In [54]:
```
import pandas as pd
import numpy as np
import datetime as dt
```

In [55]:
```
df = pd.read_csv("CAR DETAILS FROM CAR DEKHO.csv")
```

In [9]:
```
df.describe()
```

Out[9]:

	year	selling_price	km_driven
count	4340.000000	4.340000e+03	4340.000000
mean	2013.090783	5.041273e+05	66215.777419
std	4.215344	5.785487e+05	46644.102194
min	1992.000000	2.000000e+04	1.000000
25%	2011.000000	2.087498e+05	35000.000000
50%	2014.000000	3.500000e+05	60000.000000
75%	2016.000000	6.000000e+05	90000.000000
max	2020.000000	8.900000e+06	806599.000000

In [10]:
```
df.shape
```

Out[10]:
```
(4340, 8)
```

In [11]:
```
df.head()
```

Out[11]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner
0	Maruti 800 AC	2007	60000	70000	Petrol	Individual	Manual	First Owner
1	Maruti Wagon R LXI Minor	2007	135000	50000	Petrol	Individual	Manual	First Owner
2	Hyundai Verna 1.6 SX	2012	600000	100000	Diesel	Individual	Manual	First Owner
3	Datsun RediGO T Option	2017	250000	46000	Petrol	Individual	Manual	First Owner
4	Honda Amaze VX i-DTEC	2014	450000	141000	Diesel	Individual	Manual	Second Owner

In [56]:
```
df_new = df['current_year'] = '2021'
```

Out[56]:
```
'2021'
```

In [23]:
```

```

In [61]:
```
df
```

Out[61]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	current_year
0	Maruti 800 AC	2007	60000	70000	Petrol	Individual	Manual	First Owner	2021
1	Maruti Wagon R LXI Minor	2007	135000	50000	Petrol	Individual	Manual	First Owner	2021
2	Hyundai Verna 1.6 SX	2012	600000	100000	Diesel	Individual	Manual	First Owner	2021
3	Datsun RediGO T Option	2017	250000	46000	Petrol	Individual	Manual	First Owner	2021
4	Honda Amaze VX i-DTEC	2014	450000	141000	Diesel	Individual	Manual	Second Owner	2021
...	...	...	...	...	...	...	...	...	...
4335	Hyundai i20 Magna 1.4 CRDi (Diesel)	2014	409999	80000	Diesel	Individual	Manual	Second Owner	2021
4336	Hyundai i20 Magna 1.4 CRDi	2014	409999	80000	Diesel	Individual	Manual	Second Owner	2021
4337	Maruti 800 AC BSIII	2009	110000	83000	Petrol	Individual	Manual	Second Owner	2021
4338	Hyundai Creta 1.6 CRDi SX Option	2016	865000	90000	Diesel	Individual	Manual	First Owner	2021
4339	Renault KWID RXT	2016	225000	40000	Petrol	Individual	Manual	First Owner	2021

4340 rows × 9 columns

In [64]:
```
df['duration'] = 2021- df['year']
```

In [62]:
```
type(df)
```

Out[62]:
```
pandas.core.frame.DataFrame
```

In [29]:
```
type(df['year'])
```

Out[29]: pandas.core.series.Series

In [30]:
```
type(df['current_year'])
```

Out[30]: pandas.core.series.Series

In [51]:

In [65]:
```
df
```

Out[65]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	current_year	duration
0	Maruti 800 AC	2007	60000	70000	Petrol	Individual	Manual	First Owner	2021	14
1	Maruti Wagon R LXI Minor	2007	135000	50000	Petrol	Individual	Manual	First Owner	2021	14
2	Hyundai Verna 1.6 SX	2012	600000	100000	Diesel	Individual	Manual	First Owner	2021	9
3	Datsun RediGO T Option	2017	250000	46000	Petrol	Individual	Manual	First Owner	2021	4
4	Honda Amaze VX i-DTEC	2014	450000	141000	Diesel	Individual	Manual	Second Owner	2021	7
...	...	...	...	...	...	...	...	...	...	...
4335	Hyundai i20 Magna 1.4 CRDi (Diesel)	2014	409999	80000	Diesel	Individual	Manual	Second Owner	2021	7
4336	Hyundai i20 Magna 1.4 CRDi	2014	409999	80000	Diesel	Individual	Manual	Second Owner	2021	7
4337	Maruti 800 AC BSIII	2009	110000	83000	Petrol	Individual	Manual	Second Owner	2021	12
4338	Hyundai Creta 1.6 CRDi SX Option	2016	865000	90000	Diesel	Individual	Manual	First Owner	2021	5
4339	Renault KWID RXT	2016	225000	40000	Petrol	Individual	Manual	First Owner	2021	5

4340 rows × 10 columns

In [67]:
```
print(df['seller_type'].unique())
```

```
['Individual' 'Dealer' 'Trustmark Dealer']
```

In [68]:
```
print(df['fuel'].unique())
```

```
['Petrol' 'Diesel' 'CNG' 'LPG' 'Electric']
```

In [69]:
```
print(df['owner'].unique())
```

```
['First Owner' 'Second Owner' 'Fourth & Above Owner' 'Third Owner'
 'Test Drive Car']
```

In [70]:
```
print(df['transmission'].unique())
```

```
['Manual' 'Automatic']
```

In [71]:
```
#checking missing values
df.isnull().sum()
```

Out[71]:
```
name             0
year             0
selling_price    0
km_driven        0
fuel             0
seller_type      0
transmission     0
owner            0
current_year     0
duration         0
dtype: int64
```

