In [191]:
```
import pandas as pd
```

In [192]:
```
data=pd.read_csv("/home/placement/Downloads/fiat500.csv") #reads the file
```

In [193]:
```
data.describe()
```

Out[193]:

| | ID | engine_power | age_in_days | km | previous_owners | lat | lon | price |
|---|---|---|---|---|---|---|---|---|
| **count** | 1538.000000 | 1538.000000 | 1538.000000 | 1538.000000 | 1538.000000 | 1538.000000 | 1538.000000 | 1538.000000 |
| **mean** | 769.500000 | 51.904421 | 1650.980494 | 53396.011704 | 1.123537 | 43.541361 | 11.563428 | 8576.003901 |
| **std** | 444.126671 | 3.988023 | 1289.522278 | 40046.830723 | 0.416423 | 2.133518 | 2.328190 | 1939.958641 |
| **min** | 1.000000 | 51.000000 | 366.000000 | 1232.000000 | 1.000000 | 36.855839 | 7.245400 | 2500.000000 |
| **25%** | 385.250000 | 51.000000 | 670.000000 | 20006.250000 | 1.000000 | 41.802990 | 9.505090 | 7122.500000 |
| **50%** | 769.500000 | 51.000000 | 1035.000000 | 39031.000000 | 1.000000 | 44.394096 | 11.869260 | 9000.000000 |
| **75%** | 1153.750000 | 51.000000 | 2616.000000 | 79667.750000 | 1.000000 | 45.467960 | 12.769040 | 10000.000000 |
| **max** | 1538.000000 | 77.000000 | 4658.000000 | 235000.000000 | 4.000000 | 46.795612 | 18.365520 | 11100.000000 |

In [194]:
```
data1=data.drop(['ID','lat','lon'],axis=1) #removes the specific columns from dataframe
```

In [195]:
```
data1
```

Out[195]:

| | model | engine_power | age_in_days | km | previous_owners | price |
|---|---|---|---|---|---|---|
| **0** | lounge | 51 | 882 | 25000 | 1 | 8900 |
| **1** | pop | 51 | 1186 | 32500 | 1 | 8800 |
| **2** | sport | 74 | 4658 | 142228 | 1 | 4200 |
| **3** | lounge | 51 | 2739 | 160000 | 1 | 6000 |
| **4** | pop | 73 | 3074 | 106880 | 1 | 5700 |
| **...** | ... | ... | ... | ... | ... | ... |
| **1533** | sport | 51 | 3712 | 115280 | 1 | 5200 |
| **1534** | lounge | 74 | 3835 | 112000 | 1 | 4600 |
| **1535** | pop | 51 | 2223 | 60457 | 1 | 7500 |
| **1536** | lounge | 51 | 2557 | 80750 | 1 | 5990 |
| **1537** | pop | 51 | 1766 | 54276 | 1 | 7900 |

1538 rows × 6 columns

In [196]:
```
data2=pd.get_dummies(data1) #encodes the string into bits
```

In [197]:
```
data2
```

Out[197]:

| | engine_power | age_in_days | km | previous_owners | price | model_lounge | model_pop | model_sport |
|---|---|---|---|---|---|---|---|---|
| **0** | 51 | 882 | 25000 | 1 | 8900 | 1 | 0 | 0 |
| **1** | 51 | 1186 | 32500 | 1 | 8800 | 0 | 1 | 0 |
| **2** | 74 | 4658 | 142228 | 1 | 4200 | 0 | 0 | 1 |
| **3** | 51 | 2739 | 160000 | 1 | 6000 | 1 | 0 | 0 |
| **4** | 73 | 3074 | 106880 | 1 | 5700 | 0 | 1 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **1533** | 51 | 3712 | 115280 | 1 | 5200 | 0 | 0 | 1 |
| **1534** | 74 | 3835 | 112000 | 1 | 4600 | 1 | 0 | 0 |
| **1535** | 51 | 2223 | 60457 | 1 | 7500 | 0 | 1 | 0 |
| **1536** | 51 | 2557 | 80750 | 1 | 5990 | 1 | 0 | 0 |
| **1537** | 51 | 1766 | 54276 | 1 | 7900 | 0 | 1 | 0 |

1538 rows × 8 columns

In [198]:
```
data2.shape #shows num of rows and columns
```

Out[198]:
```
(1538, 8)
```

In [199]:
```
y=data2['price']#predicted value removed feom data frame
x=data2.drop(['price'],axis=1)
```

In [200]:
```
y #prices only will display
```

Out[200]:
```
0       8900
1       8800
2       4200
3       6000
4       5700
        ...
1533    5200
1534    4600
1535    7500
1536    5990
1537    7900
Name: price, Length: 1538, dtype: int64
```

