```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, recall_score, f1_score, confusion_matrix
import matplotlib.pyplot as plt
```

```
df = pd.read_csv('/content/diabetes2.csv')
df.sample(5)
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| **655** | 2 | 155 | 52 | 27 | 540 | 38.7 | 0.240 | 25 | 1 |
| **253** | 0 | 86 | 68 | 32 | 0 | 35.8 | 0.238 | 25 | 0 |
| **404** | 5 | 168 | 64 | 0 | 0 | 32.9 | 0.135 | 41 | 1 |
| **479** | 4 | 132 | 86 | 31 | 0 | 28.0 | 0.419 | 63 | 0 |
| **710** | 3 | 158 | 64 | 13 | 387 | 31.2 | 0.295 | 24 | 0 |

```
df.describe()
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| **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 |
| **max** | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | 2.420000 | 81.000000 | 1.000000 |

```
df.isnull().sum()
```

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

**dtype:** int64

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
print((df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']] == 0).sum())

df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']] = df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']].replace(0, pd.NaT)
df.fillna(df.mean(), inplace=True)
```

```
Glucose          0
BloodPressure    0
SkinThickness    0
Insulin          0
BMI              0
dtype: int64
```

```
X_train, X_test, y_train, y_test = train_test_split(df.drop(['Outcome'], axis=1),
                                                    df['Outcome'],
                                                    test_size=0.2,
                                                    random_state=2)
```

```
std = StandardScaler()
```

```
X_train = std.fit_transform(X_train)
X_test = std.transform(X_test)
```

```
model = LogisticRegression(max_iter=1000, class_weight='balanced')
```

```
model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression(class_weight='balanced', max_iter=1000)
```

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

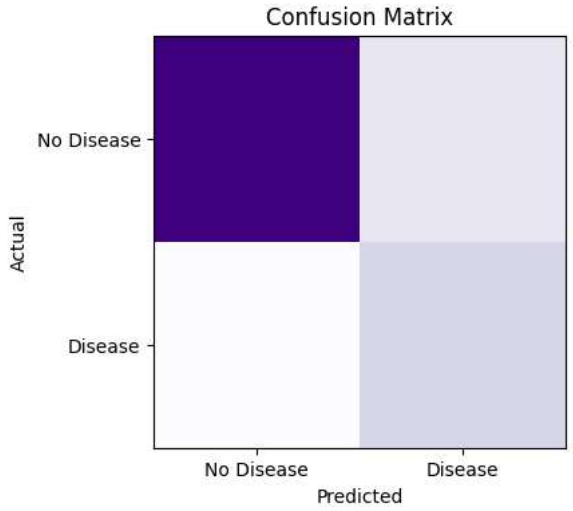
```
print('Accuracy: ', accuracy_score(y_test, y_pred))
print('Recall: ', recall_score(y_test, y_pred))
print('f1 score: ', f1_score(y_test, y_pred))
print('Confusion Matrix: ', confusion_matrix(y_test, y_pred))
```

```
Accuracy:  0.7532467532467533
Recall:  0.7111111111111111
f1 score:  0.6274509803921569
Confusion Matrix:  [[84 25]
 [13 32]]
```

```
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6,4))
plt.imshow(cm, cmap='Purples')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.xticks([0, 1], ['No Disease', 'Disease'])
plt.yticks([0, 1], ['No Disease', 'Disease'])
```

```
([<matplotlib.axis.YTick at 0x7ff1876fe030>,
  <matplotlib.axis.YTick at 0x7ff18c1fc890>],
 [Text(0, 0, 'No Disease'), Text(0, 1, 'Disease')])
```




```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.87 | 0.77 | 0.82 | 109 |
| 1 | 0.56 | 0.71 | 0.63 | 45 |
| accuracy | | | 0.75 | 154 |
| macro avg | 0.71 | 0.74 | 0.72 | 154 |
| weighted avg | 0.78 | 0.75 | 0.76 | 154 |