0
mileage          221
engine           221
max_power        215
torque           222
seats            221
dtype: int64
```

In [9]:

```
car_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   name           8128 non-null   object 
 1   year           8128 non-null   int64  
 2   selling_price  8128 non-null   int64  
 3   km_driven      8128 non-null   int64  
 4   fuel           8128 non-null   object 
 5   seller_type    8128 non-null   object 
 6   transmission   8128 non-null   object 
 7   owner          8128 non-null   object 
 8   mileage        7907 non-null   object 
 9   engine         7907 non-null   object 
 10  max_power      7913 non-null   object 
 11  torque         7906 non-null   object 
 12  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 825.6+ KB
```

In [10]:

```
car_df.describe()
```

Out[10]:

| | year | selling_price | km_driven | seats |
|---|---|---|---|---|
| **count** | 8128.000000 | 8.128000e+03 | 8.128000e+03 | 7907.000000 |
| **mean** | 2013.804011 | 6.382718e+05 | 6.981951e+04 | 5.416719 |
| **std** | 4.044249 | 8.062534e+05 | 5.655055e+04 | 0.959588 |
| **min** | 1983.000000 | 2.999900e+04 | 1.000000e+00 | 2.000000 |
| **25%** | 2011.000000 | 2.549990e+05 | 3.500000e+04 | 5.000000 |
| **50%** | 2015.000000 | 4.500000e+05 | 6.000000e+04 | 5.000000 |
| **75%** | 2017.000000 | 6.750000e+05 | 9.800000e+04 | 5.000000 |
| **max** | 2020.000000 | 1.000000e+07 | 2.360457e+06 | 14.000000 |

In [11]:

```
car_df1=car_df.copy()
```

# FILLING THE MISSING VALUES

In [12]:
```
car_df.isnull().sum()
```

Out[12]:
```
name              0
year              0
selling_price     0
km_driven         0
fuel              0
seller_type       0
transmission      0
owner             0
mileage         221
engine          221
max_power       215
torque          222
seats           221
dtype: int64
```

In [14]:
```
car_df.fillna(method="ffill",inplace=True)
```

In [15]:
```
car_df.isnull().sum()
```

Out[15]:
```
name             0
year             0
selling_price    0
km_driven        0
fuel             0
seller_type      0
transmission     0
owner            0
mileage          0
engine           0
max_power        0
torque           0
seats            0
dtype: int64
```

# EDA(Exploratort Data Analysis)

In [16]:
```
car_df.name.value_counts()
```

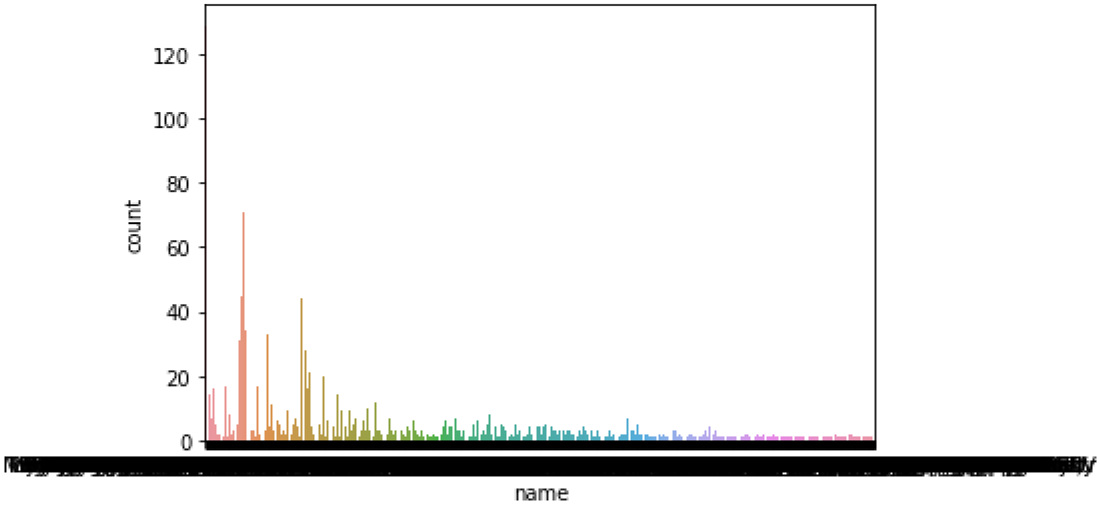
Out[16]:
```
Maruti Swift Dzire VDI                    129
Maruti Alto 800 LXI                        82
Maruti Alto LXi                            71
BMW X4 M Sport X xDrive20d                 62
Maruti Swift VDI                           61
                                         ... 
Ford Ecosport 1.5 Petrol Titanium           1
Hyundai Tucson 2.0 e-VGT 2WD AT GLS         1
Maruti Ertiga VXI CNG Limited Edition       1
Tata Indica V2 DLG TC                       1
Audi A4 2.0 TDI 177 Bhp Premium Plus        1
Name: name, Length: 2058, dtype: int64
```

In [17]:
```
sns.countplot(car_df.name)
```

```
C:\Users\Lenovvo\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pas
s the following variable as a keyword arg: x. From version 0.12, the only valid positional
argument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  warnings.warn(
```

Out[17]:
```
<AxesSubplot:xlabel='name', ylabel='count'>
```

Loading [MathJax]/extensions/Safe.js



In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

In [7]:

```
car_df["yrs"]=2020
```

In [8]:

```
car_df["year_old"]=car_df["yrs"]-car_df["year"]
```

In [9]:

```
car_df
```

Out[9]:

| | name | year | selling_price | km_driven | fuel | seller_type | transmission | owner | yrs | year_c |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Maruti 800 AC | 2007 | 60000 | 70000 | Petrol | Individual | Manual | First Owner | 2020 | |
| **1** | Maruti Wagon R LXI Minor | 2007 | 135000 | 50000 | Petrol | Individual | Manual | First Owner | 2020 | |
| **2** | Hyundai Verna 1.6 SX | 2012 | 600000 | 100000 | Diesel | Individual | Manual | First Owner | 2020 | |
| **3** | Datsun RediGO T Option | 2017 | 250000 | 46000 | Petrol | Individual | Manual | First Owner | 2020 | |
| **4** | Honda Amaze VX i-DTEC | 2014 | 450000 | 141000 | Diesel | Individual | Manual | Second Owner | 2020 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **4335** | Hyundai i20 Magna 1.4 CRDi (Diesel) | 2014 | 409999 | 80000 | Diesel | Individual | Manual | Second Owner | 2020 | |
| **4336** | Hyundai i20 Magna 1.4 CRDi | 2014 | 409999 | 80000 | Diesel | Individual | Manual | Second Owner | 2020 | |

| | name | year | selling_price | km_driven | fuel | seller_type | transmission | owner | yrs | year_ |
|---|---|---|---|---|---|---|---|---|---|---|
| **4337** | Maruti 800 AC BSIII | 2009 | 110000 | 83000 | Petrol | Individual | Manual | Second Owner | 2020 | |
| **4338** | Hyundai Creta 1.6 CRDi SX Option | 2016 | 865000 | 90000 | Diesel | Individual | Manual | First Owner | 2020 | |
| **4339** | Renault KWID RXT | 2016 | 225000 | 40000 | Petrol | Individual | Manual | First Owner | 2020 | |

4340 rows × 10 columns

```
In [10]: car_df.drop("yrs",axis=1)
```

Out[10]:

| | name | year | selling_price | km_driven | fuel | seller_type | transmission | owner | year_old |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Maruti 800 AC | 2007 | 60000 | 70000 | Petrol | Individual | Manual | First Owner | 13 |
| **1** | Maruti Wagon R LXI Minor | 2007 | 135000 | 50000 | Petrol | Individual | Manual | First Owner | 13 |
| **2** | Hyundai Verna 1.6 SX | 2012 | 600000 | 100000 | Diesel | Individual | Manual | First Owner | 8 |
| **3** | Datsun RediGO T Option | 2017 | 250000 | 46000 | Petrol | Individual | Manual | First Owner | 3 |
| **4** | Honda Amaze VX i-DTEC | 2014 | 450000 | 141000 | Diesel | Individual | Manual | Second Owner | 6 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4335** | Hyundai i20 Magna 1.4 CRDi (Diesel) | 2014 | 409999 | 80000 | Diesel | Individual | Manual | Second Owner | 6 |
| **4336** | Hyundai i20 Magna 1.4 CRDi | 2014 | 409999 | 80000 | Diesel | Individual | Manual | Second Owner | 6 |
| **4337** | Maruti 800 AC BSIII | 2009 | 110000 | 83000 | Petrol | Individual | Manual | Second Owner | 11 |
| **4338** | Hyundai Creta 1.6 CRDi SX Option | 2016 | 865000 | 90000 | Diesel | Individual | Manual | First Owner | 4 |
| **4339** | Renault KWID RXT | 2016 | 225000 | 40000 | Petrol | Individual | Manual | First Owner | 4 |

4340 rows × 9 columns

```
In [11]: car_df["fuel"].unique()
```

```
Out[11]: array(['Petrol', 'Diesel', 'CNG', 'LPG', 'Electric'], dtype=object)
```

```
In [12]: d_fuel=pd.get_dummies(car_df["fuel"])
```

```
In [13]: d_fuel
```

Out[13]:

| | CNG | Diesel | Electric | LPG | Petrol |
|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 |
| **1** | 0 | 0 | 0 | 0 | 1 |
| **2** | 0 | 1 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 1 |
| **4** | 0 | 1 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **4335** | 0 | 1 | 0 | 0 | 0 |
| **4336** | 0 | 1 | 0 | 0 | 0 |
| **4337** | 0 | 0 | 0 | 0 | 1 |
| **4338** | 0 | 1 | 0 | 0 | 0 |
| **4339** | 0 | 0 | 0 | 0 | 1 |

4340 rows × 5 columns

```
In [14]: d_fuel.columns=['Petrol', 'Diesel', 'CNG', 'LPG', 'Electric']
```

```
In [15]: d_fuel
```

Out[15]:

| | Petrol | Diesel | CNG | LPG | Electric |
|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 |
| **1** | 0 | 0 | 0 | 0 | 1 |
| **2** | 0 | 1 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 1 |
| **4** | 0 | 1 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **4335** | 0 | 1 | 0 | 0 | 0 |
| **4336** | 0 | 1 | 0 | 0 | 0 |
| **4337** | 0 | 0 | 0 | 0 | 1 |
| **4338** | 0 | 1 | 0 | 0 | 0 |
| **4339** | 0 | 0 | 0 | 0 | 1 |

4340 rows × 5 columns

