

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LogisticRegression
```

```
In [18]: df=pd.read_csv("C6_bmi - C6_bmi (1).csv")
df
```

Out[18]:

	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 [19]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 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: 15.8+ KB
```

```
In [20]: df1=df.fillna(value=0)
df1
```

Out[20]:

	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 [33]: df2=df1[["Height", "Weight", "Index"]]
df2
```

Out[33]:

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

500 rows × 3 columns

```
In [41]: feature_matrix=df2.iloc[:,0:2]
target_vector=df2.iloc[:, -1]
```

```
In [42]: feature_matrix.shape
```

```
Out[42]: (500, 2)
```

```
In [43]: target_vector.shape
```

```
Out[43]: (500,)
```

```
In [44]: from sklearn.preprocessing import StandardScaler
```

```
In [45]: fs=StandardScaler().fit_transform(feature_matrix)
```

```
In [46]: logr =LogisticRegression()  
logr.fit(fs,target_vector)
```

```
Out[46]: LogisticRegression()
```

```
In [49]: observation=[[1.4,2.3]]
```

```
In [50]: prediction=logr.predict(observation)  
print(prediction)
```

```
[5]
```

```
In [51]: logr.classes_
```

```
Out[51]: array([0, 1, 2, 3, 4, 5], dtype=int64)
```

```
In [52]: logr.predict_proba(observation)[0][0]
```

```
Out[52]: 2.603071212363146e-11
```

```
In [17]: logr.predict_proba(observation)[0][1]
```

```
Out[17]: 0.999830636327398
```

```
In [ ]:
```