

7ak7we1uk

August 1, 2023

```
[22]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

```
[23]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
[24]: df=pd.read_csv("/content/drive/MyDrive/mydatasets/C6_bmi.csv")
df
```

```
[24]:
```

	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 x 4 columns]

```
[25]: df.head()
```

```
[25]:
```

	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
```

1 Data Cleaning and Data Preprocessing

```
[26]: 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
```

```
[27]: df.describe()
```

```
[27]:
```

	Height	Weight	Index
count	500.000000	500.000000	500.000000
mean	169.944000	106.000000	3.748000
std	16.375261	32.382607	1.355053
min	140.000000	50.000000	0.000000
25%	156.000000	80.000000	3.000000
50%	170.500000	106.000000	4.000000
75%	184.000000	136.000000	5.000000
max	199.000000	160.000000	5.000000

```
[28]: df.columns
```

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

```
[29]: feature_matrix = df.iloc[:,1:3]
      target_vector = df.iloc[:,-1]
```

```
[30]: fs = StandardScaler().fit_transform(feature_matrix)
      logr = LogisticRegression()
      logr.fit(fs,target_vector)
```

```
[30]: LogisticRegression()
```

```
[31]: observation=[[1,2]]
      prediction = logr.predict(observation)
      print(prediction)
```

[5]

```
[32]: logr.classes_
```

```
[32]: array([0, 1, 2, 3, 4, 5])
```

```
[33]: logr.predict_proba(observation)
```

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