```
In [16]: car_df=car_df.drop("yrs",axis=1)
```

Loading [MathJax]/extensions/Safe.js

In [17]:
```
car_df
```

Out[17]:

| | name | year | selling_price | km_driven | fuel | seller_type | transmission | owner | year_old |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Maruti 800 AC | 2007 | 60000 | 70000 | Petrol | Individual | Manual | First Owner | 13 |
| **1** | Maruti Wagon R LXI Minor | 2007 | 135000 | 50000 | Petrol | Individual | Manual | First Owner | 13 |
| **2** | Hyundai Verna 1.6 SX | 2012 | 600000 | 100000 | Diesel | Individual | Manual | First Owner | 8 |
| **3** | Datsun RediGO T Option | 2017 | 250000 | 46000 | Petrol | Individual | Manual | First Owner | 3 |
| **4** | Honda Amaze VX i-DTEC | 2014 | 450000 | 141000 | Diesel | Individual | Manual | Second Owner | 6 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4335** | Hyundai i20 Magna 1.4 CRDi (Diesel) | 2014 | 409999 | 80000 | Diesel | Individual | Manual | Second Owner | 6 |
| **4336** | Hyundai i20 Magna 1.4 CRDi | 2014 | 409999 | 80000 | Diesel | Individual | Manual | Second Owner | 6 |
| **4337** | Maruti 800 AC BSIII | 2009 | 110000 | 83000 | Petrol | Individual | Manual | Second Owner | 11 |
| **4338** | Hyundai Creta 1.6 CRDi SX Option | 2016 | 865000 | 90000 | Diesel | Individual | Manual | First Owner | 4 |
| **4339** | Renault KWID RXT | 2016 | 225000 | 40000 | Petrol | Individual | Manual | First Owner | 4 |

4340 rows × 9 columns

In [18]:
```
car_df=car_df.drop("name",axis=1)
```

In [19]:
```
car_df
```

Out[19]:

| | year | selling_price | km_driven | fuel | seller_type | transmission | owner | year_old |
|---|---|---|---|---|---|---|---|---|
| **0** | 2007 | 60000 | 70000 | Petrol | Individual | Manual | First Owner | 13 |
| **1** | 2007 | 135000 | 50000 | Petrol | Individual | Manual | First Owner | 13 |
| **2** | 2012 | 600000 | 100000 | Diesel | Individual | Manual | First Owner | 8 |
| **3** | 2017 | 250000 | 46000 | Petrol | Individual | Manual | First Owner | 3 |
| **4** | 2014 | 450000 | 141000 | Diesel | Individual | Manual | Second Owner | 6 |

| | year | selling_price | km_driven | fuel | seller_type | transmission | owner | year_old |
|---|---|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4335** | 2014 | 409999 | 80000 | Diesel | Individual | Manual | Second Owner | 6 |
| **4336** | 2014 | 409999 | 80000 | Diesel | Individual | Manual | Second Owner | 6 |
| **4337** | 2009 | 110000 | 83000 | Petrol | Individual | Manual | Second Owner | 11 |
| **4338** | 2016 | 865000 | 90000 | Diesel | Individual | Manual | First Owner | 4 |
| **4339** | 2016 | 225000 | 40000 | Petrol | Individual | Manual | First Owner | 4 |

4340 rows × 8 columns

In [20]:
```
car_df=pd.concat([d_fuel,car_df],axis=1)
```

In [21]:
```
car_df
```

Out[21]:

| | Petrol | Diesel | CNG | LPG | Electric | year | selling_price | km_driven | fuel | seller_type | transmi |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 | 2007 | 60000 | 70000 | Petrol | Individual | M |
| **1** | 0 | 0 | 0 | 0 | 1 | 2007 | 135000 | 50000 | Petrol | Individual | M |
| **2** | 0 | 1 | 0 | 0 | 0 | 2012 | 600000 | 100000 | Diesel | Individual | M |
| **3** | 0 | 0 | 0 | 0 | 1 | 2017 | 250000 | 46000 | Petrol | Individual | M |
| **4** | 0 | 1 | 0 | 0 | 0 | 2014 | 450000 | 141000 | Diesel | Individual | M |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **4335** | 0 | 1 | 0 | 0 | 0 | 2014 | 409999 | 80000 | Diesel | Individual | M |
| **4336** | 0 | 1 | 0 | 0 | 0 | 2014 | 409999 | 80000 | Diesel | Individual | M |
| **4337** | 0 | 0 | 0 | 0 | 1 | 2009 | 110000 | 83000 | Petrol | Individual | M |
| **4338** | 0 | 1 | 0 | 0 | 0 | 2016 | 865000 | 90000 | Diesel | Individual | M |
| **4339** | 0 | 0 | 0 | 0 | 1 | 2016 | 225000 | 40000 | Petrol | Individual | M |

4340 rows × 13 columns

In [22]:
```
car_df=car_df.drop("fuel",axis=1)
```

In [23]:
```
car_df
```

Out[23]:

| | Petrol | Diesel | CNG | LPG | Electric | year | selling_price | km_driven | seller_type | transmission |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 | 2007 | 60000 | 70000 | Individual | Manual |
| **1** | 0 | 0 | 0 | 0 | 1 | 2007 | 135000 | 50000 | Individual | Manual |
| **2** | 0 | 1 | 0 | 0 | 0 | 2012 | 600000 | 100000 | Individual | Manual |

Loading [MathJax]/extensions/Safe.js

| | Petrol | Diesel | CNG | LPG | Electric | year | selling_price | km_driven | seller_type | transmission |
|---|---|---|---|---|---|---|---|---|---|---|
| **3** | 0 | 0 | 0 | 0 | 1 | 2017 | 250000 | 46000 | Individual | Manual |
| **4** | 0 | 1 | 0 | 0 | 0 | 2014 | 450000 | 141000 | Individual | Manual |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4335** | 0 | 1 | 0 | 0 | 0 | 2014 | 409999 | 80000 | Individual | Manual |
| **4336** | 0 | 1 | 0 | 0 | 0 | 2014 | 409999 | 80000 | Individual | Manual |
| **4337** | 0 | 0 | 0 | 0 | 1 | 2009 | 110000 | 83000 | Individual | Manual |
| **4338** | 0 | 1 | 0 | 0 | 0 | 2016 | 865000 | 90000 | Individual | Manual |
| **4339** | 0 | 0 | 0 | 0 | 1 | 2016 | 225000 | 40000 | Individual | Manual |

4340 rows × 12 columns

```
In [24]: car_df=pd.get_dummies(car_df,drop_first=True)
```

```
In [25]: import seaborn as sns
```

```
In [26]: import matplotlib.pyplot as plt
```

```
In [27]: car_df=car_df.drop("year",axis=1)
```

```
In [28]: car_df
```

Out[28]:

| | Petrol | Diesel | CNG | LPG | Electric | selling_price | km_driven | year_old | seller_type_Individual |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 | 60000 | 70000 | 13 | 1 |
| **1** | 0 | 0 | 0 | 0 | 1 | 135000 | 50000 | 13 | 1 |
| **2** | 0 | 1 | 0 | 0 | 0 | 600000 | 100000 | 8 | 1 |
| **3** | 0 | 0 | 0 | 0 | 1 | 250000 | 46000 | 3 | 1 |
| **4** | 0 | 1 | 0 | 0 | 0 | 450000 | 141000 | 6 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4335** | 0 | 1 | 0 | 0 | 0 | 409999 | 80000 | 6 | 1 |
| **4336** | 0 | 1 | 0 | 0 | 0 | 409999 | 80000 | 6 | 1 |
| **4337** | 0 | 0 | 0 | 0 | 1 | 110000 | 83000 | 11 | 1 |
| **4338** | 0 | 1 | 0 | 0 | 0 | 865000 | 90000 | 4 | 1 |
| **4339** | 0 | 0 | 0 | 0 | 1 | 225000 | 40000 | 4 | 1 |

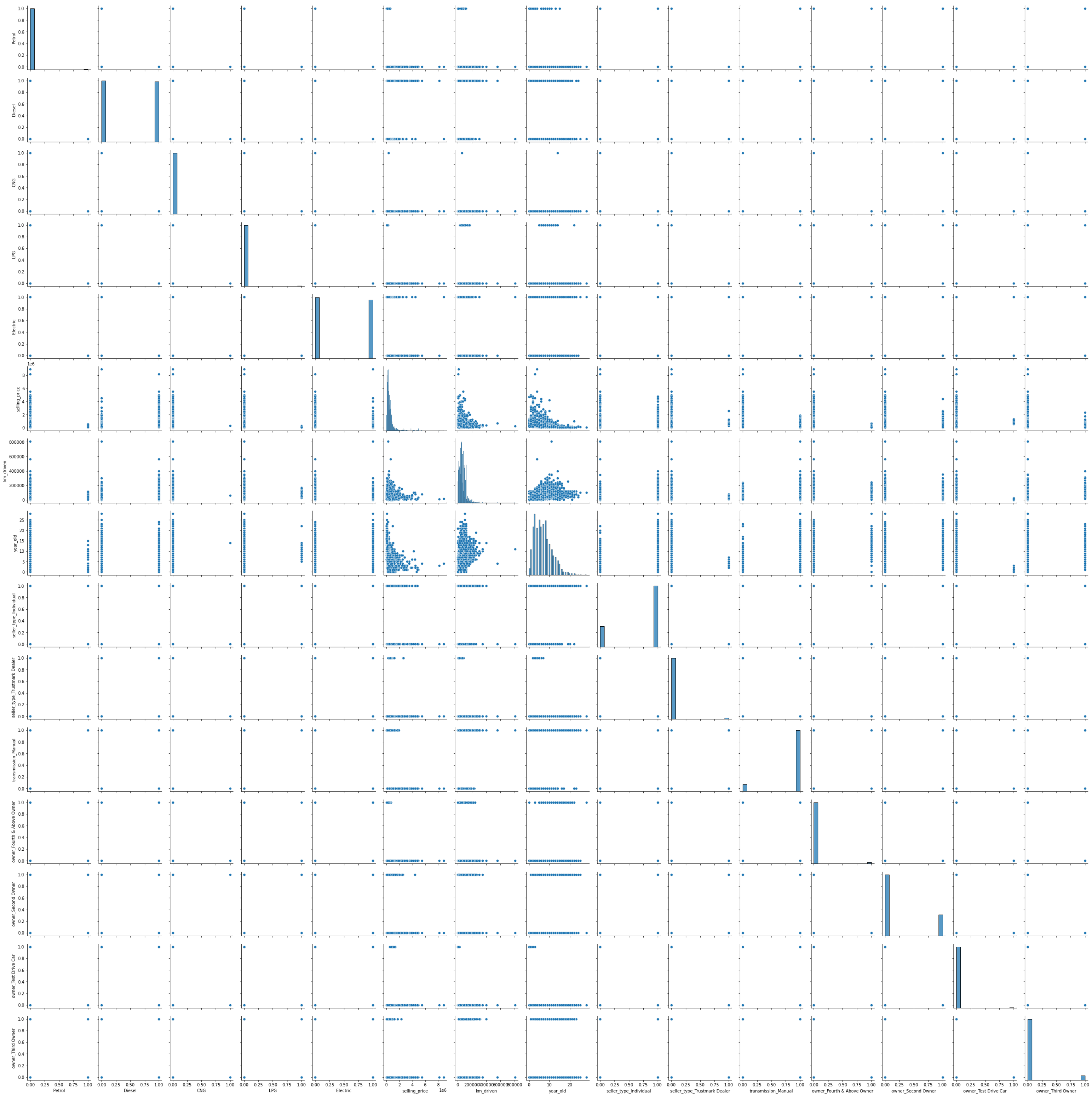
4340 rows × 15 columns