In [75]:
```
df.drop(['year'], axis=1, inplace=True)
```

In [76]:
```
df.drop(['current_year'], axis=1, inplace=True)
```

In [77]:
```
df
```

Out[77]:

	name	selling_price	km_driven	fuel	seller_type	transmission	owner	duration
0	Maruti 800 AC	60000	70000	Petrol	Individual	Manual	First Owner	14
1	Maruti Wagon R LXI Minor	135000	50000	Petrol	Individual	Manual	First Owner	14
2	Hyundai Verna 1.6 SX	600000	100000	Diesel	Individual	Manual	First Owner	9
3	Datsun RediGO T Option	250000	46000	Petrol	Individual	Manual	First Owner	4
4	Honda Amaze VX i-DTEC	450000	141000	Diesel	Individual	Manual	Second Owner	7
...	...	...	...	...	...	...	...	...
4335	Hyundai i20 Magna 1.4 CRDi (Diesel)	409999	80000	Diesel	Individual	Manual	Second Owner	7
4336	Hyundai i20 Magna 1.4 CRDi	409999	80000	Diesel	Individual	Manual	Second Owner	7

	name	selling_price	km_driven	fuel	seller_type	transmission	owner	duration
4337	Maruti 800 AC BSIII	110000	83000	Petrol	Individual	Manual	Second Owner	12
4338	Hyundai Creta 1.6 CRDi SX Option	865000	90000	Diesel	Individual	Manual	First Owner	5
4339	Renault KWID RXT	225000	40000	Petrol	Individual	Manual	First Owner	5

4340 rows × 8 columns

In [79]:
```
df.drop(['name'], axis=1, inplace=True)
```

In [80]:
```
df
```

Out[80]:

	selling_price	km_driven	fuel	seller_type	transmission	owner	duration
0	60000	70000	Petrol	Individual	Manual	First Owner	14
1	135000	50000	Petrol	Individual	Manual	First Owner	14
2	600000	100000	Diesel	Individual	Manual	First Owner	9
3	250000	46000	Petrol	Individual	Manual	First Owner	4
4	450000	141000	Diesel	Individual	Manual	Second Owner	7
...	...	...	...	...	...	...	...
4335	409999	80000	Diesel	Individual	Manual	Second Owner	7
4336	409999	80000	Diesel	Individual	Manual	Second Owner	7
4337	110000	83000	Petrol	Individual	Manual	Second Owner	12
4338	865000	90000	Diesel	Individual	Manual	First Owner	5
4339	225000	40000	Petrol	Individual	Manual	First Owner	5

4340 rows × 7 columns

In [85]:
```
df = pd.get_dummies(df, drop_first=True)
```

In [83]:
```
df
```

Out[83]:

	selling_price	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	transmission_Manual
0	60000	70000	14	0	0	0	1	1	0	1
1	135000	50000	14	0	0	0	1	1	0	1
2	600000	100000	9	1	0	0	0	1	0	1
3	250000	46000	4	0	0	0	1	1	0	1
4	450000	141000	7	1	0	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...
4335	409999	80000	7	1	0	0	0	1	0	1
4336	409999	80000	7	1	0	0	0	1	0	1
4337	110000	83000	12	0	0	0	1	1	0	1
4338	865000	90000	5	1	0	0	0	1	0	1
4339	225000	40000	5	0	0	0	1	1	0	1

4340 rows × 14 columns

In [86]:
```
df.head()
```

Out[86]:

	selling_price	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	transmission_Manual	ow
0	60000	70000	14	0	0	0	1	1	0	1	
1	135000	50000	14	0	0	0	1	1	0	1	
2	600000	100000	9	1	0	0	0	1	0	1	
3	250000	46000	4	0	0	0	1	1	0	1	
4	450000	141000	7	1	0	0	0	1	0	1	

In [87]:
```
df.corr()
```

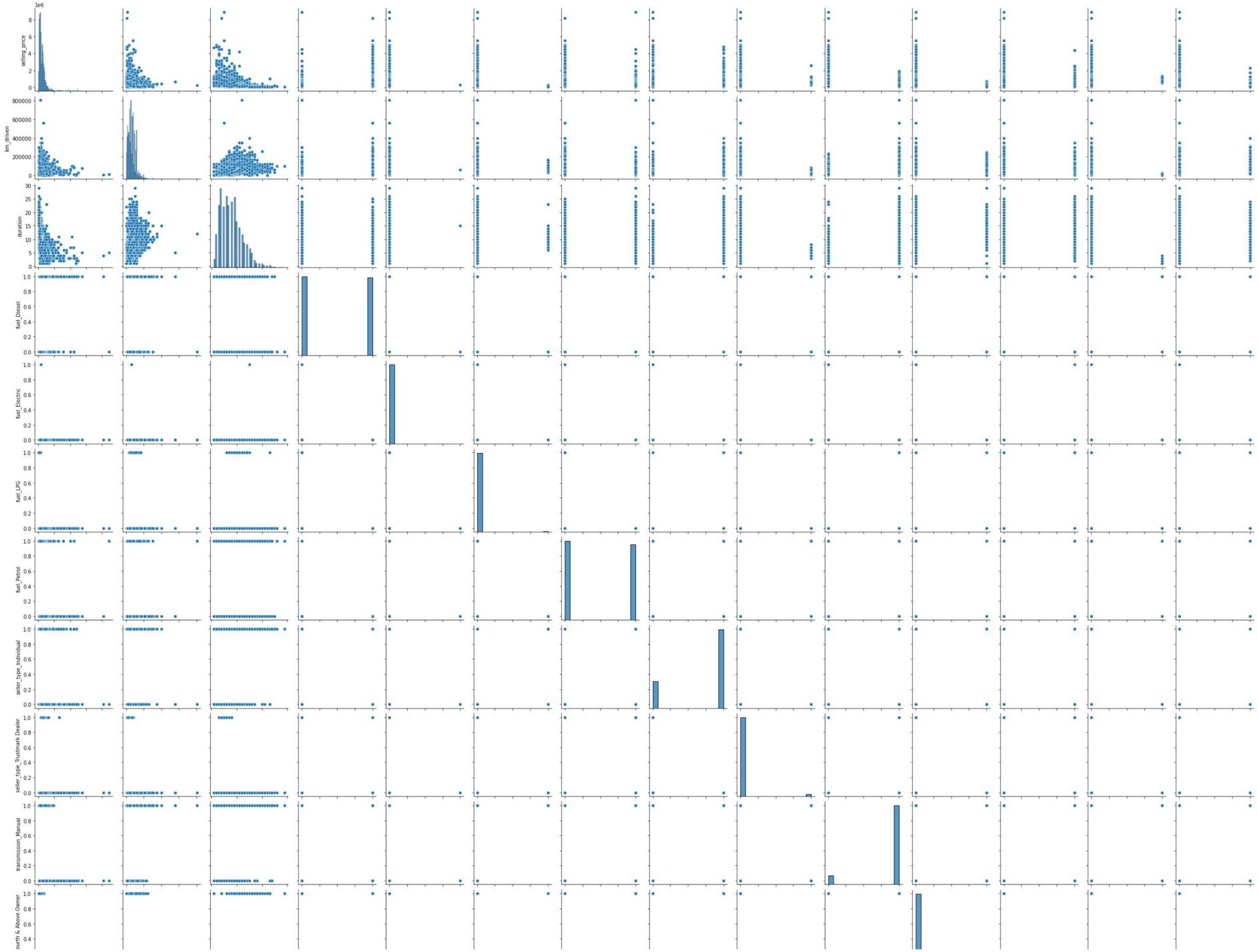
Out[87]:

