```
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


# In[3]:


df=pd.read_csv("Dataset/diabetes.csv")


# In[4]:


df.head()


# In[5]:


df.shape


# In[6]:


df.describe()


# In[7]:


#replace zeros
zero_not_accepted=["Glucose","BloodPressure","SkinThickness","BMI","Insul
in"]
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)


# In[8]:


df["Glucose"]


# In[9]:


#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_si
ze=0.2)


# In[10]:


#feature Scaling
sc_X=StandardScaler()
X_train=sc_X.fit_transform(X_train)

X_test=sc_X.transform(X_test)


# In[11]:


knn=KNeighborsClassifier(n_neighbors=11)


# In[12]:


knn.fit(X_train,y_train)


# In[13]:


y_pred=knn.predict(X_test)


# In[14]:


#Evaluate The Model
cf_matrix=confusion_matrix(y_test,y_pred)


# In[15]:


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()


# In[16]:
```

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


# In[17]:


tn, fp, fn, tp


# In[18]:


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


# In[19]:


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


# In[20]:


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


# In[22]:


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


# In[4]:


import requests
print(requests.get("D:\coding\c++\.vscode\demo.py").text)


# In[ ]:
```