```
In [29]: #VISUALISATION
```

```
In [30]: sns.pairplot(car_df)
```

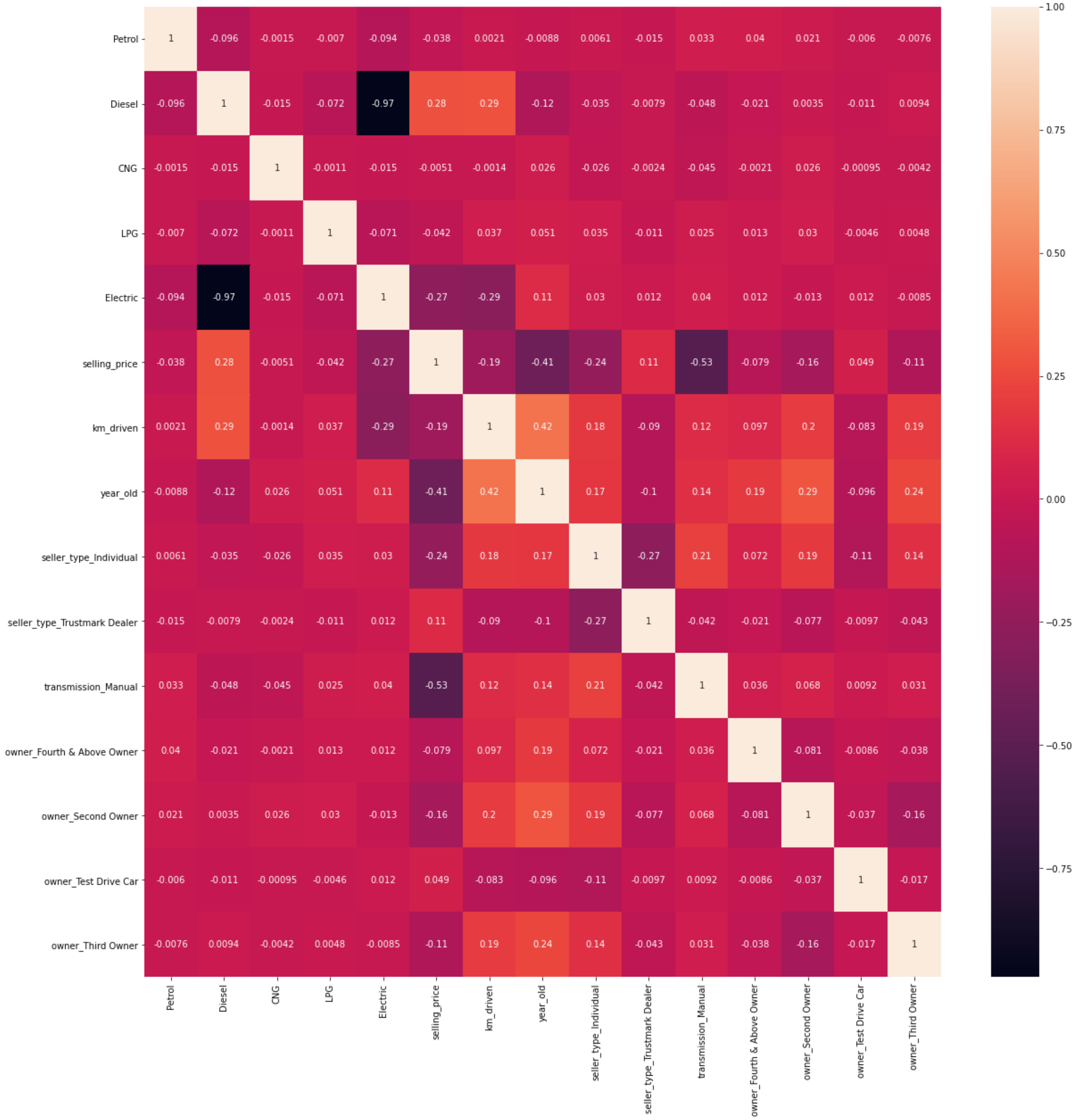
Loading [MathJax]/extensions/Safe.js

```
Out[30]: <seaborn.axisgrid.PairGrid at 0x24f96b0c250>
```



```
In [31]: corrmat=car_df.corr()
         top_corr_feature=cormat.index
         plt.figure(figsize=(20,20))
         heatmap=sns.heatmap(car_df[top_corr_feature].corr(),annot=True)
```

Loading [MathJax]/extensions/Safe.js



| | Petrol | Diesel | CNG | LPG | Electric | selling_price | km_driven | year_old | seller_type_Individual | seller_type_Trustmark Dealer | transmission_Manual | owner_Fourth & Above Owner | owner_Second Owner | owner_Test Drive Car | owner_Third Owner |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Petrol | 1 | -0.096 | -0.0015 | -0.007 | -0.094 | -0.038 | 0.0021 | -0.0088 | 0.0061 | -0.015 | 0.033 | 0.04 | 0.021 | -0.006 | -0.0076 |
| Diesel | -0.096 | 1 | -0.015 | -0.072 | -0.97 | 0.28 | 0.29 | -0.12 | -0.035 | -0.0079 | -0.048 | -0.021 | 0.0035 | -0.011 | 0.0094 |
| CNG | -0.0015 | -0.015 | 1 | -0.0011 | -0.015 | -0.0051 | -0.0014 | 0.026 | -0.026 | -0.0024 | -0.045 | -0.0021 | 0.026 | -0.00095 | -0.0042 |
| LPG | -0.007 | -0.072 | -0.0011 | 1 | -0.071 | -0.042 | 0.037 | 0.051 | 0.035 | -0.011 | 0.025 | 0.013 | 0.03 | -0.0046 | 0.0048 |
| Electric | -0.094 | -0.97 | -0.015 | -0.071 | 1 | -0.27 | -0.29 | 0.11 | 0.03 | 0.012 | 0.04 | 0.012 | -0.013 | 0.012 | -0.0085 |
| selling_price | -0.038 | 0.28 | -0.0051 | -0.042 | -0.27 | 1 | -0.19 | -0.41 | -0.24 | 0.11 | -0.53 | -0.079 | -0.16 | 0.049 | -0.11 |
| km_driven | 0.0021 | 0.29 | -0.0014 | 0.037 | -0.29 | -0.19 | 1 | 0.42 | 0.18 | -0.09 | 0.12 | 0.097 | 0.2 | -0.083 | 0.19 |
| year_old | -0.0088 | -0.12 | 0.026 | 0.051 | 0.11 | -0.41 | 0.42 | 1 | 0.17 | -0.1 | 0.14 | 0.19 | 0.29 | -0.096 | 0.24 |
| seller_type_Individual | 0.0061 | -0.035 | -0.026 | 0.035 | 0.03 | -0.24 | 0.18 | 0.17 | 1 | -0.27 | 0.21 | 0.072 | 0.19 | -0.11 | 0.14 |
| seller_type_Trustmark Dealer | -0.015 | -0.0079 | -0.0024 | -0.011 | 0.012 | 0.11 | -0.09 | -0.1 | -0.27 | 1 | -0.042 | -0.021 | -0.077 | -0.0097 | -0.043 |
| transmission_Manual | 0.033 | -0.048 | -0.045 | 0.025 | 0.04 | -0.53 | 0.12 | 0.14 | 0.21 | -0.042 | 1 | 0.036 | 0.068 | 0.0092 | 0.031 |
| owner_Fourth & Above Owner | 0.04 | -0.021 | -0.0021 | 0.013 | 0.012 | -0.079 | 0.097 | 0.19 | 0.072 | -0.021 | 0.036 | 1 | -0.081 | -0.0086 | -0.038 |
| owner_Second Owner | 0.021 | 0.0035 | 0.026 | 0.03 | -0.013 | -0.16 | 0.2 | 0.29 | 0.19 | -0.077 | 0.068 | -0.081 | 1 | -0.037 | -0.16 |
| owner_Test Drive Car | -0.006 | -0.011 | -0.00095 | -0.0046 | 0.012 | 0.049 | -0.083 | -0.096 | -0.11 | -0.0097 | 0.0092 | -0.0086 | -0.037 | 1 | -0.017 |
| owner_Third Owner | -0.0076 | 0.0094 | -0.0042 | 0.0048 | -0.0085 | -0.11 | 0.19 | 0.24 | 0.14 | -0.043 | 0.031 | -0.038 | -0.16 | -0.017 | 1 |

```
In [32]: x=car_df.drop("selling_price",axis=1)
```

```
In [33]: y=car_df["selling_price"]
```

```
In [34]: from sklearn.model_selection import train_test_split
```

```
In [35]: xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.20)
```

```
In [36]: xtrain.shape
```

```
Out[36]: (3472, 14)
```

```
In [37]: xtest.shape
```

```
Out[37]: (868, 14)
```

In [38]:
```
ytrain.shape
```
Out[38]:
```
(3472,)
```
In [39]:
```
ytest.shape
```
Out[39]:
```
(868,)
```
In [40]:
```
from sklearn.ensemble import ExtraTreesRegressor
```
In [41]:
```
model=ExtraTreesRegressor()
model.fit(x,y)
```
Out[41]:
```
ExtraTreesRegressor()
```
In [42]:
```
model.feature_importances_
```
Out[42]:
```
array([8.12481690e-05, 1.30422168e-01, 5.06684257e-05, 2.40238916e-05,
       1.37366507e-04, 2.49359762e-01, 2.44959990e-01, 4.69002285e-02,
       8.08621887e-03, 2.92827731e-01, 7.38430053e-04, 2.15933045e-02,
       5.52824715e-04, 4.26603636e-03])
```
In [43]:
```
from sklearn.ensemble import RandomForestRegressor
```
In [44]:
```
mod1=RandomForestRegressor(n_estimators=100,criterion='mse',
    max_depth=None,
    min_samples_split=2,
    min_samples_leaf=1,
    min_weight_fraction_leaf=0.0,
    max_features='auto',
    max_leaf_nodes=None,
    min_impurity_decrease=0.0,
    min_impurity_split=None,
    bootstrap=True,
    oob_score=False,
    n_jobs=None,
    random_state=None,
    verbose=0,
    warm_start=False,
    ccp_alpha=0.0,
    max_samples=None,)
```
In [45]:
```
mod1.fit(xtrain,ytrain)
```
Out[45]:
```
RandomForestRegressor()
```
In [52]:
```
from sklearn.model_selection import RandomizedSearchCV
```
In [88]:
```
random_grid={"n_estimators":[100,200,300,400,500,600,700,800,900,1000,1200],
             "max_features":["auto","sqrt"],
             "max_depth":[5,10,15,20,25,30],
            "min_samples_split":[2,15,20,100],
             "min_samples_leaf":[1,3,5,10]}
```
In [89]:
```
rf=RandomForestRegressor()
```
In [90]:
```
rf_random=RandomizedSearchCV(estimator=rf,param_distributions=random_grid,cv=5,scoring="n
```
In [91]:
```
rf_random.fit(xtrain,ytrain)
```
Loading [MathJax]/extensions/Safe.js

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV] n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_d
epth=15
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[CV]  n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_
depth=15, total=   0.9s
[CV] n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_d
epth=15
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.8s remaining:    0.0s
[CV]  n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_
depth=15, total=   0.9s
[CV] n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_d
epth=15
[CV]  n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_
depth=15, total=   0.9s
[CV] n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_d
epth=15
[CV]  n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_
depth=15, total=   1.0s
[CV] n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_d
epth=15
[CV]  n_estimators=300, min_samples_split=20, min_samples_leaf=10, max_features=auto, max_
depth=15, total=   1.2s
[CV] n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_de
pth=20
[CV]  n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_d
epth=20, total=   2.1s
[CV] n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_de
pth=20
[CV]  n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_d
epth=20, total=   2.1s
[CV] n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_de
pth=20
[CV]  n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_d
epth=20, total=   2.1s
[CV] n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_de
pth=20
[CV]  n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_d
epth=20, total=   1.8s
[CV] n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_de
pth=20
[CV]  n_estimators=800, min_samples_split=15, min_samples_leaf=3, max_features=sqrt, max_d
epth=20, total=   1.6s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=20
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=20, total=   3.8s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=20
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=20, total=   4.1s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=20
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=20, total=   3.9s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=20
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=20, total=   3.0s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=20
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=20, total=   4.1s
[CV] n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_d
epth=20
[CV]  n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_
depth=20, total=   2.5s
[CV] n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_d
epth=20
```

```
[CV]  n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_
depth=20, total=   2.5s
[CV] n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_d
epth=20
[CV]  n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_
depth=20, total=   2.5s
[CV] n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_d
epth=20
[CV]  n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_
depth=20, total=   2.5s
[CV] n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_d
epth=20
[CV]  n_estimators=700, min_samples_split=100, min_samples_leaf=1, max_features=auto, max_
depth=20, total=   2.5s
[CV] n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_de
pth=25
[CV]  n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_d
epth=25, total=   3.5s
[CV] n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_de
pth=25
[CV]  n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_d
epth=25, total=   3.5s
[CV] n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_de
pth=25
[CV]  n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_d
epth=25, total=   3.5s
[CV] n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_de
pth=25
[CV]  n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_d
epth=25, total=   3.5s
[CV] n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_de
pth=25
[CV]  n_estimators=1200, min_samples_split=2, min_samples_leaf=3, max_features=sqrt, max_d
epth=25, total=   3.4s
[CV] n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_de
pth=10
[CV]  n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_d
epth=10, total=   2.1s
[CV] n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_de
pth=10
[CV]  n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_d
epth=10, total=   2.1s
[CV] n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_de
pth=10
[CV]  n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_d
epth=10, total=   2.2s
[CV] n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_de
pth=10
[CV]  n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_d
epth=10, total=   2.1s
[CV] n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_de
pth=10
[CV]  n_estimators=500, min_samples_split=20, min_samples_leaf=5, max_features=auto, max_d
epth=10, total=   2.1s
[CV] n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_dep
th=20
[CV]  n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_de
pth=20, total=   3.2s
[CV] n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_dep
th=20
[CV]  n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_de
pth=20, total=   3.2s
[CV] n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_dep
th=20
[CV]  n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_de
pth=20, total=   3.2s
[CV] n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_dep
th=20
[CV]  n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_de
                 3.2s
```

Loading [MathJax]/extensions/Safe.js

```
[CV] n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_dep
th=20
[CV]  n_estimators=700, min_samples_split=2, min_samples_leaf=1, max_features=sqrt, max_de
pth=20, total=   3.2s
[CV] n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_d
epth=5
[CV]  n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_
depth=5, total=   0.2s
[CV] n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_d
epth=5
[CV]  n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_
depth=5, total=   0.2s
[CV] n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_d
epth=5
[CV]  n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_
depth=5, total=   0.2s
[CV] n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_d
epth=5
[CV]  n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_
depth=5, total=   0.2s
[CV] n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_d
epth=5
[CV]  n_estimators=100, min_samples_split=20, min_samples_leaf=10, max_features=sqrt, max_
depth=5, total=   0.2s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=25
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=25, total=   4.0s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=25
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=25, total=   4.0s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=25
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=25, total=   4.0s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=25
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=25, total=   4.0s
[CV] n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_de
pth=25
[CV]  n_estimators=900, min_samples_split=20, min_samples_leaf=3, max_features=auto, max_d
epth=25, total=   4.0s
[CV] n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max_
depth=5
[CV]  n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max
_depth=5, total=   3.1s
[CV] n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max_
depth=5
[CV]  n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max
_depth=5, total=   3.4s
[CV] n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max_
depth=5
[CV]  n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max
_depth=5, total=   3.1s
[CV] n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max_
depth=5
[CV]  n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max
_depth=5, total=   3.1s
[CV] n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max_
depth=5
[CV]  n_estimators=1000, min_samples_split=100, min_samples_leaf=5, max_features=auto, max
_depth=5, total=   3.2s
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:  2.1min finished
```

```
Out[91]: RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(),
                            param_distributions={'max_depth': [5, 10, 15, 20, 25, 30],
                                                 'max_features': ['auto', 'sqrt'],
                                                 'min_samples_leaf': [1, 3, 5, 10],
                                                 'min_samples_split': [2, 15, 20, 100],
                                                 'n_estimators': [100, 200, 300, 400,
                                                                  500, 600, 700, 800,
                                                                  900, 1000, 1200]},
                            random_state=42, scoring='neg_mean_squared_error',
                            verbose=2)
```

In [92]:
```
predict=rf_random.predict(xtest)
```

In [93]:
```
predict
```

Out[93]:
```
array([ 202884.98681874,  714199.60884354,  126623.00486395,
        657087.77980114,  552850.28676769,  750000.        ,
        285263.94405198,  241293.32016407, 1314924.22857143,
        357335.59376674,  277984.53332281,  448821.94977015,
        543613.38761167,  310310.60244898,  348570.12746867,
        322232.44788687,  110173.83888889,  455632.44991993,
        194939.55988456,  123589.87956092,  235482.67687075,
         93488.65467065,   86133.51587302,  379185.81017663,
        316944.69234559,  750192.845     ,  203721.65938776,
        385164.84219142,  395717.13857143,  161334.3321746 ,
        319187.09673559,  812338.79746186,  570988.08102721,
        866326.2400822 ,  364430.79113724,  321801.84380952,
         91225.08219615,  757316.19047619,  564137.74935032,
        545915.49042084,  319187.09673559,  346705.10995573,
        538090.87061087,   92957.79177662,  552667.1077551 ,
        895480.22380952,  550991.78571429,  202595.23258867,
        322232.44788687,  100677.42573696,   89238.92857143,
        298316.40524404,  670000.        , 2600000.        ,
        238110.49838708,  630406.77724851,  420558.06542608,
        258839.75238095,  594233.15809754,  311889.06176871,
        714771.06575964,   83129.9187886 ,  355374.04761905,
        379011.74246456,  336486.00918526,   98157.38095238,
        370496.56172449,  292933.30666667,  109712.70680272,
        951718.44142857,  276045.13980584,  812338.79746186,
        297224.76190476,  244777.00736961,   62896.6037415 ,
        964540.17857143,  252290.05205112,  138015.32696875,
        339259.04519331,  600000.        ,  325887.38499596,
        282912.56328726,  158823.34486735,  326527.63911565,
        172902.16677123,  714254.95601732,  645027.84816327,
        448620.98901616,  316514.04067142,  288504.51809524,
        167486.07404028,  230272.48011114,  535956.32357657,
        417611.41632123,  352037.2226415 ,  461039.22589513,
        505309.23128834, 1254709.99857143,  648389.28428571,
        157384.52380952, 2092106.16047619,  449772.9257967 ,
        138015.32696875,  465784.1382292 ,  257075.32183147,
        103030.94952381,  424351.41316409,  652261.39719663,
        117396.65194187,  845033.68462585,  300049.79559266,
        761436.00896429,   89175.28207903,  205384.44355556,
        398912.97929118,   72845.92133311,  228664.16099773,
        512938.6739164 ,  142334.96835401,  260524.57596372,
        463814.57019707,  462708.60827664,  231523.15224562,
        554578.45322619,  674033.07770408,  195368.31934127,
        308457.36231781,  518141.69580952,  988768.52380952,
       2212129.70714286,  396701.81543539,  541429.55652647,
        535956.32357657,  319187.09673559,  410329.21201814,
        143796.13605442,  121757.45721446, 1300619.23469388,
        114167.26649255,  368016.23095238,  668056.42850007,
        450000.        ,  686285.35714286,  522789.52380952,
        547356.06845125,  379021.50804859,  495830.62639456,
        319187.09673559,  130268.0761797 ,  207828.43683777,
        475206.09363792, 1013782.53817844,  395733.79578633,
        450000.        ,  734896.84082526,  155074.74196769,
        132888.0952381 ,  603488.21428571,  155150.3487415 ,
        254672.16852381,  664617.19117429,  911734.29857143,
        694078.64322682, 1201782.85571429,  107541.80117702,
        417606.39060784,  155781.6723356 ,  355909.76190476,
        852540.77292857,  168970.51137829,  203868.57142857,
 Loading [MathJax]/extensions/Safe.js 0809524,   92661.61761905, 1322877.25602451,
        336244.7283843 ,  130417.65306122,  374900.2191075 ,
        391181.1347619 ,  588578.1075659 ,  346705.10995573,
        542496.34548336,  586782.61904762,  311312.51757043,
        732004.6247619 ,  385164.84219142,  109900.94310139,
        321000.67142857,  726171.57596372,  142334.96835401,
       1230930.6122449 , 1013377.15718696,  225782.30612389,
        226401.01883778,  556830.80539792,  429731.65120594,
        110740.80238095,  916765.86957774,  452170.80788259,
        257075.32183147,  797292.6063517 ,  120755.85714286,
         93488.65467065,  475206.09363792,  357335.59376674,
       3799712.85714286,  292004.06666086, 1347711.42428571,
        133364.44823384,  208358.34318937,  355653.75931624,
        228325.23107483, 3799712.85714286,  295383.41875416,
        301691.15857103,  370999.49433673,  111277.48701299,
        595113.77408163, 1217365.71428571,  853784.85047619,
        137479.47880859,  522192.51657143,  149392.9027222 ,
        290631.05002655, 1050248.35404675,  402405.91055462,
        111844.94761541,  250443.20751914,  120320.67255033,
         79237.22479901,   80822.85571429,  225255.92035623,
        103330.05436417, 2758341.64095238,  282044.52316366,
        389331.5434261 ,  305052.42582993,  498832.45103632,
        743920.09619048,  547686.405141  ,  463814.57019707,
        661232.48309627,  333721.25332931,   82865.04995815,
        357187.30111613,   76477.14285714,  239181.00033752,
        101064.80827664,  200936.39596515, 1738834.28142857,
        497044.08260755,  126821.32162865,  412133.71761849,
       1013377.15718696,  470590.29829644,  259665.82340696,
        165174.34739763,  912376.82789827,  381638.03916667,
        750000.        , 1013377.15718696,  454080.55938792,
        236242.37439273,  123677.9200261 ,  323200.35577262,
        309951.79131823,  200505.7105885 ,  912376.82789827,
        242528.09930942,  506527.69412391, 1032243.11920016,
        448620.98901616,  442313.64566238,  526449.30045351,
        379185.81017663,  153544.08182073,  421415.15898784,
        118976.75211205,  102871.43098117,  546646.22290249,
        299873.77669949, 1013782.53817844,  100284.05952381,
        115805.47512916,  490883.36198722,  187394.59897449,
        422567.82972317,  101600.58591012,  233302.24737938,
        678847.03123067,  365925.8787619 ,  424351.41316409,
        283669.14989487,  417606.39060784,  527127.69824919,
        166788.00448134,  239191.88642213,  158182.12042687,
       1004448.10023988,  214384.04875283, 1930178.4554731 ,
        491158.93120793,  678847.03123067,  647005.00332024,
         71102.14      ,  307148.67389614,  477739.93808163,
        308999.35901884,  583324.00520202,  292004.06666086,
        276100.61918067,  935648.60544218,  576123.61915069,
        424351.41316409,   74052.22222222,  241141.7502551 ,
        297504.84560356,  344291.5560401 ,  694078.64322682,
        659498.16927811,   83305.23809524,  496383.31774229,
        638544.14597006,  239829.0685017 ,  147849.27994024,
        709758.13952656,  130919.95748299,  240555.        ,
        463870.93653061,  390936.45590681,  385164.84219142,
        420558.06542608,  290371.98996032,  882890.9422449 ,
        150947.06100134,  502609.38764215,  343033.60723099,
        389140.23408719,  293211.23847163,  556701.90487605,
        125342.37666667,  316724.03861905,  305496.68199931,
        335317.69436508,  225782.30612389,  718304.17220146,
       1185635.24892857,  324347.08536681,  700963.71465729,
        617340.5594369 ,  245867.78452758,  348343.38029514,
        506453.94339105, 1193309.07029478,  585347.98529412,
       1013377.15718696,  111844.94761541,  372899.22350262,
        788685.34417027, 1164371.9047619 , 1013782.53817844,
        504347.73809524,  308999.35901884,  643518.17619048,
        829057.41371059,  213635.22666667,  341989.35492977,
        653516.68696323,  463814.57019707,  244777.00736961,
         78278.57142857,  165535.52993197,  623608.56009127,
        327132.4660102 ,  162932.14199844,  558909.06462585,
       1013392.86976706,  535956.32357657,  513955.91319203,
        314303.27025492,  320498.35004762,  464651.13790781,
               4246456,  421013.28941042,  440434.52095238,
```