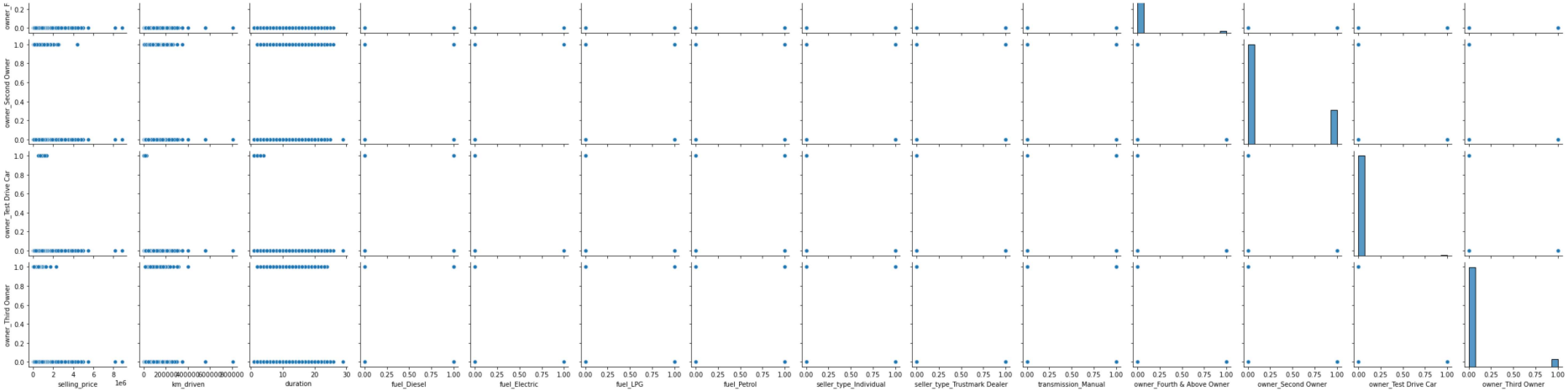
	selling_price	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	tran
selling_price	1.000000	-0.192289	-0.413922	0.282947	-0.005095	-0.042434	-0.269453	-0.236798	0.110176	
km_driven	-0.192289	1.000000	0.419688	0.285396	-0.001372	0.036652	-0.291139	0.182663	-0.089858	
duration	-0.413922	0.419688	1.000000	-0.121424	0.025540	0.051266	0.114908	0.173703	-0.100225	
fuel_Diesel	0.282947	0.285396	-0.121424	1.000000	-0.015063	-0.072422	-0.970934	-0.035314	-0.007911	
fuel_Electric	-0.005095	-0.001372	0.025540	-0.015063	1.000000	-0.001108	-0.014856	-0.026118	-0.002355	
fuel_LPG	-0.042434	0.036652	0.051266	-0.072422	-0.001108	1.000000	-0.071427	0.035122	-0.011324	
fuel_Petrol	-0.269453	-0.291139	0.114908	-0.970934	-0.014856	-0.071427	1.000000	0.029845	0.012489	
seller_type_Individual	-0.236798	0.182663	0.173703	-0.035314	-0.026118	0.035122	0.029845	1.000000	-0.266904	
seller_type_Trustmark Dealer	0.110176	-0.089858	-0.100225	-0.007911	-0.002355	-0.011324	0.012489	-0.266904	1.000000	
transmission_Manual	-0.530205	0.120226	0.143800	-0.048098	-0.044746	0.024764	0.039614	0.214163	-0.042346	
owner_Fourth & Above Owner	-0.078725	0.097349	0.186361	-0.021054	-0.002094	0.013384	0.011502	0.072322	-0.021395	
owner_Second Owner	-0.161986	0.195203	0.292349	0.003524	0.025960	0.030141	-0.012717	0.186573	-0.076764	
owner_Test Drive Car	0.048799	-0.083445	-0.095795	-0.010575	-0.000952	-0.004577	0.012427	-0.107886	-0.009729	
owner_Third Owner	-0.111326	0.194713	0.244308	0.009373	-0.004166	0.004836	-0.008502	0.140823	-0.042578	

In [95]:
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
plt.show()
```

In [96]:
```
sns.pairplot(df)
```

Out[96]:
```
<seaborn.axisgrid.PairGrid at 0x27aebdd99d0>
```


selling_price
km_driven
duration
fuel_Diesel
fuel_Electric
fuel_LPG
fuel_Petrol
seller_type_Individual
seller_type_Trustmark Dealer
transmission_Manual
fourth & Above Owner
1e6



In [117…

