import Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

import Linear Regression

In [2]:

```
from sklearn.linear_model import LogisticRegression
```

In [3]:

```
lgr=LogisticRegression()
```

Select Required data from certain columns

In [4]:

```
a=pd.read_csv("health.csv")
a
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFun
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

768 rows × 9 columns

In [5]:

```
c=a.dropna()
c
```

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFun
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

768 rows × 9 columns

In [6]:

```
c.columns
```

Out[6]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insuli
n',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [18]:

```
fm=c.iloc[:,0:9]
tv=c.iloc[:,-1]
```

Shape

In [19]:

```
fm.shape
```

Out[19]:

```
(768, 9)
```

In [20]:

```
tv.shape
```

Out[20]:

```
(768,)
```

To make the data in order (feature matrix)

In [21]:

```
from sklearn.preprocessing import StandardScaler
```

In [22]:

```
fs=StandardScaler().fit_transform(fm)
```

Imply Logistic Regression

In [23]:

```
lgr.fit(fs,tv)
```

Out[23]:

```
LogisticRegression()
```

Prediction

In [25]:

```
ab=[[3,90,543,34,56,45,24,45,23]]
```

In [26]:

```
pre=lgr.predict(ab)
```

In [27]:

```
print(pre)
```

```
[1]
```

To check the output var we have got

In [28]:

```
lgr.classes_
```

Out[28]:

```
array([0, 1], dtype=int64)
```

Prediction in Probablity value

In [34]:

```
lgr.predict_proba(ab)[0][0]
```

Out[34]:

```
0.0
```