In [2]:
```
import pandas as pd
```

In [3]:
```
data=pd.read_csv("/home/placement/Desktop/fiat500.csv")
```

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

Out[4]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	lounge	51	882	25000	1	44.907242	8.611560	8900
1	2	pop	51	1186	32500	1	45.666359	12.241890	8800
2	3	sport	74	4658	142228	1	45.503300	11.417840	4200
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6000
4	5	pop	73	3074	106880	1	41.903221	12.495650	5700

In [5]:
```
data1=data.drop(['lat','lon','ID'],axis=1)
```

In [6]:
```
data1.head()
```

Out[6]:

	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

In [7]:
```
data=pd.get_dummies(data)
```

In [8]:
```
data
```

Out[8]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price	model_lounge	model_pop	model_sport
0	1	51	882	25000	1	44.907242	8.611560	8900	1	0	0
1	2	51	1186	32500	1	45.666359	12.241890	8800	0	1	0
2	3	74	4658	142228	1	45.503300	11.417840	4200	0	0	1
3	4	51	2739	160000	1	40.633171	17.634609	6000	1	0	0
4	5	73	3074	106880	1	41.903221	12.495650	5700	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
1533	1534	51	3712	115280	1	45.069679	7.704920	5200	0	0	1
1534	1535	74	3835	112000	1	45.845692	8.666870	4600	1	0	0
1535	1536	51	2223	60457	1	45.481541	9.413480	7500	0	1	0
1536	1537	51	2557	80750	1	45.000702	7.682270	5990	1	0	0
1537	1538	51	1766	54276	1	40.323410	17.568270	7900	0	1	0

1538 rows × 11 columns

In [9]:
```
data.shape
```

Out[9]:
```
(1538, 11)
```

In [10]:
```
data
```

Out[10]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price	model_lounge	model_pop	model_sport
0	1	51	882	25000	1	44.907242	8.611560	8900	1	0	0
1	2	51	1186	32500	1	45.666359	12.241890	8800	0	1	0
2	3	74	4658	142228	1	45.503300	11.417840	4200	0	0	1
3	4	51	2739	160000	1	40.633171	17.634609	6000	1	0	0
4	5	73	3074	106880	1	41.903221	12.495650	5700	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
1533	1534	51	3712	115280	1	45.069679	7.704920	5200	0	0	1
1534	1535	74	3835	112000	1	45.845692	8.666870	4600	1	0	0
1535	1536	51	2223	60457	1	45.481541	9.413480	7500	0	1	0
1536	1537	51	2557	80750	1	45.000702	7.682270	5990	1	0	0
1537	1538	51	1766	54276	1	40.323410	17.568270	7900	0	1	0

1538 rows × 11 columns

In [11]:
```
y=data['price']
```

In [12]:
```
x=data.drop('price',axis=1)
```

In [13]:
```
y
```

Out[13]:
```
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 [14]:
```
!pip install scikit-learn
```

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

In [15]:
```
from sklearn.model_selection import train_test_split
```

In [16]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33,random_state=42)
```

In [17]:
```
x_test.head(5)
```

Out[17]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	model_lounge	model_pop	model_sport
481	482	51	3197	120000	2	40.174702	18.167629	0	1	0
76	77	62	2101	103000	1	45.797859	8.644440	0	1	0
1502	1503	51	670	32473	1	41.107880	14.208810	1	0	0
669	670	51	913	29000	1	45.778591	8.946250	1	0	0
1409	1410	51	762	18800	1	45.538689	9.928310	1	0	0

In [18]:
```
x_train.shape
```

Out[18]:
```
(1030, 10)
```

In [19]:
```
x_train.head()
```

Out[19]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	model_lounge	model_pop	model_sport
527	528	51	425	13111	1	45.022388	7.58602	1	0	0
129	130	51	1127	21400	1	44.332531	7.54592	1	0	0
602	603	51	2039	57039	1	40.748241	14.52835	0	1	0
331	332	51	1155	40700	1	42.143860	12.54016	1	0	0
323	324	51	425	16783	1	41.903221	12.49565	1	0	0

In [20]:
```
y_train.head()
```

Out[20]:
```
527    9990
129    9500
602    7590
331    8750
323    9100
Name: price, dtype: int64
```

