

# 1).Advertising

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso,RidgeCV,Ridge
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import ElasticNet
```

```
In [2]: df=pd.read_csv(r"C:\Users\HP\Downloads\Advertising.csv")
df
```

```
Out[2]:
```

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

200 rows × 4 columns

## Linear Regression

```
In [4]: feature=df.columns[0:3]
target=df.columns[-1]
x=df[feature].values
y=df[target].values
```

```
In [5]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

0.8293443335615364

## Ridge

```
In [ ]: ridgeReg=Lasso(alpha=10)
ridgeReg.fit(x_train,y_train)
train_score_ridge=ridgeReg.score(x_train,y_train)
test_score_ridge=ridgeReg.score(x_test,y_test)
print("\n Lasso Model \n")
print("train score for ridge model is {}".format(train_score_ridge))
print("test score for ridge model is {}".format(test_score_ridge))
```

## Lasso

```
In [7]: lassoReg=Lasso(alpha=10)
lassoReg.fit(x_train,y_train)
train_score_lasso=lassoReg.score(x_train,y_train)
test_score_lasso=lassoReg.score(x_test,y_test)
print("\n Lasso Model \n")
print("train score for lasso model is {}".format(train_score_lasso))
print("test score for lasso model is {}".format(test_score_lasso))
```

Lasso Model

train score for lasso model is 0.8998900992678511  
test score for lasso model is 0.8227168476751803

## ElasticNet

```
In [8]: regr=ElasticNet()
regr.fit(x,y)
print(regr.coef_)
print(regr.intercept_)
```

[0.05440081 0.1046715 0. ]  
4.696191158087224

```
In [9]: y_pred_elastic=regr.predict(x_train)
```

```
In [10]: mean_squared_error=np.mean((y_pred_elastic-y_train)**2)
print("Mean Squared Error on test set",mean_squared_error)
```

Mean Squared Error on test set 2.62976797697343

## 2).vehicle selection

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso,RidgeCV,Ridge
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import ElasticNet
```

```
In [2]: df=pd.read_csv(r"C:\Users\HP\Downloads\fiat500_VehicleSelection_Dataset (5).csv")
df
```

Out[2]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
0	1	lounge	51	882	25000	1	44.907242	8.611560
1	2	pop	51	1186	32500	1	45.666359	12.241890
2	3	sport	74	4658	142228	1	45.503300	11.417840
3	4	lounge	51	2739	160000	1	40.633171	17.634609
4	5	pop	73	3074	106880	1	41.903221	12.495650
...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870
1535	1536	pop	51	2223	60457	1	45.481541	9.413480
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270
1537	1538	pop	51	1766	54276	1	40.323410	17.568270

1538 rows × 9 columns



## Linear Regression

```
In [5]: feature=df.columns[3:6]
        target=df.columns[-1]
        x=df[feature].values
        y=df[target].values
```

```
In [6]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
        regr=LinearRegression()
        regr.fit(x_train,y_train)
        print(regr.score(x_test,y_test))
```

0.8582418048544962

## Ridge

```
In [7]: ridgeReg=Lasso(alpha=10)
        ridgeReg.fit(x_train,y_train)
        train_score_ridge=ridgeReg.score(x_train,y_train)
        test_score_ridge=ridgeReg.score(x_test,y_test)
        print("\n Lasso Model \n")
        print("train score for ridge model is {}".format(train_score_ridge))
        print("test score for ridge model is {}".format(test_score_ridge))
```

Lasso Model

train score for ridge model is 0.8346372139730224  
test score for ridge model is 0.858367839830971

## Lasso

```
In [8]: lassoReg=Lasso(alpha=10)
        lassoReg.fit(x_train,y_train)
        train_score_lasso=lassoReg.score(x_train,y_train)
        test_score_lasso=lassoReg.score(x_test,y_test)
        print("\n Lasso Model \n")
        print("train score for lasso model is {}".format(train_score_lasso))
        print("test score for lasso model is {}".format(test_score_lasso))
```

Lasso Model

train score for lasso model is 0.8346372139730224  
test score for lasso model is 0.858367839830971

## ElasticNet

```
In [9]: regr=ElasticNet()  
        regr.fit(x,y)  
        print(regr.coef_)  
        print(regr.intercept_)
```

```
[-0.87262858 -0.01820307  4.45095932]  
10983.666980630238
```

```
In [10]: y_pred_elastic=regr.predict(x_train)
```

```
In [11]: mean_squared_error=np.mean((y_pred_elastic-y_train)**2)  
        print("Mean Squared Error on test set",mean_squared_error)
```

```
Mean Squared Error on test set 611371.307963663
```

```
In [ ]:
```