```
corrmat = df.corr()
corrmat
```

Out[117…

	selling_price	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	tran
selling_price	1.000000	-0.192289	-0.413922	0.282947	-0.005095	-0.042434	-0.269453	-0.236798	0.110176	
km_driven	-0.192289	1.000000	0.419688	0.285396	-0.001372	0.036652	-0.291139	0.182663	-0.089858	
duration	-0.413922	0.419688	1.000000	-0.121424	0.025540	0.051266	0.114908	0.173703	-0.100225	
fuel_Diesel	0.282947	0.285396	-0.121424	1.000000	-0.015063	-0.072422	-0.970934	-0.035314	-0.007911	
fuel_Electric	-0.005095	-0.001372	0.025540	-0.015063	1.000000	-0.001108	-0.014856	-0.026118	-0.002355	
fuel_LPG	-0.042434	0.036652	0.051266	-0.072422	-0.001108	1.000000	-0.071427	0.035122	-0.011324	
fuel_Petrol	-0.269453	-0.291139	0.114908	-0.970934	-0.014856	-0.071427	1.000000	0.029845	0.012489	
seller_type_Individual	-0.236798	0.182663	0.173703	-0.035314	-0.026118	0.035122	0.029845	1.000000	-0.266904	
seller_type_Trustmark Dealer	0.110176	-0.089858	-0.100225	-0.007911	-0.002355	-0.011324	0.012489	-0.266904	1.000000	
transmission_Manual	-0.530205	0.120226	0.143800	-0.048098	-0.044746	0.024764	0.039614	0.214163	-0.042346	
owner_Fourth & Above Owner	-0.078725	0.097349	0.186361	-0.021054	-0.002094	0.013384	0.011502	0.072322	-0.021395	

	selling_price	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	tran
owner_Second Owner	-0.161986	0.195203	0.292349	0.003524	0.025960	0.030141	-0.012717	0.186573	-0.076764	
owner_Test Drive Car	0.048799	-0.083445	-0.095795	-0.010575	-0.000952	-0.004577	0.012427	-0.107886	-0.009729	
owner_Third Owner	-0.111326	0.194713	0.244308	0.009373	-0.004166	0.004836	-0.008502	0.140823	-0.042578	

In [118…
```
top_corr_features=corrmat.index
top_corr_features
```

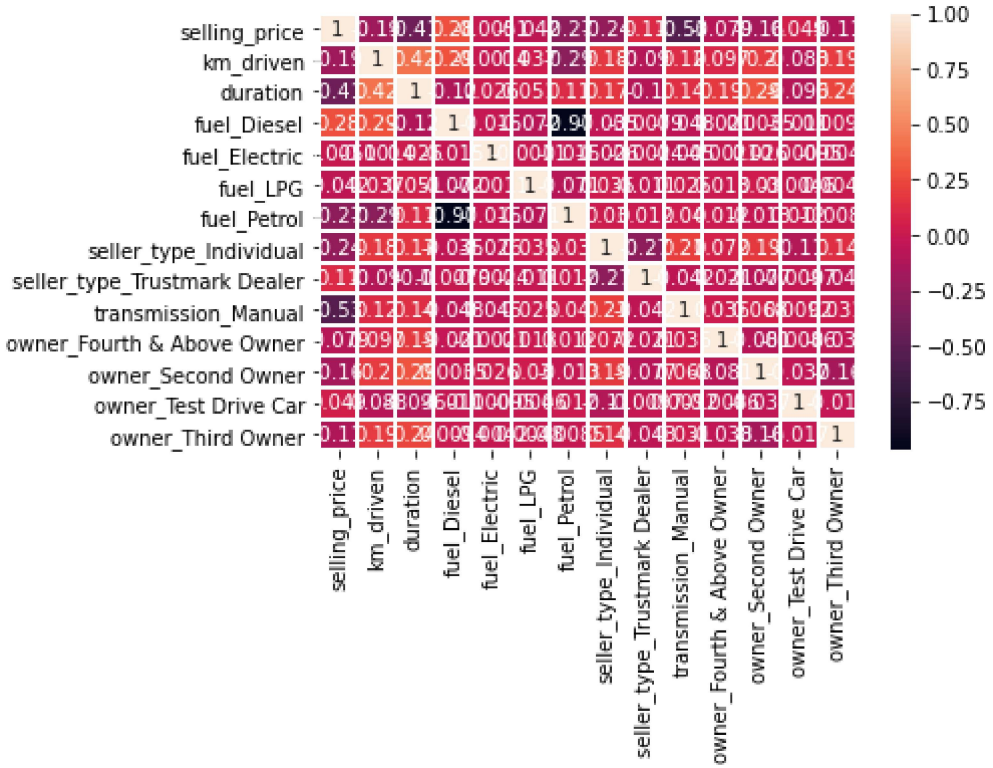
Out[118…
```
Index(['selling_price', 'km_driven', 'duration', 'fuel_Diesel',
       'fuel_Electric', 'fuel_LPG', 'fuel_Petrol', 'seller_type_Individual',
       'seller_type_Trustmark Dealer', 'transmission_Manual',
       'owner_Fourth & Above Owner', 'owner_Second Owner',
       'owner_Test Drive Car', 'owner_Third Owner'],
      dtype='object')
```

In [119…
```
plt.figure(figsize=(20,15))
```

Out[119…
```
<Figure size 1440x1080 with 0 Axes>
```

```
<Figure size 1440x1080 with 0 Axes>
```

In [120…
```
#HEATMAP
g=sns.heatmap(df[top_corr_features].corr(), annot=True, linewidths=1)
```



In [125…
```
df.head()
```

Out[125…

	selling_price	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	transmission_Manual	ow
0	60000	70000	14	0	0	0	1	1	0	1	
1	135000	50000	14	0	0	0	1	1	0	1	
2	600000	100000	9	1	0	0	0	1	0	1	
3	250000	46000	4	0	0	0	1	1	0	1	
4	450000	141000	7	1	0	0	0	1	0	1	

In [126…
```
#Independent and dependent features
```