In [201]:
```
x #no prices
```

Out[201]:

| | engine_power | age_in_days | km | previous_owners | model_lounge | model_pop | model_sport |
|---|---|---|---|---|---|---|---|
| **0** | 51 | 882 | 25000 | 1 | 1 | 0 | 0 |
| **1** | 51 | 1186 | 32500 | 1 | 0 | 1 | 0 |
| **2** | 74 | 4658 | 142228 | 1 | 0 | 0 | 1 |
| **3** | 51 | 2739 | 160000 | 1 | 1 | 0 | 0 |
| **4** | 73 | 3074 | 106880 | 1 | 0 | 1 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1533** | 51 | 3712 | 115280 | 1 | 0 | 0 | 1 |
| **1534** | 74 | 3835 | 112000 | 1 | 1 | 0 | 0 |
| **1535** | 51 | 2223 | 60457 | 1 | 0 | 1 | 0 |
| **1536** | 51 | 2557 | 80750 | 1 | 1 | 0 | 0 |
| **1537** | 51 | 1766 | 54276 | 1 | 0 | 1 | 0 |

1538 rows × 7 columns

In [202]:
```
!pip3 install scikit-learn  #to install sklearn library
```

```
Requirement already satisfied: scikit-learn in ./anaconda3/lib/python3.10/site-packages (1.2.1)
Requirement already satisfied: joblib>=1.1.1 in ./anaconda3/lib/python3.10/site-packages (from scikit-lear
n) (1.1.1)
Requirement already satisfied: scipy>=1.3.2 in ./anaconda3/lib/python3.10/site-packages (from scikit-learn)
(1.10.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in ./anaconda3/lib/python3.10/site-packages (from sciki
t-learn) (2.2.0)
Requirement already satisfied: numpy>=1.17.3 in ./anaconda3/lib/python3.10/site-packages (from scikit-lear
n) (1.23.5)
```

In [203]:
```
om sklearn.model_selection import train_test_split
train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)#splitting into training and
```

In [204]:
```
x_test.head(5)#shows the testing column values
```

Out[204]:

| | engine_power | age_in_days | km | previous_owners | model_lounge | model_pop | model_sport |
|---|---|---|---|---|---|---|---|
| **481** | 51 | 3197 | 120000 | 2 | 0 | 1 | 0 |
| **76** | 62 | 2101 | 103000 | 1 | 0 | 1 | 0 |
| **1502** | 51 | 670 | 32473 | 1 | 1 | 0 | 0 |
| **669** | 51 | 913 | 29000 | 1 | 1 | 0 | 0 |
| **1409** | 51 | 762 | 18800 | 1 | 1 | 0 | 0 |

In [205]:
```
y_test.head(5)
```

Out[205]:
```
481     7900
76      7900
1502    9400
669     8500
1409    9700
Name: price, dtype: int64
```

In [206]:
```
x_train.head(5)
```

Out[206]:

| | engine_power | age_in_days | km | previous_owners | model_lounge | model_pop | model_sport |
|---|---|---|---|---|---|---|---|
| **527** | 51 | 425 | 13111 | 1 | 1 | 0 | 0 |
| **129** | 51 | 1127 | 21400 | 1 | 1 | 0 | 0 |
| **602** | 51 | 2039 | 57039 | 1 | 0 | 1 | 0 |
| **331** | 51 | 1155 | 40700 | 1 | 1 | 0 | 0 |
| **323** | 51 | 425 | 16783 | 1 | 1 | 0 | 0 |

In [207]:
```
y_train.head(5)
```

Out[207]:
```
527    9990
129    9500
602    7590
331    8750
323    9100
Name: price, dtype: int64
```

In [208]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression() #creating object of linearregression
reg.fit(x_train,y_train) #training and fitting LR object using training data
```

Out[208]:
```
LinearRegression()
```
**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**
**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

In [209]:
```
ypred=reg.predict(x_test) #prediction of values(x_tesy*reg)
```

In [210]:
```
ypred
```

