

✓ Step 1 :- Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler
```

✓ Step 2 :- import the dataset

```
col_names = ['Pregnant', 'Glucose', 'BP', 'Skin', 'Insulin', 'BMI', 'Pedigree', 'Age', 'Label']
```

```
df = pd.read_csv('pima-indians-diabetes.csv', header=None, names=col_names)
```

```
df
```



	Pregnant	Glucose	BP	Skin	Insulin	BMI	Pedigree	Age	Label
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

✓ Step 3 :- Identify the Dependent and Independent Variables

```
Feature_Columns = ['Pregnant', 'Glucose', 'BP', 'Skin', 'Insulin', 'BMI', 'Pedigree', 'Age']
```

```
X = df[Feature_Columns]
```

```
y = df['Label']
```

✓ Step 4 :- Train the Data

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

✓ Step 5 :- Feature Scaling

```
Scaler = StandardScaler()
```

```
X_train = Scaler.fit_transform(X_train)
```


```
X_test = Scaler.fit_transform(X_test)
```

✓ Step 6 :- Model Building

```
Model = LogisticRegression()
```

```
Model.fit(X_train,y_train) # Training the Data
```

```


  ▾ LogisticRegression ⓘ ?
  LogisticRegression()

```

```
y_pred = Model.predict(X_test)
```

```
Confusion_Matrix = confusion_matrix(y_test,y_pred)
```

✓ Step 7 :- Classification Report

```
Accuracy = accuracy_score(y_test,y_pred)
```

```
print(F"Accuracy Score :{Accuracy*100:2f}%" )
```

```



  Accuracy Score :78.571429%

```

```
Names = ['Without_Diabetes','With_Diabetes']
```

```
print(classification_report(y_test,y_pred,target_names=Names))
```

```



```

	precision	recall	f1-score	support
Without_Diabetes	0.81	0.87	0.84	99
With_Diabetes	0.73	0.64	0.68	55
accuracy			0.79	154
macro avg	0.77	0.75	0.76	154
weighted avg	0.78	0.79	0.78	154

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