```
X = df.iloc[:,1:]
Y =df.iloc[:,0]
```

In [127…
```
X.head()
```

Out[127…

	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	transmission_Manual	owner_Fourth & Above Owner	o
0	70000	14	0	0	0	1	1	0	1	0	
1	50000	14	0	0	0	1	1	0	1	0	
2	100000	9	1	0	0	0	1	0	1	0	
3	46000	4	0	0	0	1	1	0	1	0	
4	141000	7	1	0	0	0	1	0	1	0	

In [128…
```
Y.head()
```

Out[128…
```
0     60000
1    135000
2    600000
3    250000
4    450000
Name: selling_price, dtype: int64
```

In [130…
```
#Finding the order of important feature
from sklearn.ensemble import ExtraTreesRegressor
model=ExtraTreesRegressor()
model.fit(X,Y)
```

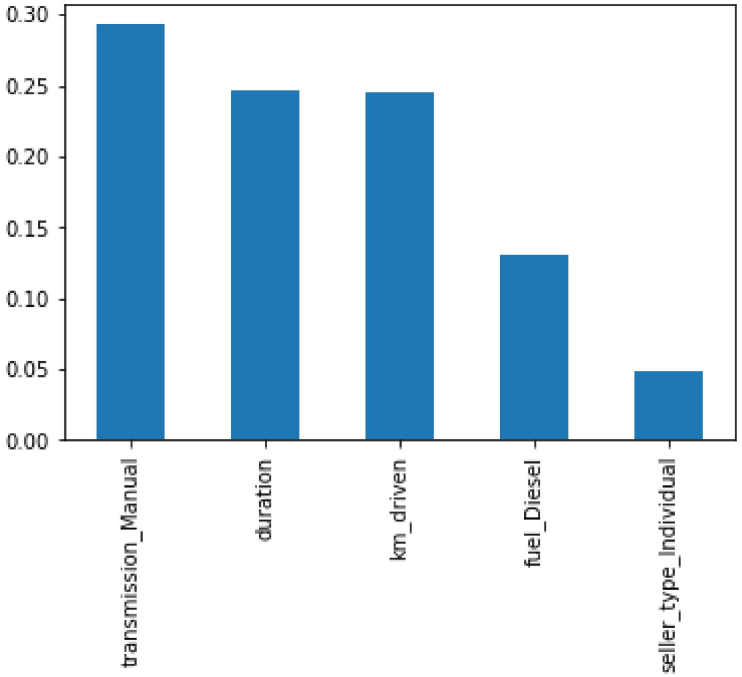
Out[130…
```
ExtraTreesRegressor()
```

In [131…
```
print(model.feature_importances_)
```

```
[2.44744018e-01 2.46782514e-01 1.30387786e-01 2.45924432e-05
 2.60367925e-05 2.13027883e-04 4.87826386e-02 8.88211520e-03
 2.92828169e-01 8.12208410e-04 2.12718010e-02 5.99180906e-04
 4.64591109e-03]
```

In [134…
```
#Plot showing the feature importance
```

```
feature_importances = pd.Series(model.feature_importances_, index=X.columns)
feature_importances.nlargest(5).plot(kind='bar')
plt.show()
```



In [135…
```
#Train test model for predicting the Y value
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(X,Y, test_size=0.2)
```

In [136…
```
X_train
```

Out[136…

	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	transmission_Manual	owner_Fourth & Above Owner
911	58500	8	1	0	0	0	0	0	0	0
774	75000	5	1	0	0	0	0	0	1	0
3544	78000	7	1	0	0	0	1	0	1	0
1330	80000	6	1	0	0	0	1	0	0	0

	km_driven	duration	fuel_Diesel	fuel_Electric	fuel_LPG	fuel_Petrol	seller_type_Individual	seller_type_Trustmark Dealer	transmission_Manual	owner_Fourth & Above Owner
225	213000	13	1	0	0	0	1	0	1	0
...	...	...	...	...	...	...	...	...	...	...
1090	14000	4	1	0	0	0	0	0	1	0
3375	40000	7	1	0	0	0	1	0	1	0
3948	25000	6	0	0	0	1	1	0	1	0
1116	223000	16	1	0	0	0	1	0	1	0
4045	120000	12	1	0	0	0	1	0	1	0

3472 rows × 13 columns

In [137...
```
X_train.shape
```

Out[137...
```
(3472, 13)
```

In [145...
```
from sklearn.ensemble import RandomForestRegressor
rf_random=RandomForestRegressor()
```

In [143...
```
from sklearn.model_selection import RandomizedSearchCV
```

In [148...
```
rf_random.fit(X_train, y_train)
```

Out[148...
```
RandomForestRegressor()
```

In [149...
```
predictions=rf_random.predict(X_test)
```

In [150...
```
predictions
```

Out[150...
```
array([ 149237.77747619,  142109.91666667,  705186.01787102,
        324499.72      ,  756821.54761905,  815599.37509524,
        959800.        ,  270242.42063492,  198826.66666667,
        359914.32052381,  278206.83333333,  791091.66666667,
```

```
 437556.41025641,  418993.91053391,  151050.        ,
 174850.        ,  276060.72333333,  128809.78      ,
 409040.48259039,  118902.13285714,  400265.07936508,
 224820.02813853,  198565.8509329 , 2610582.14285714,
 275237.24186111,  332158.33333333,  162286.66666667,
 566966.47343254,  570395.71428571,  639605.58221429,
 236118.92857143,  611600.        ,  707800.        ,
 142749.99      ,   83559.56333333,  541260.        ,
 163500.        ,  458570.8606838 ,  317648.11111111,
 374937.61904762,  409763.33333333,  272053.65800866,
 451086.23809524,  256440.        ,  334414.16666667,
 374937.61904762,  696519.99416667,  504939.83585859,
 329487.        ,  403111.66666667,  451383.09333333,
 207990.3773254 ,  201673.33333333,  415449.3608254 ,
1347609.52380952,  287264.88095238,  242056.97380952,
  94995.83333333,  385000.        , 2482791.66666667,
 226533.18823304,  414064.45268665, 1249860.        ,
1535561.66666667,  209232.38095238,  757578.35907531,
 613925.        ,  157359.99      ,  346549.0471082 ,
 598633.03932179,  235950.        ,  939634.41982038,
 275588.45959596,  173623.08080808,  102793.33333333,
  58469.95      ,  300992.45238095,  406333.47185714,
 246499.94      ,  162049.99714286,  443556.41025641,
 218349.21      ,  954531.22515476,  119750.        ,
  79690.0352381 ,  239000.        ,  560171.13095238,
 215150.        ,  566083.03809524,  196500.        ,
 741615.88342213,   79420.        ,  911370.        ,
 190786.66666667,  576376.74603175,  336154.75809524,
 549425.        , 1008319.58333333,  587180.83333333,
 488939.96      ,  192561.66666667,  597340.        ,
 929127.8142203 , 1388799.71      ,   80439.11      ,
 403856.19047619,  279662.01298701,  294304.46428571,
 628841.66666667,  149098.94666667,  435721.36324603,
 946913.35775336,  414064.45268665,   99243.88888889,
 372573.33333333,  493317.42660118,  365370.        ,
 414500.        ,  327596.26190476, 1027081.17965368,
 280422.5       ,  374174.38744118,  484758.92857143,
1234759.98      ,   95275.99      ,  228934.96      ,
 108774.57142857,  246875.89285714,  549550.        ,
 176827.26773227,  199453.00307576,  452720.        ,
 382550.        ,  515850.        ,  341962.49951122,
 406333.47185714,  190786.66666667, 2600000.        ,
 449664.7987013 ,  611951.47025019,  245969.52      ,
 344800.        ,  209095.3015873 ,  676462.6984127 ,
  93901.1525974 ,  440613.33333333,  695119.69      ,
 482216.36904762,  129166.66666667,  376049.94      ,
 458672.04856746,  414064.45268665,  502764.59040959,
```