Loading [MathJax]/extensions/Safe.js

```
 228509.09990822,   92957.79177662,  357335.59376674,
 329260.71428571,  304483.57142857,  533195.36228571,
 120795.67679607,   96115.254329  ,  527127.69824919,
 300049.79559266,  301654.06114966,  442313.64566238,
1300619.23469388,  916765.86957774,  911734.29857143,
 173069.77814286,  242414.68945667,   96115.254329  ,
 740256.53619048,  429640.64365365,  221125.23567663,
 512938.6739164 ,  405012.04113854,  198131.19587897,
 340354.20884547, 1645751.18904762,  687073.00574302,
 521931.57699443,  605236.85697268,  381933.65421769,
 515828.12642878,  247185.21032396,  158182.12042687,
 131693.74761224,  355924.97868407,  317137.1328065 ,
 623608.56009127,  202595.23258867,  102871.43098117,
 376687.9175213 ,  289560.32800247,  465784.1382292 ,
 283669.14989487,  268086.62405549,  628441.87146517,
 235492.7578458 ,  730761.65452381,  352707.36105958,
 556751.18769231,  190699.97714286,  967056.53764172,
 535870.09328785,  200936.39596515,  509221.40547619,
 225255.92035623,  804318.87646877,  906109.6425    ,
 188453.15564626,   62896.6037415 , 3799712.85714286,
 766483.62653955,  429640.64365365,   98493.10262214,
 172747.39512657,  285432.12510302,  290631.05002655,
 268086.62405549,  142334.96835401,  693473.113118  ,
 372899.22350262,  558467.11681176,  420558.06542608,
 607230.64316457,  182872.94124923,  316944.69234559,
 842766.99546485,  225655.02749504,  164740.18521954,
 145542.92052381,  287662.14285714,  922162.38095238,
 664617.19117429,  345837.38095238,  325885.8587897 ,
 112937.07807301, 1392225.71285714,   66366.56462585,
 611281.86888889,  125113.86571429,  601521.41714286,
 351426.09756263, 1113843.0952381 ,  125096.49904762,
 243510.04595238,   96264.89455782,  232400.60260153,
 253771.9047619 ,  195270.44758813,  150865.85067108,
 665881.32591837,  436358.52897888,  237323.0015856 ,
 366445.05251234,  175598.29646259,  309951.79131823,
 916765.86957774,  793857.44583643,  357362.37952381,
  59458.74006803,  276100.61918067,  285199.68202432,
 226239.20005877,  297585.97673418,  701437.62455741,
 197037.61904762,   97003.07653061,  609839.38208617,
 744636.74174603,  315580.87006803,  625000.        ,
 489394.17064796,   98160.96526093,  494545.51585034,
 556701.90487605,  750000.        ,  990980.40816327,
 529216.78287982,  372604.83402597,  465356.34199134,
  88223.2012987 , 3321338.56714286,  153736.18745855,
 463814.57019707,  421415.15898784,  455632.44991993,
 299689.69835059,  396740.47213215,  417606.39060784,
 236242.37439273,  295130.55412698, 1282032.87029762,
 675415.23809524,  174607.14285714,  111385.69285714,
 547686.405141  ,  455507.76688645,  734896.84082526,
 116129.25814265,  597854.99952381,  392028.24963456,
 140002.25599258,   96115.254329  ,  540156.72369749,
  99061.94244898,  766571.42      ,  344570.23347666,
 421013.28941042,  103330.05436417,  418477.17510981,
 239829.0685017 , 3799712.85714286,  299689.69835059,
 357463.74370424,  538793.33583327,   79640.40816327,
 814622.6444898 ,  670000.        , 1419562.85714286,
  92202.1547619 ,  384563.24635539,  516415.30330913,
  88294.21708792,  281584.5149662 ,  534343.77834467,
 641541.02690383,  405012.04113854,  287325.40361472,
1257424.25285714,  600418.43130933,  565579.09543528,
 360382.47188545,  526814.3393164 ,  149107.69752381,
 339259.04519331,  141648.97830822,  293467.53919453,
 417606.39060784,  156057.05598639, 1354040.20408163,
 369183.24256919,  238110.49838708,  348462.46107143,
 667592.15567246, 1013377.15718696,  362703.94189255,
  72845.92133311,  670677.68516866,  538621.05173409,
 699884.83611626,  332000.97376682, 1512512.26557823,
 142334.96835401,  431375.47923469,  558761.55311355,
 475206.09363792,  314936.85676866,  717065.7259168 ,
          1587302,  286016.48407318,  740375.06035744,
```

