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("bmi.csv")
a
```

Out[4]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

In [5]:

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

Out[5]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

In [6]:

```
c.columns
```

Out[6]:

```
Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')
```

In [7]:

```
fm=c[['Height', 'Weight']]
tv=c[['Index']]
```

Shape

In [8]:

```
fm.shape
```

Out[8]:

```
(500, 2)
```

In [9]:

```
tv.shape
```

Out[9]:

```
(500, 1)
```

To make the data in order (feature matrix)

In [10]:

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

In [11]:

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

Imply Logistic Regression

In [12]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63:
DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using r
avel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.
py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sc
ikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression (https://scikit-learn.org/stable/modules/linear_model.html#logisti
c-regression)
  n_iter_i = _check_optimize_result(
```

Out[12]:

```
LogisticRegression()
```

Prediction

In [13]:

```
ab=[[3,90]]
```

In [14]:

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

In [15]:

```
print(pre)
```

```
[5]
```

To check the output var we have got

In [16]:

```
lgr.classes_
```

Out[16]:

```
array([0, 1, 2, 3, 4, 5], dtype=int64)
```

Prediction in Probablity value

In [17]:

```
lgr.predict_proba(ab)[0][1]
```

Out[17]:

```
1.4446356598602695e-39
```

In [ ]:

```

```