```
 544998.47619048,  600000.        ,  131550.        ,
 374174.38744118,  317075.23559524,  611951.47025019,
 489219.08616667,  437260.        ,  676952.09859307,
 208500.95238095,   62355.        ,  264796.19047619,
 118095.83333333,  286613.5       ,  465744.11172161,
 233748.79770238, 2101628.57142857, 1027081.17965368,
1708421.03174603,  646597.3992674 ,  240359.62      ,
 904680.        ,  705186.01787102,  254853.71666667,
  96663.96248196,  518492.17222222,  278206.83333333,
 797045.        ,  360437.4925    ,  206905.        ,
 535000.        ,  247100.        ,  239000.        ,
 455385.81709957,  265967.99694697,  570722.61904762,
 706147.1488234 ,  400920.5952381 ,  135107.34511905,
 287264.88095238,  492706.93444444,  199788.21428571,
 372718.21206349,  161935.99      ,  162186.63095238,
1516500.        ,  529799.9875    , 1047400.        ,
 362450.        ,  327596.26190476,  895690.        ,
 390249.97      ,  327773.25333333,  337198.27675506,
 757578.35907531,  220748.33333333,  190500.        ,
 670000.        ,  467438.02655678, 1841050.        ,
 792670.83333333,  131758.38238095,  374174.38744118,
 560171.13095238,  552714.98      ,   72195.        ,
 160840.        ,  311428.22333333,  566083.03809524,
 283679.998     ,  652167.68759019,  437360.        ,
 122577.60714286,  297429.88095238,  539224.99      ,
 439423.30592857,  700051.08563792,  474197.38095238,
1038750.        , 1300000.        ,  317310.        ,
 288949.99      ,  217950.        , 1608359.92      ,
 265981.67207792,  457759.34053571,  324237.85714286,
 496999.16666667,  556886.78321429,   66191.        ,
 470997.39180264,  356004.88095238,  550874.853     ,
 281250.        ,  194448.32333333,  651986.81271861,
 553959.99      ,  742576.38430556, 3777350.        ,
4950000.        ,  308958.61111111, 1300000.        ,
 508621.41857143,  550011.39747997,  652167.68759019,
 141400.        ,  155799.75      ,  502764.59040959,
 451437.5       ,  541500.        ,  312112.13233333,
1918359.97      ,  809434.84126984,  503900.        ,
 319373.45238095, 2520433.33333333,  301273.22333333,
 467438.02655678,  119100.        ,  277600.        ,
 457759.34053571,  550490.        ,  572779.98      ,
 220748.33333333,  236200.        ,  229543.57142857,
 536093.33333333,  670000.        ,  240270.        ,
 308405.71035714,  117700.        ,  484544.915     ,
 313242.        ,  297570.        ,  783046.03142857,
  62118.33333333,  337255.94704762,  362210.7547619 ,
 217852.45475   ,  337198.27675506,  155762.22633333,
```

```
 750000.        ,  396146.76004744,   95878.95      ,
 509960.        ,  646597.3992674 ,  526080.8022533 ,
 128438.0952381 ,  153803.0952381 ,  528349.74      ,
 677139.36507937,  206892.5       ,  464750.        ,
 403111.66666667,  497250.        ,  153522.35714286,
 522310.        ,  351033.21206349,  650985.66666667,
 679900.        ,  821398.78571429,  540349.42      ,
1769490.        ,  432339.86      ,  852210.        ,
 318540.50828505,  136284.        ,  526080.8022533 ,
 239001.9047619 ,  517401.15766667,  279662.01298701,
 284192.84798535,  222412.16666667,  121290.        ,
 220748.33333333,  357800.        , 3632200.        ,
  85150.        ,  234206.71603175,  605749.18039033,
 446200.        ,  455385.81709957, 2443750.        ,
 490248.01587302,  470578.37301587,  535000.        ,
 536559.85      ,  210000.        ,  149098.94666667,
 154197.5       ,  287264.88095238,  561246.52380952,
 815599.37509524,  364676.59666667,  483749.99      ,
 125742.49667   ,  162320.        ,  278206.83333333,
 386266.66666667,  360474.21428571,  900742.22222222,
 339449.89      ,  350818.1403254 ,  119242.        ,
  77601.18547619,  217449.3       ,  541092.19047619,
 487504.08696859,  700051.08563792,   95456.78571429,
 330073.33333333,  557876.66083333,  490950.        ,
 281316.8947619 ,  135579.97      ,  599330.15873016,
  82403.30333333,  284521.76323676,  194870.        ,
 458570.8606838 ,  336154.75809524,  333919.21897547,
 531430.55555556,   70200.        , 1580000.        ,
  82896.57692308, 1008319.58333333,  482216.36904762,
 593114.46428571, 1921440.        ,  199499.55      ,
 356183.33333333,  276359.38      ,  781692.17622655,
2151550.        ,  277126.80952381,  116056.19047619,
 367714.        ,  858794.04761905,  321658.33333333,
 528027.24331224,  246365.        ,  396146.76004744,
  98839.98      ,  242450.61562338,  455270.        ,
 658439.41757143,  150089.32907143,  814440.83333333,
 450575.34492063,  411635.45071429,  365370.        ,
  77810.        ,  248632.66333333,  760000.        ,
 217668.7175308 ,  462200.        ,  903200.        ,
 670000.        ,  375802.56285381,  388400.        ,
1156300.        ,  108737.49      , 1619597.14285714,
 259053.22333333,   91490.        ,  176827.26773227,
 244715.        , 1565300.        ,  369127.06285714,
 253983.9345754 ,  410326.61666667,  742459.29706349,
 409040.48259039,  185580.        ,  170880.        ,
 196600.90435714,  443720.        ,  351762.5       ,
 217668.7175308 ,  467416.66666667,  493317.42660118,
```