Loading [MathJax]/extensions/Safe.js

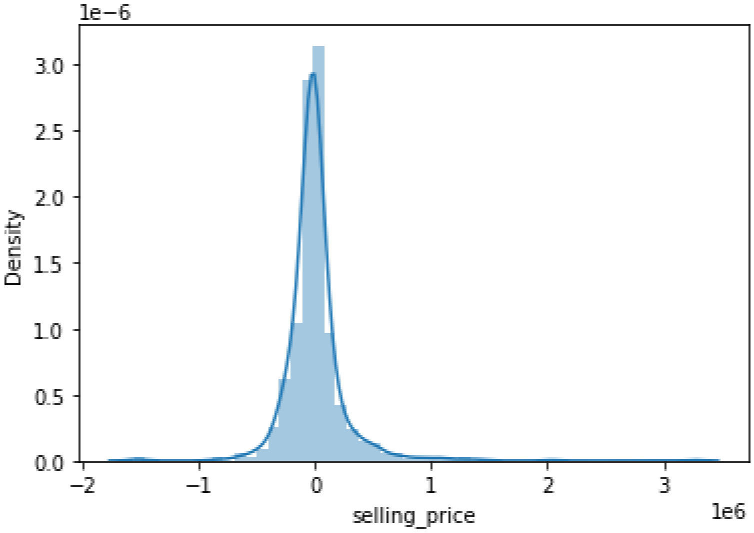
```
  337447.88086315,  799295.43882705,  520000.        ,
 1362030.70849313,  103520.45171315,   77424.44262108,
  362875.71428571,  271750.52721088,  455167.59452381,
  919397.2957398 ,  392028.24963456,  385164.84219142,
  197172.71322403,  692314.07342125,  532378.1402415 ,
  463814.57019707,  311312.51757043,   86133.51587302,
  336896.50861753,  323427.83810799,  268086.62405549,
  652261.39719663,  223040.77333617,  636878.19094288,
  250758.23064626,  336896.50861753,  101600.58591012,
  719353.26630314,  118849.99571429,  372639.99428571,
  307148.67389614,  124199.03612153,  190708.37092857,
  299266.82306958,  672651.01523735,  197172.71322403,
  766571.42       ,  605236.85697268,  537508.82962276,
 2628687.6984127 ,  196949.64285714,  324164.22989827,
  919397.2957398 ,  107525.18017642,  988932.85714286,
  657087.77980114,  265434.07725137,  199365.25998066,
  308457.36231781,  375101.5299124 ,   78278.57142857,
 1350174.50862629,  535166.02979925,   88294.21708792,
  860421.96092112,  647517.4312585 ,  872485.17982018,
  245759.22557932,  219202.51003596,  785805.28911565,
  461039.22589513,  244777.00736961,  951718.44142857,
  680997.18778218,  199365.25998066,  668342.95365808,
  799295.43882705,  343433.78169898,  219930.59839002,
  659204.15571429,  130329.42831512,  226017.12535818,
  225255.92035623,  137067.38626289,  104729.74783438,
  479789.1991342 ,  166827.25389251,  331134.90233825,
  526814.3393164 ,  458352.52380952,  547686.405141  ,
  693585.23857143, 2160141.12244898,  250758.23064626,
  244765.36831388,  429640.64365365, 1701435.71428571,
  399465.72345605,  551648.47685153,  512938.6739164 ,
  510597.97560905, 1181541.46315476,  911923.49903772,
  664700.54761905,  169419.39391327,  813962.53673782,
  465784.1382292 ,  750000.        ,  327132.4660102 ,
  740651.29620539,  692342.85714286,  148398.09380952,
  546637.59300224,  245759.22557932, 1540975.42517007,
  287662.14285714,  418477.17510981,  873427.34540476,
 1720218.0952381 ,  793857.44583643,  355924.97868407,
  257075.32183147,  181117.61814739,  290631.05002655,
  191749.69106576,  292004.06666086,  739036.5037534 ,
  156779.88492197,  428559.97352459,  379859.43722373,
  479991.01173197,  153180.67960269,  839749.73639456,
  143999.98241922,  231523.15224562, 1452641.40285714,
  339754.69954649,  123272.23666598,  292004.06666086,
  550991.78571429,  347598.12090348,  555331.30809524,
  420558.06542608,  197172.71322403, 2600000.        ,
  364033.08809524,  816359.36507937,  250422.76428571,
  299969.2902595 ,  167895.9568623 , 1004924.63409606,
  410329.21201814,  734896.84082526,  563023.63889027,
  741242.85714286,  262729.73637188, 3799712.85714286,
  209381.36285714,  187907.08604076,  420558.06542608,
  254672.16852381,  324306.20014853,  127772.77891156,
  465442.13857143,  331050.5174163 ,  807624.60547449,
  217576.84930374,  310231.37813522,  137293.3416226 ,
  405818.60679639,  910046.24271429, 1170521.41857143,
  252290.05205112, 4950000.        , 1300619.23469388,
  227941.27945012,   80000.        ,  202393.1455578 ,
  848891.93809524,  211442.68642857,  569621.15860692,
  121956.67800454,  363837.99315306,  116940.00755707,
  164740.18521954,  366586.43260824,   92957.79177662,
  125096.49904762,  521316.06235828,   92202.1547619 ,
  277984.53332281,  232152.97795918,  664617.19117429,
 1579767.32142857,  552850.28676769,  672651.01523735,
  877542.42494216,  933336.54297619,  490320.85871985,
  121469.00285714,  775249.7755102 ,  259219.96656573,
  196339.48744113,  301654.06114966, 2628687.6984127 ,
  450000.        ,  538621.05173409,  219202.51003596,
  417785.68571429,  250021.38       ,  424351.41316409,
  149650.4570329 , 1458635.        ,  761110.47571429,
  685729.88067714,   59553.56714286,  319187.09673559,
         7619048,  540596.07142857,  274671.80308726,
  245439.6435034 ,  279543.6399485 ,  555331.30809524,
  220875.09033707,  745769.63500379,  773468.45594295,
  123272.23666598,  414390.76705272,  139111.39455782,
  188670.86204889,  598670.        ,  388646.17254766,
  121757.45721446,   64770.82482993,  305829.17816327,
 2358876.19047619,  357335.59376674,  165174.34739763,
  476076.59837757,  540638.92857143,  276100.61918067,
  295906.71904762,  221077.93186588,  894364.19420643,
  299551.8547619 ,  426451.41955885,  279044.16666667,
  103520.45171315,  265434.07725137,  375211.56633065,
  287477.78290607,  172747.39512657, 1181541.46315476,
  471188.2888383 ,  230272.48011114,  563625.30845918,
  232062.8114966 ,  652261.39719663,  773468.45594295,
  240288.61381055,  101128.37301587,  168970.51137829,
  465356.34199134, 1461779.91571429,  440326.07340517,
  139811.11725895,  362126.95112724,  623608.56009127,
  194040.40163989,  385164.84219142,  154342.25881262,
  123589.87956092,  402569.5358414 ,  417606.39060784,
  461039.22589513,  231523.15224562, 1053613.28324998,
  207861.81703154,  376687.9175213 ,  352758.23593074,
  234816.0952381 ,  166702.55289116,  693473.113118  ,
  209623.76904762])
```

In [95]:

```
sns.distplot(ytest-predict)
```

Out[95]:

```
<AxesSubplot:xlabel='selling_price', ylabel='Density'>
```

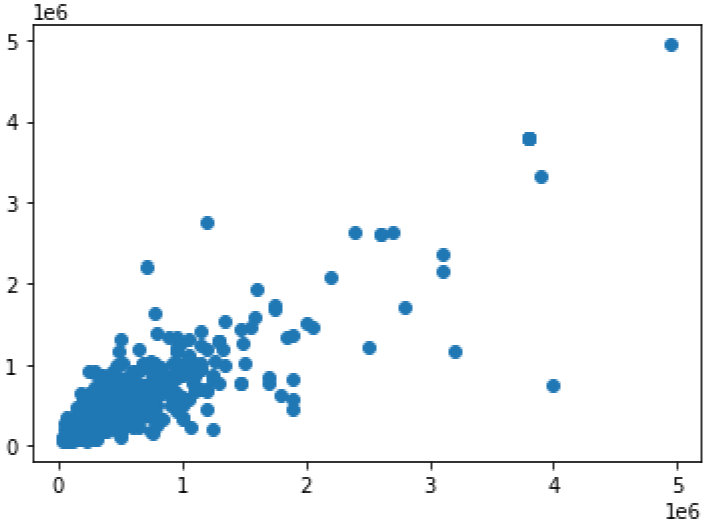


In [96]:

```
plt.scatter(ytest,predict)
```

Out[96]:

```
<matplotlib.collections.PathCollection at 0x24fb03f17f0>
```



Loading [MathJax]/extensions/Safe.js

```
In [97]:  import pickle
```

```
In [104…  rf_random.score(xtest,ytest)

Out[104…  -72224596634.23013
```

```
In [99]:  from sklearn.metrics import accuracy_score
```

```
In [101…  from sklearn.metrics import confusion_matrix
```

```
In [102…  rf_random.confusion_matrix(ytest,predict)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-102-062fbe660a3d> in <module>
----> 1 rf_random.confusion_matrix(ytest,predict)

AttributeError: 'RandomizedSearchCV' object has no attribute 'confusion_matrix'
```

```
In [106…  from sklearn.model_selection import cross_val_score
```

```
In [107…  from sklearn.linear_model import LinearRegression
```

```
In [113…  lr=LinearRegression()
```

```
In [122…  model=cross_val_score(lr,x,y,cv=20)
```

```
In [123…  model

Out[123…  array([0.43360053, 0.48901824, 0.4920512 , 0.37981326, 0.47238352,
                 0.46967994, 0.49587967, 0.52424644, 0.44157967, 0.35883599,
                 0.47944236, 0.42824479, 0.47808319, 0.09621042, 0.33942696,
                 0.40880557, 0.27924115, 0.31627039, 0.3583534 , 0.54628606])
```

```
In [126…  from sklearn.linear_model import Lasso
```

```
In [127…  l=Lasso()
```

```
In [128…  model1=cross_val_score(l,x,y,cv=10)
```

```
C:\Users\Lenovvo\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:5
29: ConvergenceWarning: Objective did not converge. You might want to increase the number
of iterations. Duality gap: 79821038078733.38, tolerance: 121813521599.65625
  model = cd_fast.enet_coordinate_descent(
C:\Users\Lenovvo\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:5
29: ConvergenceWarning: Objective did not converge. You might want to increase the number
of iterations. Duality gap: 78973652635441.0, tolerance: 120741105130.33423
  model = cd_fast.enet_coordinate_descent(
C:\Users\Lenovvo\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:5
29: ConvergenceWarning: Objective did not converge. You might want to increase the number
of iterations. Duality gap: 88699501840697.12, tolerance: 136616128310.62645
  model = cd_fast.enet_coordinate_descent(
C:\Users\Lenovvo\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:5
29: ConvergenceWarning: Objective did not converge. You might want to increase the number
of iterations. Duality gap: 281646978047103.4, tolerance: 136530550559.32599
  model = cd_fast.enet_coordinate_descent(
```

```
In [129…  model1

Out[129…  array([0.45463194, 0.49263852, 0.49558823, 0.50730906, 0.4233251 ,
                 0.47411384, 0.38126479, 0.38134635, 0.32829662, 0.4431634 ])
```

Loading [MathJax]/extensions/Safe.js

```
In [130… from sklearn.linear_model import Ridge

In [131… r=Ridge()

In [132… model3=cross_val_score(r,x,y,cv=10)

In [133… model3

Out[133… array([0.45443655, 0.49233964, 0.49513658, 0.50799087, 0.42394116,
       0.47399736, 0.38226262, 0.38223174, 0.32838997, 0.44203268])

In [136… from sklearn.preprocessing import StandardScaler

In [137… sc=StandardScaler()

In [139… train_sc=sc.fit_transform(xtrain)

In [150… test_sc=sc.transform(xtest)

In [141… from sklearn.linear_model import LinearRegression

In [142… lr1=LinearRegression()

In [143… lr1.fit(train_sc,ytrain)

Out[143… LinearRegression()

In [154… pred=lr1.predict(test_sc)

In [156… lr1.score(xtest,ytest)

Out[156… -1.278269461410842e+26

In [155… lr1.confusion_matrix(ytest,pred)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-155-07b3bb7017b2> in <module>
----> 1 lr1.confusion_matrix(ytest,pred)

AttributeError: 'LinearRegression' object has no attribute 'confusion_matrix'
```

```
In [157… mod=LinearRegression()

In [158… mod.fit(xtrain,ytrain)

Out[158… LinearRegression()

In [159… mod.predict(xtest)

Out[159… array([ 4.25006408e+05,  1.29435289e+06,  1.78735459e+05,  7.01992956e+05,
         4.27310291e+05,  8.72633181e+05,  4.87739478e+05,  2.49756840e+05,
         1.25854332e+06,  4.61128307e+05,  3.18997858e+05,  4.87739178e+05,
         4.83376805e+05,  2.91534259e+05,  4.63097082e+05,  4.16717797e+05,
        -2.82240885e+04,  5.32149688e+05,  1.16489747e+04, -5.10968243e+02,
         4.07016171e+05, -5.57753746e+04, -1.78151807e+05,  3.99006145e+05,
         4.07818278e+05,  8.47280855e+05,  2.22542196e+05,  6.01004597e+05,
         1.37486954e+06,  9.18078006e+04,  4.25617316e+05,  7.99712895e+05,
         6.30884189e+05,  7.81913856e+05,  4.59348403e+05,  4.85212405e+05,
                39e+04,  8.13952126e+05,  4.25529929e+05,  4.20277605e+05,
```