Type *Markdown* and LaTeX: α^2

In [21]:
```
y_test.head()
```

Out[21]:
```
481     7900
76      7900
1502    9400
669     8500
1409    9700
Name: price, dtype: int64
```

In [22]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
```

Out[22]:
```
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 [23]:
```
ypred=reg.predict(x_test)
```

In [24]:
```
ypred
```

Out[24]:
```
array([ 5819.19308764,  7248.82914161,  9741.8936974 ,  9798.98033074,
       10055.00624601,  9551.4955679 ,  9758.01743879, 10122.9778365 ,
        9654.9661814 ,  9251.1403257 , 10478.09512253,  7807.3005255 ,
        7705.15873781,  6295.63244894,  9545.40486313, 10422.92177704,
        9616.90811615,  7756.9171161 ,  4893.88454414, 10581.46142719,
       10465.24078346, 10443.29318231,  7518.43696046, 10028.21911459,
        6990.73118896,  8989.86900819,  4823.51364349,  6989.03118684,
        7822.83203734,  9683.17944083,  7344.21343132,  5341.43860798,
        5420.78405336,  5092.38401339,  8971.44357515,  5702.81242412,
        9920.16285466,  8334.58448277,  6220.93323723,  8389.23958511,
        9695.84208061,  6859.59630725,  9101.22635456, 10063.22592995,
        8621.83915759, 10175.06753933,  9063.21918346,  8867.24865352,
        7094.44228184,  9058.37693565,  9474.82390731, 10406.09102832,
       10112.65006224,  6820.90463865,  9700.36507783,  9382.18149429,
        9632.57617775, 10553.81356008,  9847.21129432,  7247.16814789,
        9990.23331336,  7084.23300123,  9977.34233656,  7245.01115798,
        6490.89305576,  9737.86785115,  9853.54349825,  8568.7125607 ,
        8506.81438703,  6484.69051659,  7883.1895563 ,  6870.28308427,
        8263.36833348, 10551.03496347,  7434.71134313,  8637.85174602,
```

In [25]:
```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[25]:
```
0.8428319728488683
```

In [26]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[26]:
```
577189.6736608233
```

In [27]:
```
import math
a=577189.6736608233
print(math.sqrt(a))
```

```
759.7300005007195
```

In [28]:
```
y_test.head()
```

Out[28]:
```
481     7900
76      7900
1502    9400
669     8500
1409    9700
Name: price, dtype: int64
```

In [29]:
```
Results= pd.DataFrame(columns=['price','predicate'])
Results['price']=y_test
Results['predicate']=ypred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(15)
```

Out[29]:

	index	price	predicate	Id
0	481	7900	5819.193088	0
1	76	7900	7248.829142	1
2	1502	9400	9741.893697	2
3	669	8500	9798.980331	3
4	1409	9700	10055.006246	4
5	1414	9900	9551.495568	5
6	1089	9900	9758.017439	6
7	1507	9950	10122.977837	7
8	970	10700	9654.966181	8
9	1198	8999	9251.140326	9
10	1088	9890	10478.095123	10
11	576	7990	7807.300526	11
12	965	7380	7705.158738	12
13	1488	6800	6295.632449	13
14	1432	8900	9545.404863	14

In [30]:
```
Results['diff_price']=Results.apply(lambda row:row.price-row.predicate,axis=1)
```

In [31]:
```
Results.head(15)
```

Out[31]:

	index	price	predicate	Id	diff_price
0	481	7900	5819.193088	0	2080.806912
1	76	7900	7248.829142	1	651.170858
2	1502	9400	9741.893697	2	-341.893697
3	669	8500	9798.980331	3	-1298.980331
4	1409	9700	10055.006246	4	-355.006246
5	1414	9900	9551.495568	5	348.504432
6	1089	9900	9758.017439	6	141.982561
7	1507	9950	10122.977837	7	-172.977837
8	970	10700	9654.966181	8	1045.033819
9	1198	8999	9251.140326	9	-252.140326
10	1088	9890	10478.095123	10	-588.095123
11	576	7990	7807.300526	11	182.699474
12	965	7380	7705.158738	12	-325.158738
13	1488	6800	6295.632449	13	504.367551
14	1432	8900	9545.404863	14	-645.404863

In [ ]:
```

```