

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [2]: from sklearn.linear_model import LogisticRegression
```

```
In [3]: df=pd.read_csv("C6_bmi.csv")
df
```

Out[3]:

	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 [4]: df=df.dropna()
df
```

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

	Gender	Height	Weight	Index
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

In [5]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   Gender  500 non-null     object
 1   Height  500 non-null     int64
 2   Weight  500 non-null     int64
 3   Index   500 non-null     int64
dtypes: int64(3), object(1)
memory usage: 19.5+ KB
```

In [6]: `df.columns`

Out[6]: `Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')`

In [7]: `feature_matrix=df[['Height', 'Weight']]`  
`target_vector=df[ 'Index']`

In [8]: `feature_matrix.shape`

Out[8]: `(500, 2)`

In [9]: `target_vector.shape`

Out[9]: `(500,)`

In [10]: `from sklearn.preprocessing import StandardScaler`

In [11]: `fs=StandardScaler().fit_transform(feature_matrix)`

In [12]: `logr=LogisticRegression()`  
`logr.fit(fs,target_vector)`

Out[12]: `LogisticRegression()`

In [13]: `observation=[[1,2]]`

```
In [14]: prediction=logr.predict(observation)
         print(prediction)
```

```
[5]
```

```
In [15]: logr.classes_
```

```
Out[15]: array([0, 1, 2, 3, 4, 5], dtype=int64)
```

```
In [16]: logr.predict_proba(observation)[0][0]
```

```
Out[16]: 5.5956697582538237e-11
```

```
In [17]: logr.predict_proba(observation)
```

```
Out[17]: array([[5.59566976e-11, 6.05990036e-10, 1.19071465e-07, 4.99471797e-05,
                  2.03791363e-02, 9.79570797e-01]])
```