

```
In [25]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

```
In [27]: data = pd.read_csv("Downloads/diabetes (1).csv")
```

```
In [29]: data
```

```
Out[29]:
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction |
|-----|-------------|---------|---------------|---------------|---------|------|--------------------------|
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 763 | 10 | 101 | 76 | 48 | 180 | 32.9 | |
| 764 | 2 | 122 | 70 | 27 | 0 | 36.8 | |
| 765 | 5 | 121 | 72 | 23 | 112 | 26.2 | |
| 766 | 1 | 126 | 60 | 0 | 0 | 30.1 | |
| 767 | 1 | 93 | 70 | 31 | 0 | 30.4 | |

768 rows × 9 columns

```
In [31]: data.tail()
```

```
Out[31]:
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction |
|-----|-------------|---------|---------------|---------------|---------|------|--------------------------|
| 763 | 10 | 101 | 76 | 48 | 180 | 32.9 | |
| 764 | 2 | 122 | 70 | 27 | 0 | 36.8 | |
| 765 | 5 | 121 | 72 | 23 | 112 | 26.2 | |
| 766 | 1 | 126 | 60 | 0 | 0 | 30.1 | |
| 767 | 1 | 93 | 70 | 31 | 0 | 30.4 | |

```
In [33]: data.head()
```

```
Out[33]:
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction |
|---|-------------|---------|---------------|---------------|---------|------|--------------------------|
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | |

```
In [35]: data.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   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: 54.1 KB
```

```
In [37]: data.head(10)
```

```
Out[37]:
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction |
|---|-------------|---------|---------------|---------------|---------|------|--------------------------|
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | |
| 5 | 5 | 116 | 74 | 0 | 0 | 25.6 | |
| 6 | 3 | 78 | 50 | 32 | 88 | 31.0 | |
| 7 | 10 | 115 | 0 | 0 | 0 | 35.3 | |
| 8 | 2 | 197 | 70 | 45 | 543 | 30.5 | |
| 9 | 8 | 125 | 96 | 0 | 0 | 0.0 | |

```
In [39]: data.isnull().sum()
```

```
Out[39]:
```

| | |
|--------------------------|---|
| Pregnancies | 0 |
| Glucose | 0 |
| BloodPressure | 0 |
| SkinThickness | 0 |
| Insulin | 0 |
| BMI | 0 |
| DiabetesPedigreeFunction | 0 |
| Age | 0 |
| Outcome | 0 |

```
dtype: int64
```

```
In [41]: X = data.drop("Outcome", axis=1)
y = data["Outcome"]
```

```
In [43]: X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
In [45]: scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
In [47]: model = LogisticRegression()
model.fit(X_train, y_train)
```

```
Out[47]: LogisticRegression  
LogisticRegression()
```

```
In [49]: y_pred = model.predict(X_test)
```

```
In [51]: print("Accuracy:", accuracy_score(y_test, y_pred))  
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Accuracy: 0.7532467532467533

Classification Report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.81 | 0.80 | 0.81 | 99 |
| 1 | 0.65 | 0.67 | 0.66 | 55 |
| accuracy | | | 0.75 | 154 |
| macro avg | 0.73 | 0.74 | 0.73 | 154 |
| weighted avg | 0.76 | 0.75 | 0.75 | 154 |

```
In [ ]:
```