```
 254177.38095238,  281316.8947619 ,  211576.73      ,
 253983.9345754 ,  638283.71031746,  437556.41025641,
1300000.        , 1580000.        ,  176827.26773227,
 173584.99833333,  197462.61904762,  540976.38888889,
 347371.41422551,  368443.74625   ,  548807.5       ,
 245338.9181688 ,   95230.        ,  179674.96031746,
 175769.45305952,  254086.91444444, 4950000.        ,
 313720.77897727,  315170.82      , 1543191.66666667,
 493317.42660118,  661050.        ,  144116.6540873 ,
 438195.90845429,  439423.30592857, 1248130.95238095,
 849491.19425   ,  467438.02655678,   84240.        ,
 518549.66699967,  762000.        ,  237749.99      ,
 231822.30264833,  541080.98      ,  490960.89646465,
 643919.54365079,  470950.        ,  819779.28571429,
 158614.14285714,  543380.        ,  494549.99      ,
 141279.99      ,  376672.73881674, 1025248.01587302,
 486834.48967699,  393850.        ,  313804.14666667,
 932557.5       ,  111406.3       ,  220988.57142857,
 815599.37509524, 1580000.        , 1541989.83      ,
 410181.06892857,  472945.47063384, 1736130.        ,
 260396.62666667,  602234.71560859,  487504.08696859,
 758101.1575873 ,  781692.17622655,  200801.92460317,
 625000.        ,  953508.7989788 ,  693059.10533911,
 500590.        ,  757578.35907531,  623208.38400433,
 948026.56666667,  236857.36326582,  734640.        ,
 104030.        ,  120770.80966667,  472909.96      ,
 409178.95825397,  261449.72      ,  738078.48124098,
 327596.26190476,  309821.55610428,  470997.39180264,
  90132.5       ,  415449.3608254 , 2283190.        ,
 260990.35714286,  947323.32      ,  622036.50793651,
 199788.21428571,  406333.47185714, 1493100.        ,
 858950.        ,  210266.65666667,  188777.5       ,
 560171.13095238,  243009.52795455,  101649.28571429,
 845498.59111111,  253983.9345754 ,  439423.30592857,
 281316.8947619 ,  428924.47      ,  116387.14285714,
 513000.58694841,   92899.99      ,  418870.        ,
 677323.5271645 ,  231822.30264833,  338770.        ,
1209186.66666667,  602234.71560859, 2702450.        ,
 281790.        ,  104167.5       ,  376419.28571429,
 750000.        , 1293542.85542857,   83493.33333333,
 189388.        ,  520000.        ,  740269.16166667,
2600000.        ,  351659.15412698,  493317.42660118,
 201271.81795238,  939634.41982038,  368443.74625   ,
 387269.92444495,  302833.16666667,  517053.61095238,
 101050.        ,  553074.995     ,  410791.88852814,
 714662.79761905,  473080.        ,  259036.73414164,
 167620.        ,  480053.26333333,  178799.64      ,
```

