

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

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

```
In [3]: df=pd.read_csv("C5 health.csv").dropna()
```

df

0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

```
In [4]: df.dropna(inplace=True)
```

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Pregnancies            768 non-null   int64
1   Glucose                768 non-null   int64
2   BloodPressure          768 non-null   int64
3   SkinThickness          768 non-null   int64
4   Insulin                768 non-null   int64
5   BMI                   768 non-null   float64
6   DiabetesPedigreeFunction 768 non-null   float64
7   Age                   768 non-null   int64
8   Outcome                768 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 60.0 KB
```

```
In [7]: feature_matrix = df[['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age',
target_vector = df['Outcome']
```

```
In [8]: feature_matrix.shape
```

```
Out[8]: (768, 8)
```

```
In [11]: target_vector.shape
```

```
Out[11]: (768,)
```

```
In [12]: from sklearn.preprocessing import StandardScaler
```

```
In [13]: fs = StandardScaler().fit_transform(feature_matrix)
```

```
In [14]: logr = LogisticRegression()
logr.fit(fs,target_vector)
```

```
Out[14]: LogisticRegression()
```

```
In [15]: feature_matrix.shape
```

```
Out[15]: (768, 8)
```

```
In [16]: target_vector.shape
```

Out[16]: (768,)

```
In [17]: from sklearn.preprocessing import StandardScaler
```

```
In [18]: fs = StandardScaler().fit_transform(feature_matrix)
```

```
In [19]: logr = LogisticRegression()
logr.fit(fs,target_vector)
```

Out[19]: LogisticRegression()

```
In [20]: observation=df[['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age']]
```

```
In [21]: prediction = logr.predict(observation)
          prediction
```

[illegible]

```
In [22]: logr.classes_
```

```
Out[22]: array([0, 1], dtype=int64)
```

```
In [23]: logit.predict_proba(observation)[0][1]
```

Out[23]: 1.0

In []: