

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn import metrics
```

```
df=pd.read_csv('/content/diabetes.csv')
```

```
df.columns
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

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

```
Pregnancies      0
Glucose           0
BloodPressure     0
SkinThickness     0
Insulin           0
BMI               0
DiabetesPedigreeFunction  0
Age               0
Outcome           0
dtype: int64
```

ndarray: y_pred

ndarray with shape (154,)

```
X = df.drop('Outcome',axis = 1)
y = df['Outcome']
```

B

```
from sklearn.preprocessing import scale
X = scale(X)
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=0)
```

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```

```
print("Confusion matrix: ")
cs = metrics.confusion_matrix(y_test,y_pred)
print(cs)
```

Confusion matrix:

```
[[91 16]
 [18 29]]
```

```
print("Accuracy ",metrics.accuracy_score(y_test,y_pred))
```

Accuracy 0.7792207792207793

ndarray: y_pred

ndarray with shape (154,)

```
total_misclassified = cs[0,1] + cs[1,0]
print(total_misclassified)
total_examples = cs[0,0]+cs[0,1]+cs[1,0]+cs[1,1]
print(total_examples)
print("Error rate",total_misclassified/total_examples)
print("Error rate ",1-metrics.accuracy_score(y_test,y_pred))
```

34

154

Error rate 0.22077922077922077

Error rate 0.22077922077922074

```
print("Precision score".metrics.precision_score(y_test,y_pred))
```

```
➦ Precision score 0.6444444444444445
```

```
print("Recall score ",metrics.recall_score(y_test,y_pred))
```

```
Recall score 0.6170212765957447
```

```
print("Classification report ",metrics.classification_report(y_test,y_pred))
```

| Classification report | | precision | recall | f1-score | support |
|-----------------------|------|-----------|--------|----------|---------|
| 0 | 0.83 | 0.85 | 0.84 | 107 | |
| 1 | 0.64 | 0.62 | 0.63 | 47 | |
| accuracy | | | 0.78 | 154 | |
| macro avg | | 0.74 | 0.73 | 0.74 | 154 |
| weighted avg | | 0.78 | 0.78 | 0.78 | 154 |

ndarray: y_pred

ndarray with shape (154,)

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 11:27 AM



```
ndarray: y_pred  
ndarray with shape (154,)
```