```
array([ 8398.59735084,  9680.77538859,  4334.81943405, 10015.00600846,
        9850.72458719,  7864.73798641, 10072.71245374, 10552.64805598,
       10253.47474908,  6861.80736606,  6484.22649656, 10374.62123623,
        8426.37409382,  5447.47569851,  9914.20077691,  4687.39013431,
        7885.32100747,  5431.00822998,  9911.86294348, 10390.16991322,
        9680.84745901,  8844.57815539,  7764.08471024,  4257.54640953,
        9882.76503303, 10341.35258769,  5736.4484335 , 10179.87154436,
        9501.423448  ,  7997.3181334 ,  5532.33458288,  9894.57834738,
       10437.97459358,  6381.35845844,  9591.23555726,  9574.27908517,
       10322.30715736,  9501.22785499,  9789.955758  ,  9593.26549752,
        6775.82788536,  7915.34831306, 10389.98590521, 10351.58343315,
        7381.32686464,  9966.53983093, 10430.87188433, 10554.43156462,
       10285.85574963, 10035.88086558,  9526.63034431,  7742.78157141,
        9297.64938364, 10051.42272678, 10004.81256571,  9985.84167026,
        9374.6573594 ,  9561.57499854,  9754.94184269,  9819.85893758,
        8780.31447831,  6255.99008069,  6281.53627686,  8190.88781577,
        8588.91394592,  6566.97963218,  6850.70237466,  5511.29438169,
        8119.97866315,  9847.74830838,  7775.93862032,  9875.05509733,
       10121.29366536,  5791.92464084,  9835.42728501, 10043.91426822,
        8027.28015259,  4527.22080416, 10609.02444098,  3808.29240951,
```

In [211]:
```
from sklearn.metrics import r2_score#efficiency
r2_score(y_test,ypred) #ypred is predicted value
```

Out[211]:
```
0.8415526986865394
```

In [212]:
```
from sklearn.metrics import mean_squared_error#to calculate rms value
```

In [213]:
```
mean_squared_error(ypred,y_test)
```

Out[213]:
```
581887.727391353
```

In [214]:
```
import math
```

In [215]:
```
n=581887.727391353 #to find square root
math.sqrt(n)
```

Out[215]:
```
762.8156575420782
```

In [216]:
```
#from sklearn.metrics import accuracy_score
#accuracy_score(y_test,ypred)
```

In [224]:
```
results=pd.DataFrame(columns=['Price','Predicted'])
results['Price']=y_test
results['Predicted']=ypred
results=results.reset_index()
results['Id']=results.index
results.head(15)
```

Out[224]:

| | index | Price | Predicted | Id |
|---|---|---|---|---|
| **0** | 481 | 7900 | 5867.650338 | 0 |
| **1** | 76 | 7900 | 7133.701423 | 1 |
| **2** | 1502 | 9400 | 9866.357762 | 2 |
| **3** | 669 | 8500 | 9723.288745 | 3 |
| **4** | 1409 | 9700 | 10039.591012 | 4 |
| **5** | 1414 | 9900 | 9654.075826 | 5 |
| **6** | 1089 | 9900 | 9673.145630 | 6 |
| **7** | 1507 | 9950 | 10118.707281 | 7 |
| **8** | 970 | 10700 | 9903.859527 | 8 |
| **9** | 1198 | 8999 | 9351.558284 | 9 |
| **10** | 1088 | 9890 | 10434.349636 | 10 |
| **11** | 576 | 7990 | 7732.262557 | 11 |
| **12** | 965 | 7380 | 7698.672401 | 12 |
| **13** | 1488 | 6800 | 6565.952404 | 13 |
| **14** | 1432 | 8900 | 9662.901035 | 14 |

In [226]:
```
results['final']=results.apply(lambda row:row.Price - row.Predicted,axis=1)#difference between prices and pr
```

In [227]:
```
results.head(15)
```

Out[227]:

| | index | Price | Predicted | Id | final |
|---|---|---|---|---|---|
| **0** | 481 | 7900 | 5867.650338 | 0 | 2032.349662 |
| **1** | 76 | 7900 | 7133.701423 | 1 | 766.298577 |
| **2** | 1502 | 9400 | 9866.357762 | 2 | -466.357762 |
| **3** | 669 | 8500 | 9723.288745 | 3 | -1223.288745 |
| **4** | 1409 | 9700 | 10039.591012 | 4 | -339.591012 |
| **5** | 1414 | 9900 | 9654.075826 | 5 | 245.924174 |
| **6** | 1089 | 9900 | 9673.145630 | 6 | 226.854370 |
| **7** | 1507 | 9950 | 10118.707281 | 7 | -168.707281 |
| **8** | 970 | 10700 | 9903.859527 | 8 | 796.140473 |
| **9** | 1198 | 8999 | 9351.558284 | 9 | -352.558284 |
| **10** | 1088 | 9890 | 10434.349636 | 10 | -544.349636 |
| **11** | 576 | 7990 | 7732.262557 | 11 | 257.737443 |
| **12** | 965 | 7380 | 7698.672401 | 12 | -318.672401 |
| **13** | 1488 | 6800 | 6565.952404 | 13 | 234.047596 |
| **14** | 1432 | 8900 | 9662.901035 | 14 | -762.901035 |

In [ ]:
```

```