Loading [MathJax]/extensions/Safe.js

```
 4.25617316e+05,  5.24352118e+05,  7.92797280e+05, -1.94993428e+04,
 5.57094051e+05,  1.04568268e+06,  5.85372030e+05,  8.70327284e+04,
 4.16717797e+05, -8.56590876e+04, -8.11757390e+04,  2.46787142e+05,
 6.48561624e+05,  1.79362122e+06,  3.54421462e+05,  7.32367935e+05,
 5.18887495e+05,  4.87914252e+05,  5.23250168e+05,  3.62605503e+05,
 6.74105062e+05, -8.05192551e+04,  2.73485700e+05,  5.59862808e+05,
 4.82611764e+05, -6.96937507e+04,  5.14593679e+05,  3.55523411e+05,
-5.49226464e+04,  8.36824810e+05,  3.77914284e+05,  7.99712895e+05,
 3.81498794e+05,  3.27897378e+05, -7.99285293e+04,  1.45550094e+06,
 2.56788610e+05,  1.60113476e+04,  5.33339024e+05,  9.43180217e+05,
 4.78839658e+05,  3.72511588e+05,  1.44567923e+05,  3.37724373e+05,
 1.11061430e+05,  6.30876192e+05,  5.94271549e+05,  4.56678247e+05,
 3.98919059e+05,  2.77661744e+05,  3.37229994e+04,  3.08766454e+05,
 6.64168391e+05,  3.10098339e+05,  5.10025500e+05,  5.86473979e+05,
 6.60369216e+05,  1.67933748e+06,  5.22919897e+05,  1.38690207e+05,
 1.50513425e+06,  4.74477286e+05,  1.60113476e+04,  4.61128007e+05,
 2.56875997e+05,  8.73241156e+04,  4.97740946e+05,  6.57582747e+05,
-1.42553730e+05,  6.36213995e+05,  3.26795428e+05,  5.57539027e+05,
-1.05124367e+04,  8.88127901e+04,  4.64017882e+05, -7.27216849e+04,
 3.63050479e+05,  4.79941908e+05, -2.83988621e+04,  3.77772210e+05,
 4.70027526e+05,  1.28554076e+06,  1.40255071e+05,  6.06731034e+05,
 7.28516740e+05,  2.20350744e+05,  2.66665468e+05,  3.76844734e+05,
 1.30325271e+06,  1.41590328e+06,  6.27437778e+05,  6.38769146e+05,
 6.64168391e+05,  4.25617316e+05,  4.96551610e+05,  1.43638309e+05,
-1.69339674e+05,  1.48175453e+06, -1.50495831e+04,  5.45586353e+05,
 6.66394879e+05,  6.45290196e+05,  8.13062174e+05,  5.76763898e+05,
 6.72503711e+05,  3.91121489e+05,  3.72394974e+05,  4.25617316e+05,
 5.33192446e+03,  2.30942938e+05,  5.05451130e+05,  7.37503647e+05,
 4.32897080e+05,  6.45290196e+05,  7.85677664e+05,  2.37334578e+05,
 3.31012999e+05,  8.13268844e+05,  2.56701524e+05,  2.29712367e+05,
 5.14437736e+05,  6.50455134e+05,  6.75381785e+05,  1.33600632e+06,
 1.23645368e+05,  3.94556386e+05,  8.17805614e+04,  4.80494952e+05,
 5.63707284e+05,  2.17002934e+05,  5.30829357e+05,  8.35765767e+05,
-5.38206972e+04,  7.24241754e+05,  1.50343926e+05,  7.42742207e+04,
 5.09222935e+05,  3.97430242e+05,  6.73601881e+05,  5.24352118e+05,
 5.68762327e+05,  6.54492316e+05,  2.74587649e+05,  5.48659952e+05,
 6.01004597e+05,  2.59254293e+04,  3.20447792e+05,  7.19704608e+05,
-2.83988621e+04,  1.61967589e+06,  7.64202204e+05,  3.18998159e+05,
 2.30264826e+05,  6.59060268e+05,  4.64156241e+05,  4.82874333e+03,
 6.48683227e+05,  5.66078434e+05,  2.56875997e+05,  7.20681947e+05,
-4.44178389e+04, -5.57753746e+04,  5.05451130e+05,  4.61128307e+05,
 1.70247065e+06,  1.23732755e+05,  1.21976479e+06,  4.11839020e+03,
 3.72307888e+05,  2.74500262e+05,  3.69270872e+05,  1.70247065e+06,
 3.81207407e+05,  4.68868809e+05,  3.81885061e+05,  1.36408758e+04,
 5.16334253e+05,  1.45558833e+06,  6.52235038e+05,  5.70737468e+04,
 4.70049086e+05,  1.22630806e+05,  4.96638997e+05,  1.31215223e+06,
 4.08920527e+05, -5.98355020e+02,  2.83574555e+05,  1.41444707e+05,
-3.30283424e+05, -9.62175350e+04,  4.52228788e+05, -1.06872103e+04,
 1.61356746e+06,  2.66877766e+05,  4.41017964e+05,  3.31369799e+05,
 5.70184642e+05,  8.84281590e+05,  4.38966595e+05,  4.70027526e+05,
 5.99580674e+05,  5.89435222e+05,  7.04230257e+04,  4.34516836e+05,
-1.31815612e+05,  1.67041015e+05, -2.17024034e+05,  4.18498159e+05,
 1.47823726e+06,  5.10313289e+05, -4.08581891e+04,  4.31846980e+05,
 7.64202204e+05,  3.54595936e+05,  4.10178559e+05,  1.27080565e+05,
 7.40024466e+05,  2.22883508e+05,  8.72633181e+05,  7.64202204e+05,
 5.44018447e+05,  3.50146176e+05,  6.15235064e+04,  4.35531698e+05,
 3.19085245e+05,  4.71129775e+05,  7.40024466e+05,  2.55774048e+05,
 5.13460697e+05,  8.02382751e+05,  4.56678247e+05,  4.43416355e+05,
 7.66098722e+05,  3.99006145e+05,  1.31946323e+05,  5.05538517e+05,
-1.40350207e+04, -1.25335003e+04,  6.95967293e+05,  3.63408068e+05,
 7.37503647e+05, -6.27202165e+04,  5.27113739e+04,  4.28732148e+05,
 2.31192302e+05,  7.55506686e+05, -8.17085910e+04,  2.88958567e+05,
 7.94221203e+05,  4.46855331e+05,  4.97740946e+05,  3.00184257e+05,
 3.94556386e+05,  4.97653560e+05,  9.58448609e+04,  2.79239800e+05,
 2.48234801e+04,  7.80376221e+05,  3.32347137e+05,  1.60187685e+06,
 4.77859103e+05,  7.94221203e+05,  4.52228487e+05, -2.11649125e+05,
 3.81207107e+05,  4.37746673e+05,  2.77257504e+05,  7.31477984e+05,
 1.23732755e+05,  3.90019539e+05,  1.28554106e+06,  6.57495360e+05,
 4.97740946e+05, -9.75527943e+04,  2.99797690e+05,  4.31161028e+05,
Loading [MathJax]/extensions/Safe.js 59e+05,  6.75381785e+05,  6.04056071e+05, -1.45517570e+04,
 6.26143788e+05,  1.17002178e+06,  1.59243445e+05,  6.04215573e+04,
 7.84710287e+05,  1.01397626e+04,  8.06693473e+05,  4.47895341e+05,
 5.01125981e+05,  6.01004597e+05,  5.18887495e+05,  2.78022846e+05,
 1.00916542e+06,  5.16094248e+04,  6.75294398e+05,  4.52236698e+05,
 5.60640626e+05,  4.07818578e+05,  3.59045695e+05,  8.92820099e+04,
 3.10010952e+05,  2.33824634e+05,  4.41840451e+05,  3.18998159e+05,
 7.08222620e+05,  1.03419914e+06,  2.63995612e+05,  7.15342235e+05,
 4.72171738e+05,  3.10389726e+05,  3.03066110e+05,  3.89844766e+05,
 1.49970715e+06,  4.35406787e+05,  7.64202204e+05, -5.98355020e+02,
 5.59950195e+05,  5.13334195e+05,  1.65073682e+06,  7.37503647e+05,
 5.80922270e+05,  2.77257504e+05,  5.50982851e+05,  7.55302685e+05,
 2.78862935e+05,  5.15539985e+05,  6.93093437e+05,  4.70027526e+05,
 3.27897378e+05, -9.63049218e+04,  2.21190833e+05,  5.95460885e+05,
 3.54508849e+05,  1.49241977e+05,  6.43591295e+05,  8.03272703e+05,
 6.64168391e+05,  5.66982423e+05,  3.96336289e+05,  3.05561193e+05,
 6.08810164e+05,  5.59862808e+05,  4.78927045e+05,  4.88579267e+05,
 8.08030649e+04, -1.94993428e+04,  4.61128307e+05,  1.23820141e+05,
 5.06867416e+05,  4.69659287e+05,  7.97575342e+01,  1.60987344e+04,
 4.97653560e+05,  3.26795428e+05,  2.39164345e+05,  4.43416355e+05,
 1.48175453e+06,  6.48683227e+05,  6.50455134e+05,  1.63007253e+05,
 4.71042389e+05,  1.60987344e+04,  8.05995267e+05,  4.25530230e+05,
 1.94754135e+05,  4.79941908e+05,  3.01198820e+05,  1.75853148e+05,
 3.14635485e+05,  1.25892959e+06,  6.07832825e+05,  3.97342855e+05,
 5.16217639e+05,  3.50138179e+05,  6.19161225e+05,  4.58662151e+05,
 2.48234801e+04, -3.98806925e+04,  5.33251637e+05,  4.28491173e+05,
 5.95460885e+05,  8.70327284e+04, -1.25335003e+04,  4.26632179e+05,
 2.37887622e+05,  4.61128007e+05,  3.00184257e+05,  5.06553079e+05,
 7.43937311e+05,  1.93652186e+05,  8.46996961e+05,  2.30177439e+05,
 4.67383680e+05,  2.43302424e+05,  1.36958234e+06,  5.41049207e+05,
 4.18498159e+05,  2.49494679e+05,  4.52228788e+05,  7.93578658e+05,
 7.50610342e+05,  2.86602789e+05, -7.99285293e+04,  1.70247065e+06,
 7.86358478e+05,  4.25530230e+05, -1.17219281e+05,  1.94666749e+05,
 4.62230256e+05,  4.96638997e+05,  5.06553079e+05, -2.83988621e+04,
 6.39783708e+05,  5.59950195e+05,  6.28418333e+05,  5.18887495e+05,
 6.36427901e+05,  8.71201151e+04,  4.07818278e+05,  1.25884250e+06,
 1.36337674e+05,  2.21452693e+05,  8.81346775e+04,  6.49587037e+04,
 1.34930705e+06,  5.14437736e+05,  3.29502966e+05,  4.78482069e+05,
 7.39831337e+04,  1.26851400e+06, -2.59174656e+05,  6.13085150e+05,
 2.23684757e+04,  5.40687788e+05,  3.89337909e+05,  8.44123104e+05,
-4.43304521e+04,  3.30559237e+05, -1.79341143e+05,  1.77633209e+05,
 4.44605991e+05,  1.06021103e+05,  1.20850902e+05,  1.19663326e+06,
 4.31979784e+05,  8.86614374e+05,  4.96638697e+05,  6.34494915e+04,
 3.19085245e+05,  6.48683227e+05,  7.06066468e+05,  1.94841522e+05,
-1.68574175e+05,  3.90019539e+05,  7.88894435e+05,  9.59322476e+04,
 3.75568573e+05,  7.28604127e+05,  1.79909220e+05, -1.33238258e+05,
 6.39696321e+05,  7.49750976e+05,  3.33025092e+05,  6.08766705e+05,
 6.82798846e+05, -4.61105139e+04,  4.04133860e+05,  3.59045695e+05,
 8.72633181e+05,  1.46309451e+06,  5.00305299e+05,  2.99710303e+05,
 4.77949707e+05,  4.91515662e+04,  1.69179122e+06,  7.14747602e+04,
 4.70027526e+05,  5.05538517e+05,  5.32149688e+05,  4.34517136e+05,
 3.98918758e+05,  3.94556386e+05,  3.50146176e+05,  4.51657110e+05,
 8.41929574e+05,  1.14679595e+06,  6.16985803e+04,  2.55009164e+05,
 4.38966595e+05,  5.47812743e+05,  7.85677664e+05, -6.21294907e+04,
 5.76763598e+05,  6.24945912e+05,  7.93225450e+04,  1.60987344e+04,
 5.09987976e+05, -1.22558993e+05,  1.40678305e+06,  4.21371557e+05,
 4.78927045e+05, -1.06872103e+04,  4.65577766e+05,  1.59243445e+05,
 1.70247065e+06,  4.34517136e+05,  5.95373498e+05,  5.67660378e+05,
-1.50862523e+05,  4.95952746e+05,  6.48561624e+05,  1.48664926e+06,
-9.55610511e+04,  4.03455905e+05,  4.21080470e+05, -3.61964322e+04,
 5.50694695e+05,  1.31660169e+06,  6.04273018e+05,  3.01198820e+05,
 2.94079662e+05,  6.59391877e+05,  5.10966923e+05,  5.01088457e+05,
 4.93291187e+05,  6.39871095e+05,  1.16759220e+05,  5.33339024e+05,
 3.38103862e+04,  5.15452598e+05,  3.94556386e+05,  6.22016190e+04,
 1.64006650e+06,  4.88841427e+05,  3.54421462e+05,  3.40975063e+05,
 6.66482266e+05,  7.64202204e+05,  3.24213117e+05, -7.27216849e+04,
 7.11782427e+05,  1.36398365e+06,  6.62119893e+05,  4.30330703e+05,
 1.36190215e+06, -2.83988621e+04,  5.07894579e+05,  5.83205559e+05,
 5.05451130e+05,  3.17758327e+05,  6.79058207e+05, -1.78151807e+05,
 6.02467837e+04,  7.37793082e+05,  2.65775516e+05,  5.68674940e+05,
         40e+06,  1.36378074e+06, -2.83114753e+04,  3.49123354e+04,
```