```
 923722.03555952,  469185.95238095,  549974.78354978,
 472909.96      ,  270242.42063492,  278345.17857143,
 652167.68759019,  450000.        ,  628057.26190476,
 457759.34053571,  277426.54666667,  175250.95238095,
 385375.9489122 ,  321146.66666667,  504329.85      ,
 312691.66666667,  760000.        ,  480099.93      ,
 221616.66666667,   77650.        ,  437556.41025641,
 472460.        ,  300138.33333333,  744654.17204762,
 187154.59137302, 1210878.33333333,  760000.        ,
 351033.21206349,  425777.53777778,  193020.        ,
 590930.39215686, 1565300.        ,  403111.66666667,
 261205.42321429,  646543.75      ,  289269.78      ,
 337386.58666667,  878412.5       ,  154667.5       ,
 236857.36326582,  224820.02813853,  120155.4047619 ,
 550194.66666667,  334850.47186147,  176184.99714286,
 305399.99      ,  833938.89792308,  351033.21206349,
 822700.        ,  458349.91      ,  123089.05      ,
 538010.55555556,  279718.85209235,  113049.9       ,
 285228.44904762,  390300.        ,  191285.68333333,
 529799.9875    , 1064340.        ,  250561.98      ,
 315048.32195671,  778850.        ,  371470.47619048,
 587579.85      ,  647475.62925   ,  229999.        ,
 296750.6997619 ,  257400.        ,  155799.75      ,
 320755.        ,  410791.88852814,  462826.65666667,
 385000.        ,  415449.3608254 ,  131482.63095238,
 188271.86507937,  340959.23809524,   95878.95      ,
 865849.92      ,  133225.        ,  550194.66666667,
3276766.65666667,  236857.36326582, 1307941.66666667,
 186739.99      , 2046599.94      ,  413400.11904762,
1252140.        ,  414661.38461538,  342350.        ,
 700350.        ,  100858.53174603,  356183.33333333,
  59180.        ,  843436.9047619 ,  932557.5       ,
 541052.84404751,  368443.74625   ,  135046.66666667,
 162286.66666667,  817359.99      ,  600000.        ,
3800000.        ,  119750.        ,  600000.        ,
 482047.14285714,  408449.28571429,  437556.41025641,
 287647.8045119 ,  472909.96      ,  243416.91558442,
 739030.        ,  449114.96033333,  308841.66666667,
 151189.75      ,  236118.92857143,  387716.47123149,
 344083.69047619,  125861.45403571,  749282.11442857,
 467438.02655678,  622036.50793651, 1028353.76906089,
 300138.33333333,  214339.76634921, 1028353.76906089,
 543287.5       ,  119242.        ,  560327.78994228,
 281316.8947619 ,  528027.24331224,  190575.95238095,
 296003.5447619 , 1340550.        ,  374174.38744118,
 175769.45305952,  864440.3159188 ,  231822.30264833,
 698715.49360714,  531430.55555556,  342016.66666667,
```

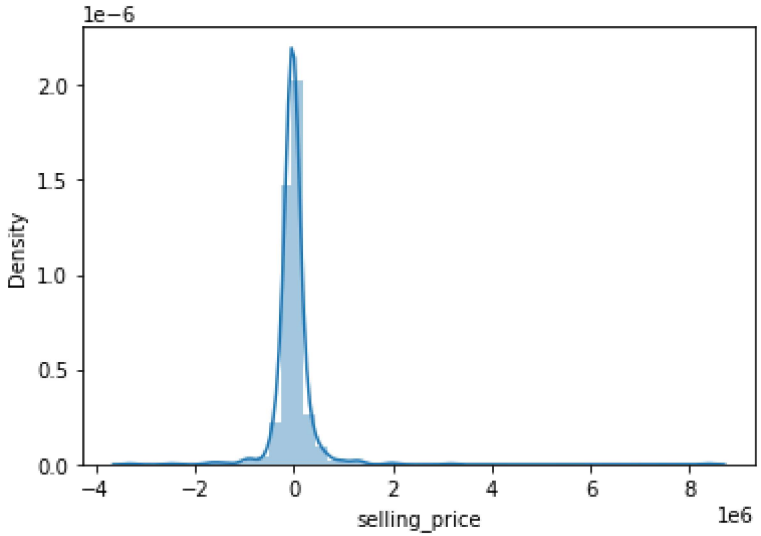
```
 718100.        ,  616881.42099567,  410791.88852814,
 705186.01787102,  611951.47025019,  133656.224     ,
 273640.90727273,  308958.61111111, 2154061.11111111,
 561967.85714286,  135018.0952381 ,  438195.90845429,
 273139.36904762,  192741.62733333,  445548.33333333,
 933740.96062302,   80195.83333333,  517248.47404144,
 158575.        ,  584420.        ,  797196.66666667,
 245338.9181688 ,  653849.58      ,  570610.        ,
 150268.66      ,  424809.99      ,  337198.27675506,
2760583.33333333,  300138.33333333,  209096.46793651,
1347609.52380952,   91419.36147186,  396146.76004744,
 383300.        ,  560171.13095238, 1308585.27777778,
 575152.52484921,  327374.34669048,  246457.63777778,
 622036.50793651,  672473.33333333,  208053.57142857,
 760000.        ,  116304.78736905, 2185356.51666667,
2381142.85714286,  386750.        ,  418993.91053391,
  53340.        ,  321146.66666667,  208331.81318681,
 517401.15766667,  316260.        ,  229410.71428571,
 241440.95238095,  449577.5       ,  750000.        ,
 437970.        ,  471770.        , 1300000.        ,
 335310.38275613,  239000.        ,  443700.        ,
 469161.88492063,  366391.66666667,  315048.32195671,
 369711.25      ,  436695.91883333,  114899.16666667,
 789040.36796537,  414509.33133333,  284521.76323676,
 111638.33333333,  947447.97619048,  109695.        ,
 706879.52241146,  452720.        ,  184894.74673127,
 119750.        , 1873390.        ,   67250.66666667,
1239276.10868254,  293460.83333333,  514941.35714356,
 959450.        ,  156890.        ,  953508.7989788 ,
 229999.        ,  513454.75190476,  189981.94150072,
 243800.        ,  405119.75409094,  809137.25188889,
 239657.5       ,  228360.        ,  518549.66699967,
 112658.33333333,  161935.99      ,  191865.        ,
 132589.08730159,   74046.66666667,  440653.33333333,
 374174.38744118,   70200.        ,  374174.38744118,
 410791.88852814,  190308.33333333, 1758349.67      ,
  69932.5       ,  646543.75      ,  301799.99      ,
 414381.66666667,  109695.        ,  280693.44904762,
 133656.224     ,  639779.17005159,  466650.        ,
 267844.96      ,  242450.61562338, 1575250.        ,
 188514.60137302,  101910.        ,  296003.5447619 ,
 158614.14285714,  208331.81318681,  458672.04856746,
 334850.47186147,   88937.14285714,  122577.60714286,
 300992.45238095,  368140.        ,  436400.        ,
1725550.        ,  395782.73015873,  189025.        ,
  96769.99      ,  175271.66666667,  592579.90079365,
 670000.        ,  149098.94666667,  549974.78354978,
```

```
       705186.01787102,  216799.67      ,  351727.40384615,
        92538.0952381 ,  804779.92      ,  336031.34923814,
        93901.1525974 ,  211576.73      ,  232923.33333333,
       340959.23809524])
```

In [151…
```
sns.distplot(y_test-predictions)
```

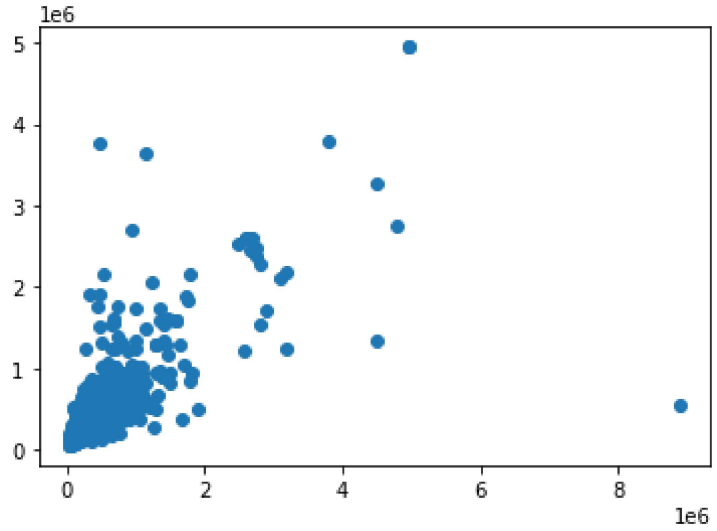
```
C:\Users\fambareen\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function a
nd will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexi
bility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[151…
```
<AxesSubplot:xlabel='selling_price', ylabel='Density'>
```



In [152…
```
plt.scatter(y_test,predictions)
```

Out[152…
```
<matplotlib.collections.PathCollection at 0x27aed532eb0>
```



In [ ]: