

```
In [1]: import pandas as pd
import numpy as np
df=pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/Diabetes.csv')
df.head()
```

```
Out[1]:
```

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age	diabetes
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

```
In [2]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 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   diastolic       768 non-null   int64  
 3   triceps        768 non-null   int64  
 4   insulin        768 non-null   int64  
 5   bmi            768 non-null   float64 
 6   dpf            768 non-null   float64 
 7   age            768 non-null   int64  
 8   diabetes       768 non-null   int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

```
In [4]: df.describe()
```

```
Out[4]:
```

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age	diabetes
--	-------------	---------	-----------	---------	---------	-----	-----	-----	----------

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age	diabetes
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000

In [7]: `df.shape`

Out[7]: (768, 9)

In [8]: `df.columns`

Out[8]: Index(['pregnancies', 'glucose', 'diastolic', 'triceps', 'insulin', 'bmi',
 'dpf', 'age', 'diabetes'],
 dtype='object')

In [10]: `df['diabetes'].value_counts()`

Out[10]: 0 500
 1 268
 Name: diabetes, dtype: int64

In [11]: `df.groupby('diabetes').mean()`

Out[11]:

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age
diabetes								
0	3.298000	109.980000	68.184000	19.664000	68.792000	30.304200	0.429734	31.190000
1	4.865672	141.257463	70.824627	22.164179	100.335821	35.142537	0.550500	37.067164

```
In [12]: y=df['diabetes']
```

```
In [13]: y.shape
```

```
Out[13]: (768,)
```

```
In [14]: y
```

```
Out[14]: 0      1
         1      0
         2      1
         3      0
         4      1
         ..
        763     0
        764     0
        765     0
        766     1
        767     0
        Name: diabetes, Length: 768, dtype: int64
```

```
In [15]: X=df[['pregnancies','glucose','diastolic','triceps','insulin','bmi','dpf','age']]
         X.shape
```

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

```
In [16]: X
```

```
Out[16]:
```

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33
...

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age
763	10	101	76	48	180	32.9	0.171	63
764	2	122	70	27	0	36.8	0.340	27
765	5	121	72	23	112	26.2	0.245	30
766	1	126	60	0	0	30.1	0.349	47
767	1	93	70	31	0	30.4	0.315	23

```
In [17]: from sklearn.preprocessing import MinMaxScaler
mm=MinMaxScaler()
X = mm.fit_transform(X)
X
```

```
Out[17]: array([[0.35294118, 0.74371859, 0.59016393, ..., 0.50074516, 0.23441503,
0.48333333],
[0.05882353, 0.42713568, 0.54098361, ..., 0.39642325, 0.11656704,
0.16666667],
[0.47058824, 0.91959799, 0.52459016, ..., 0.34724292, 0.25362938,
0.18333333],
...,
[0.29411765, 0.6080402 , 0.59016393, ..., 0.390462 , 0.07130658,
0.15 ],
[0.05882353, 0.63316583, 0.49180328, ..., 0.4485842 , 0.11571307,
0.43333333],
[0.05882353, 0.46733668, 0.57377049, ..., 0.45305514, 0.10119556,
0.03333333]])
```

```
In [18]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3, random_state=2529)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
Out[18]: ((537, 8), (231, 8), (537,), (231,))
```

```
In [23]: from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train, y_train)
```

```
Out[23]: LogisticRegression()
```

```
In [24]: y_pred= lr.predict(X_test)
         y_pred.shape
```

```
Out[24]: (231,)
```

```
In [21]: y_pred
```

```
Out[21]: array([0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
                0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
                0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0,
                0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1,
                0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1,
                0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
                0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
                0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
                1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1], dtype=int64)
```

```
In [26]: lr.predict_proba(X_test)
```

```
Out[26]:
```

```
In [27]: from sklearn.metrics import confusion_matrix, classification_report
         confusion_matrix(y_test,y_pred)
```

```
Out[27]: array([[133, 12],
               [ 46, 40]], dtype=int64)
```

```
In [28]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.74	0.92	0.82	145
1	0.77	0.47	0.58	86
accuracy			0.75	231
macro avg	0.76	0.69	0.70	231
weighted avg	0.75	0.75	0.73	231

```
In [31]: X_new=df.sample(1)
         X_new.shape
```

```
Out[31]: (1, 9)
```

```
In [35]: X_new = df[['pregnancies','glucose','diastolic','triceps','insulin','bmi','dpf','age']]
         X_new
```

```
Out[35]:
```

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33
...
763	10	101	76	48	180	32.9	0.171	63

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age
764	2	122	70	27	0	36.8	0.340	27
765	5	121	72	23	112	26.2	0.245	30
766	1	126	60	0	0	30.1	0.349	47
767	1	93	70	31	0	30.4	0.315	23

In []: