Diabetes (R^2 Score)

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
dia=pd.read_csv("diabetes.csv")
dia.head()
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

In [3]:

```
dia.isna().sum()
```

Out[3]:

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [4]:

```
x=dia.drop("Outcome",axis=1)
```

In [5]:

```
y=dia.Outcome
```

In [6]:

```
for i in range(len(dia)):
    if dia['Insulin'][i]==0:
        dia['Insulin'][i]=np.mean(dia.Insulin)
```

```
C:\Users\Nitin Thakur\Anaconda3\lib\site-packages\ipykernel_launcher.py:3: S
ettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  This is separate from the ipykernel package so we can avoid doing imports
 until
```

In [7]:

```
dia['SkinThickness'].replace(0,np.mean(dia['SkinThickness']),inplace=True)
```

In [8]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

In [9]:

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
```

In [10]:

```
model.fit(x_train,y_train)
```

Out[10]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fal
se)
```

In [11]:

```
ypred=model.predict(x_test)
```

In [12]:

```
model.score(x_train,y_train)
```

Out[12]:

```
0.310994466010l0194
```

In [13]:

```
from sklearn.metrics import r2_score
print(r2_score(y_test,ypred))
```

```
0.2658087735702177
```

In [14]:

```
dia.head()
```

Out[14]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35.000000	79	33.6	0.62
1	1	85	66	29.000000	79	26.6	0.35
2	8	183	64	20.536458	80	23.3	0.67
3	1	89	66	23.000000	94	28.1	0.16
4	0	137	40	35.000000	168	43.1	2.28

In [43]:

```
from sklearn.linear_model import Ridge

from sklearn.model_selection import cross_val_score
import numpy as np

ridge=Ridge(alpha=.8).fit(x_train,y_train)

score_dr=cross_val_score(ridge,x_train,y_train,cv=5,scoring='r2')
print('CV Mean: ',np.mean(score_dr))
print("STD : ",np.std(score_dr))
print('\n')
```

```
CV Mean:  0.2762029530736161
STD :  0.047900647776724856
```

In [44]:

```
from sklearn.linear_model import Lasso

from sklearn.model_selection import cross_val_score
import numpy as np

lasso=Lasso(alpha=1).fit(x_train,y_train)

score_dr=cross_val_score(lasso,x_train,y_train,cv=5,scoring='r2')
print('CV Mean: ',np.mean(score_dr))
print("STD : ",np.std(score_dr))
print('\n')
```

```
CV Mean:  0.19959021470844365
STD :  0.05113434017487795
```