HW 4

Bryan Mui 506021334

In this dataset, there are 6 predictors X_1...,X_6 and 1 response variable. There are 3 predictors that should not be included in the regression model. Please apply LASSO to figure out those variables

In [33]:
```
# import
import pandas as pd # type: ignore
from sklearn.model_selection import train_test_split # type: ignore
from sklearn.linear_model import Lasso # type: ignore
```

In [34]:
```
df = pd.read_csv('DF_LASSO.csv')
print(df.shape)
df.head()
```

```
(1000, 8)
```

Out[34]:

	Unnamed: 0	X_1	X_2	X_3	X_4	X_5	X_6	Y
0	0	-6.394606	-7.305487	-6.824976	-7.248253	4.890602	-5.595641	-6.231352
1	1	-9.610495	-6.184431	-9.256275	-8.800796	-8.456382	-10.867742	-22.526547
2	2	-0.735629	-1.965451	1.420229	-0.653601	-8.352232	-0.529532	-9.229125
3	3	4.498679	-1.973594	5.109008	4.659780	-3.902677	4.093653	0.377013
4	4	-1.595928	-9.023107	-2.153985	-2.877928	8.012144	-1.016704	-0.540507

In [35]:
```
# split the data 70-30
x = df[['X_1', 'X_2', 'X_3', 'X_4', 'X_5', 'X_6']]
y = df['Y']
# split the training and testing data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=777)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(700, 6)
(300, 6)
(700,)
(300,)
```

In [36]:
```
# standardize the data so that lasso can compare the features more effectively
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_train_scale = scaler.fit_transform(x_train)
x_test_scale = scaler.transform(x_test)

print(x_train_scale)
print(x_test_scale)
```

```
[[ 1.34099278 -0.3137271   1.36284311  1.24701832 -1.65278517  1.2569556 ]
 [-1.12978166  0.63323593 -1.42247517 -1.14193977  0.19060871 -0.87327622]
 [ 0.57359796  0.09157047  0.53531181  0.54837947 -0.93235326  0.76274614]
 ...
 [ 0.87529363  0.17726518  0.91349385  0.97506891 -0.11249501  1.05045473]
 [-1.10295211  0.91081167 -1.05909659 -1.00700534 -0.31168293 -1.31667324]
 [-0.69126793  0.05978308 -0.96914841 -0.78966224 -1.50831773 -1.15962283]]
[[-1.00972121 -0.17211761 -1.12689469 -1.17503268 -0.23024407 -0.97406974]
 [-0.63184477  0.72245959 -0.36924473 -0.04938881 -0.91871048 -0.38102932]
 [-0.00932703  1.03378624 -0.02140949 -0.10124237  0.3599867  -0.04841046]
 ...
 [ 1.67174819  1.00059634  1.41482724  1.49628001  1.43207387  1.77995892]
 [ 1.00720301 -1.00919743  1.00370442  0.88836527 -1.05168642  0.68077894]
 [-0.4894885   0.17324024 -0.4174558  -0.70537229  0.25894102 -0.28460656]]
```

In [37]:
```
# Train the Lasso model
m1 = Lasso(alpha=1)
m1.fit(x_train_scale, y_train)

# Evaluate the model
from sklearn.metrics import (r2_score)
y_pred = m1.predict(x_test_scale)
r2 = r2_score(y_test, y_pred)
print(f"R-squared: {r2}")

print("Lasso Coefficients:", m1.coef_)

coef = pd.DataFrame({
    'feature': x.columns,
    'coefficient': m1.coef_
})
coef.head(10)
```

```
R-squared: 0.9664702635867173
Lasso Coefficients: [4.75697658 4.62344221 0.         0.         4.77275264 0.        ]
```

Out[37]:

	feature	coefficient
0	X_1	4.756977
1	X_2	4.623442
2	X_3	0.000000
3	X_4	0.000000
4	X_5	4.772753
5	X_6	0.000000

LASSO shrinks the coefficients that are not important to 0, so the variables that should not be included in the model are X_3, X_4, and X_6