

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn import metrics
```

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

```
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Pedigree	Age	Outcome
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

```
df.columns
```

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

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

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

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

```
from sklearn.preprocessing import scale
X = scale(X)
# split into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)
```

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

```
cs = metrics.confusion_matrix(y_test,y_pred)
print(cs)
```

```
[[128  27]
 [ 36  40]]
```

```
print("Accuracy ",metrics.accuracy_score(y_test,y_pred))
print("Error rate ",1-metrics.accuracy_score(y_test,y_pred))
print("Precision score",metrics.precision_score(y_test,y_pred))
print("Recall score ",metrics.recall_score(y_test,y_pred))
print("Classification report ",metrics.classification_report(y_test,y_pred))
```

```
Accuracy 0.7272727272727273
Error rate 0.2727272727272727
Precision score 0.5970149253731343
Recall score 0.5263157894736842
Classification report          precision    recall  f1-score   support
```

0	0.78	0.83	0.80	155
1	0.60	0.53	0.56	76
accuracy			0.73	231
macro avg	0.69	0.68	0.68	231
weighted avg	0.72	0.73	0.72	231