Loading [MathJax]/extensions/Safe.js

```
        4.91449906e+05,  3.23535005e+05,  4.38423926e+05,  7.90813376e+05,
        6.24945912e+05,  6.01004597e+05,  1.76955097e+05,  5.37701396e+05,
        4.84508135e+05,  4.70027526e+05,  2.74587649e+05, -1.78151807e+05,
        3.01286207e+05,  4.25350331e+05,  5.06553079e+05,  6.57582747e+05,
        1.68142964e+05,  7.39487551e+05,  2.92250008e+05,  3.01286207e+05,
       -8.17085910e+04,  5.05894197e+05,  3.69257072e+04,  1.24939574e+06,
        3.81207107e+05,  3.55904479e+04,  2.21172170e+05,  4.70231827e+05,
        7.10892475e+05,  1.76955097e+05,  1.40678305e+06,  5.16217639e+05,
        4.89519540e+05,  1.57081592e+06,  1.68733690e+05,  3.04634017e+05,
        7.90813376e+05,  2.60128161e+04,  1.25454038e+06,  7.01992956e+05,
        3.10185726e+05,  6.92336898e+04,  2.66665468e+05,  5.33164250e+05,
       -9.63049218e+04,  8.08612414e+05,  4.85273322e+05, -3.61964322e+04,
        7.81038345e+05,  5.89821789e+05,  6.56480797e+05,  2.47976478e+05,
        1.13643899e+05,  7.06530103e+05,  5.86473979e+05,  3.27897378e+05,
        8.36824810e+05,  6.13172537e+05,  6.92336898e+04,  6.11392633e+05,
        5.68674940e+05,  3.69633073e+05,  2.09584092e+05,  8.37207490e+05,
        1.04831767e+05, -5.26310610e+04,  4.52228788e+05,  1.05933716e+05,
       -4.52707250e+04,  5.50231060e+05,  1.06524442e+05,  4.30550521e+05,
        6.39871095e+05,  6.29869626e+05,  4.38966595e+05,  4.27135518e+05,
        1.31927155e+06,  2.92250008e+05,  2.71597783e+05,  4.25530230e+05,
        1.49124196e+06,  3.87349841e+05,  5.21020911e+05,  4.79941908e+05,
        7.48387070e+05,  8.41306608e+05,  7.39934581e+05,  7.60759010e+05,
        7.11011383e+04,  6.79831545e+05,  4.61128007e+05,  8.72633181e+05,
        3.54508849e+05,  7.58551010e+05,  5.97788087e+05,  1.17707092e+05,
        6.30094113e+05,  2.47976478e+05,  1.60276680e+06,  6.49587037e+04,
        4.65577766e+05,  7.00815135e+05,  1.65963634e+06,  7.06066468e+05,
        5.33251637e+05,  2.56875997e+05,  4.62142869e+05,  4.96638997e+05,
        1.01721882e+05,  1.23732755e+05,  7.16709231e+05,  1.10383476e+05,
        7.00964373e+05,  5.06640466e+05,  4.36685648e+05, -3.41032824e+04,
        7.32039325e+05,  1.14745849e+05,  1.40255071e+05,  1.50245640e+06,
        4.24195791e+05,  1.76955397e+05,  1.23732755e+05,  5.85372030e+05,
        4.24292178e+05,  4.84383370e+05,  5.18887495e+05,  1.76955097e+05,
        1.79362122e+06,  5.24177344e+05,  8.16738595e+05,  2.33816637e+05,
        4.54695636e+05,  7.89357742e+04,  8.04162654e+05,  4.96551610e+05,
        7.85677664e+05,  3.90019239e+05,  1.27833376e+06,  3.93134860e+05,
        1.70247065e+06,  2.61413143e+05,  4.35619085e+05,  5.18887495e+05,
        2.29712367e+05,  1.84665280e+05,  2.52513624e+05,  4.06654195e+05,
        2.29075490e+05,  6.86739163e+05,  1.66953628e+05,  4.28821443e+05,
        4.27099055e+04,  4.82486853e+05,  1.06247515e+06,  1.46092804e+06,
        2.56788610e+05,  1.72018230e+06,  1.48175453e+06,  1.57421713e+05,
        5.41791061e+04,  1.84752667e+05,  4.96842697e+05,  1.83010445e+05,
        5.20044433e+05,  8.98271946e+04,  5.14196760e+05,  1.59921558e+05,
        2.21452693e+05,  4.43241882e+05, -1.94993428e+04, -4.43304521e+04,
        1.28109100e+06, -9.55610511e+04,  3.18997858e+05,  2.74762723e+05,
        5.14437736e+05,  1.45728085e+06,  4.27310291e+05,  7.10892475e+05,
        7.19791995e+05,  8.57987867e+05,  4.77862620e+05,  1.87526010e+04,
        6.35413338e+05,  3.09499774e+05,  4.27397678e+05,  2.39164345e+05,
        1.57081592e+06,  6.45290196e+05,  1.36398365e+06,  1.13643899e+05,
        3.31897503e+05,  4.04084155e+05,  4.97740946e+05,  4.17820046e+05,
        1.10845883e+06,  4.86545034e+05,  6.08882969e+05, -2.64871468e+05,
        4.25617316e+05,  6.49864566e+05,  5.68587554e+05,  3.90706655e+05,
        2.92678075e+05,  2.96836447e+05,  4.84383370e+05,  1.40342457e+05,
        4.76257189e+05,  7.68651964e+05,  1.76955397e+05,  3.80606218e+05,
        3.66523689e+05,  4.53330737e+05,  1.10075381e+06,  3.09083777e+05,
       -1.69339674e+05, -2.29003677e+05,  3.33274613e+05,  1.66408610e+06,
        4.61128307e+05,  1.27080565e+05,  4.80706949e+05,  5.69477506e+05,
        3.90019539e+05,  3.17895909e+05,  4.17907433e+05,  6.46903323e+05,
        2.38815098e+05,  4.18374857e+05,  4.53410127e+05, -2.83114753e+04,
        3.10185726e+05,  2.02464319e+05,  4.16718097e+05,  1.94666749e+05,
        8.41306608e+05,  2.84239870e+05,  3.08766454e+05,  4.31759593e+05,
        3.27139610e+05,  6.57582747e+05,  7.68651964e+05,  5.14438036e+05,
       -1.42641117e+05,  2.17002934e+05,  4.77949707e+05,  1.53175808e+06,
        3.85656866e+05,  2.03566268e+05,  6.54833886e+05,  5.95460885e+05,
        1.41444406e+05,  6.01004597e+05,  1.41531793e+05, -5.10968243e+02,
        4.11378085e+05,  3.94556386e+05,  5.86473979e+05,  1.40255071e+05,
        7.89930745e+05,  2.04244381e+05,  4.26632179e+05,  5.64697400e+05,
        4.09686026e+05, -1.29442088e+04,  6.39783708e+05,  2.86922365e+05])
```

In [160…

```
mod.score(xtest,ytest)
```

Loading [MathJax]/extensions/Safe.js

```
Out[160… 0.5446347861009214

In [161… sc1=StandardScaler()

In [163… train_sc1=sc1.fit_transform(xtrain)

In [164… mod1.fit(train_sc1,ytrain)

Out[164… RandomForestRegressor()

In [168… pred1=mod1.predict(xtest)

In [166… test_sc1=sc1.transform(xtest)

In [167… mod1.score(test_sc1,ytest)

Out[167… 0.7300472420772091

In [169…

         ---------------------------------------------------------------------------
         AttributeError                            Traceback (most recent call last)
         <ipython-input-169-2b555aa123b4> in <module>
         ----> 1 mod1.accuracy_score(ytest,pred1)

         AttributeError: 'RandomForestRegressor' object has no attribute 'accuracy_score'

 In [ ]:
```

Loading [MathJax]/extensions/Safe.js