

```
In [1]: import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: data = pd.read_csv("/home/placement/Downloads/fiat500.csv")
```

```
In [3]: data.describe()
```

```
Out[3]:
```

	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 [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
```

```
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
...	...	...	...	...	...	...
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 [7]: data3=data.loc[(data.model=='lounge')]
data3
```

Out[7]:

	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
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6000
6	7	lounge	51	731	11600	1	44.907242	8.611560	10750
7	8	lounge	51	1521	49076	1	41.903221	12.495650	9190
11	12	lounge	51	366	17500	1	45.069679	7.704920	10990
...	...	...	...	...	...	...	...	...	...
1528	1529	lounge	51	2861	126000	1	43.841980	10.515310	5500
1529	1530	lounge	51	731	22551	1	38.122070	13.361120	9900
1530	1531	lounge	51	670	29000	1	45.764648	8.994500	10800
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990

1094 rows × 9 columns

```
In [8]: data=pd.get_dummies(data1)
data
```

Out[8]:

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

1538 rows × 8 columns

data.shape

```
In [9]: data2=pd.get_dummies(data3)
data2
```

Out[9]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price	model_lounge
0	1	51	882	25000	1	44.907242	8.611560	8900	True
3	4	51	2739	160000	1	40.633171	17.634609	6000	True
6	7	51	731	11600	1	44.907242	8.611560	10750	True
7	8	51	1521	49076	1	41.903221	12.495650	9190	True
11	12	51	366	17500	1	45.069679	7.704920	10990	True
...	...	...	...	...	...	...	...	...	...
1528	1529	51	2861	126000	1	43.841980	10.515310	5500	True
1529	1530	51	731	22551	1	38.122070	13.361120	9900	True
1530	1531	51	670	29000	1	45.764648	8.994500	10800	True
1534	1535	74	3835	112000	1	45.845692	8.666870	4600	True
1536	1537	51	2557	80750	1	45.000702	7.682270	5990	True

1094 rows × 9 columns

In [10]: data2.shape

Out[10]: (1094, 9)

```
In [11]: y=data2['price']  
x=data2.drop('price',axis=1)
```

```
In [12]: from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [13]: x\_test.head(5)

```
Out[13]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	model_lounge	
	676	677	51	762	18609	1	41.572239	13.33369	True
	215	216	51	701	25000	1	44.988739	9.01050	True
	146	147	51	4018	152900	1	43.067532	12.55155	True
	1319	1320	51	731	20025	1	41.689281	13.25494	True
	1041	1042	51	640	38231	1	41.107880	14.20881	True

```
In [14]: x_train.shape
```

```
Out[14]: (732, 8)
```

```
In [15]: y_train
```

```
Out[15]: 441      8980
         701     10300
         695      5880
         1415    10490
         404      9499
         ...
         459     10850
         654      5900
         189     10000
         1455      9400
         1218      8900
         Name: price, Length: 732, dtype: int64
```

```
In [16]: y_test.head()
```

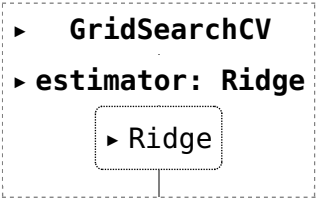
```
Out[16]: 676      10250
         215      9790
         146      5500
         1319     9900
         1041      8900
         Name: price, dtype: int64
```

```
In [17]: y_train.shape
```

Out[17]: (732,)

```
In [18]: from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge
alpha=[1e-15,1e-10,1e-8,1e-4,1e-3,1e-2,1,5,10,20,30]
ridge=Ridge()
parameters={'alpha':alpha}
ridge_regressor=GridSearchCV(ridge,parameters)
ridge_regressor.fit(x_train,y_train)
```

Out[18]:



```

  ▸ GridSearchCV
    ▸ estimator: Ridge
      ▸ Ridge
```

```
In [19]: ridge_regressor.best_params_
```

Out[19]: {'alpha': 30}

```
In [20]: ridge = Ridge(alpha=30)
ridge.fit(x_train,y_train)
y_pred_ridge=ridge.predict(x_test)
```

```
In [21]: from sklearn.metrics import mean_squared_error
Ridge_Error = mean_squared_error(y_pred_ridge,y_test)
Ridge_Error
```

Out[21]: 529111.0455362239

```
In [22]: from sklearn.metrics import r2_score
r2_score(y_test,y_pred_ridge)
```

Out[22]: 0.8343797517106646

```
In [23]: import seaborn as sns
Results=pd.DataFrame(columns=['Actual','Predicted'])
Results['Actual']=y_test
Results['Predicted']=y_pred_ridge
Results=Results.reset_index()
Results['ID']=Results.index
Results.head(10)
```

```
Out[23]:
```

	index	Actual	Predicted	ID
0	676	10250	9912.601754	0
1	215	9790	10141.748493	1
2	146	5500	4775.235521	2
3	1319	9900	9870.926966	3
4	1041	8900	9630.417885	4
5	1425	9500	8697.092014	5
6	409	10450	10265.822884	6
7	617	9790	10293.851867	7
8	1526	9300	8614.349738	8
9	1010	4600	5749.673567	9

```
In [24]: import matplotlib.pyplot as plt
sns.lineplot(x='ID',y='Actual',data=Results.head(50))
sns.lineplot(x='ID',y='Predicted',data=Results.head(50))
plt.plot()
```

```
Out[24]: []
```



