K NEAREST NEIGHBOUR 3 ML

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, f1_score, recall_score, precision_score, accuracy_score

df = pd.read_csv("diabetes.csv")
df.head()

df.shape

df.describe()

# replace zeros
zero_not_accepted = ["Glucose", "BloodPressure", "SkinThickness", "BMI", "Insulin"]
for column in zero_not_accepted:
    df[column] = df[column].replace(0, np.NaN)
    mean = int(df[column].mean(skipna=True))
    df[column] = df[column].replace(np.NaN, mean)

print(df["Glucose"])

# split dataset
X = df.iloc[:, 0:8]
y = df.iloc[:, 8]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, test_size=0.2)
# feature Scaling
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)

X_test = sc_X.transform(X_test)
knn = KNeighborsClassifier(n_neighbors=11)
print(knn.fit(X_train, y_train))

y_pred = knn.predict(X_test)
# Evaluate The Model
cf_matrix = confusion_matrix(y_test, y_pred)
ax = sns.heatmap(cf_matrix, annot=True, cmap='Blues')

ax.set_title('Seaborn Confusion Matrix with labels\n\n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ');

## Display the visualization of the Confusion Matrix.
plt.show()

tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
print(tn, fp, fn, tp)

# The accuracy rate is equal to (tn+tp)/(tn+tp+fn+fp)
print(accuracy_score(y_test, y_pred))

# The precision is the ratio of tp/(tp + fp)
print(precision_score(y_test, y_pred))
```

```
##The recall is the ratio of tp/(tp + fn)
print(recall_score(y_test, y_pred))

# error rate=1-accuracy which is lies bertween 0 and 1
error_rate = 1 - accuracy_score(y_test, y_pred)
error_rate
```