In [1]:
```
#In this file,I performed linear_regression,ridge_regression,elastic_net and checked the error of them
```

In [2]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

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

In [4]:
```
data
```

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
...	...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	5200
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
1535	1536	pop	51	2223	60457	1	45.481541	9.413480	7500
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990
1537	1538	pop	51	1766	54276	1	40.323410	17.568270	7900

1538 rows × 9 columns

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

Out[5]:

	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 [6]:
```
data1['model']=data1['model'].map({'lounge':1,'pop':2,'sport':3})
```

In [7]:
```
data1.head()
```

Out[7]:

	model	engine_power	age_in_days	km	previous_owners	price
0	1	51	882	25000	1	8900
1	2	51	1186	32500	1	8800
2	3	74	4658	142228	1	4200
3	1	51	2739	160000	1	6000
4	2	73	3074	106880	1	5700

In [8]:
```
data1.shape
```

Out[8]:
```
(1538, 6)
```

In [9]:
```
y=data['price']
X=data1.drop('price',axis=1)
```

In [10]:
```
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)
```

Linear regression

In [11]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(X_train,y_train)
```

Out[11]:
```
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 [12]:
```
ypred=reg.predict(X_test)
ypred
```

```
array([ 6349.19977644,  4992.62377497,  9157.57378424,  9843.66900765,
        5300.60403791,  5481.29193836, 10061.18107826,  8216.74437963,
       10365.58751936,  6942.04538818,  6646.91921519,  5747.75873375,
        8993.90183905,  9927.58506055, 10363.24676269,  9396.86589768,
        9076.99796116, 10215.3021386 , 10383.15669206, 10126.03200513,
        9691.06492408,  9388.16610785, 10320.28226749,  5168.15653507,
        9748.75001733,  6076.81735982,  9002.6037807 , 10194.30552754,
        9367.47951184,  9906.72847636,  8421.42411454,  8493.56714248,
        7513.9076281 , 10525.09542651, 10437.87520301, 10100.9899145 ,
       10212.09402583,  6948.1022781 ,  9602.27041522, 10385.65404439,
        9629.55224111,  8066.17394592,  9681.08600118,  7955.02256935,
       10372.69178007,  9154.76238436,  5812.28163235,  6805.34855395,
        8358.805056  , 10425.80776532,  9923.80795634,  9781.53621223,
       10586.81080578,  7672.28847193,  6792.75056277,  8166.24671727,
       10283.85072091,  8816.21082546,  8489.96779496,  9558.69157657,
        9721.52332944, 10138.28055739, 10267.73796468,  7250.72913511,
        9704.08855317,  6272.77563612,  7885.75505518,  9483.71113195,
        5035.58827017,  9328.81619306,  9955.6070855 , 10069.12877714,
        6349.16007365,  9831.73092008,  9196.20340881,  5346.71715499,
        5528.38460908,  4612.02299242, 10173.80266537,  9999.55096614,
```

Ridge regression

In [13]:
```
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[13]:
```
GridSearchCV(estimator=Ridge(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.0001, 0.001, 0.01, 1,
                                   5, 10, 20, 30]})
```

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 [14]:
```
ridge_regressor.best_params_
```

Out[14]:
```
{'alpha': 30}
```

In [15]:
```
ridge=Ridge(alpha=30)
ridge.fit(X_train,y_train)
y_pred_ridge=ridge.predict(X_test)
```

Elastic Net

In [16]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import ElasticNet

elastic = ElasticNet()

parameters = {'alpha': [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20]}

elastic_regressor = GridSearchCV(elastic, parameters)

elastic_regressor.fit(X_train, y_train)
```

Out[16]:
```
GridSearchCV(estimator=ElasticNet(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.0001, 0.001, 0.01, 1,
                                   5, 10, 20]})
```

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 [17]:
```
elastic_regressor.best_params_
```

Out[17]:
```
{'alpha': 0.01}
```

In [18]:
```
elastic=ElasticNet(alpha=0.1)
elastic.fit(X_train,y_train)
y_pred_elastic=elastic.predict(X_test)
```

Errors of linreg,ridreg,elasnet

In [19]:
```
from sklearn.metrics import mean_squared_error#Linear regression MSE
mean_squared_error(ypred,y_test)
```

Out[19]:
```
593504.2888137382
```

In [20]:
```
from sklearn.metrics import mean_squared_error#Ridge Regression MSE
Ridge_Error=mean_squared_error(y_pred_ridge,y_test)
Ridge_Error
```

Out[20]: 590569.9121697354

In [21]:
```
from sklearn.metrics import mean_squared_error#Elastic net MSE
elastic_Error=mean_squared_error(y_pred_elastic,y_test)
elastic_Error
```

Out[21]: 588989.0937739909

Accuracies of linreg,ridreg,elasnet

In [22]:
```
from sklearn.metrics import r2_score#Linear regression Accuracy
r2_score(y_test,ypred)
```

Out[22]: 0.8383895235218549

In [23]:
```
from sklearn.metrics import r2_score#Ridge Regression Accuracy
r2_score(y_test,y_pred_ridge)
```

Out[23]: 0.8391885506165899

In [25]:
```
from sklearn.metrics import r2_score#Elastic net Accuracy
r2_score(y_test,y_pred_elastic)
```

Out[25]: 0.8396190054910987

In [ ]:
```

```