**Implement k Nearest neighbours algorithm on diabetes .csv dataset**

**pip install numpy**

**pip install pandas**

**pip install matplotlib**

**pip install seaborn**

**pip install sklearn**

**diabetes.csv dataset required**

**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\diabetes.csv')**

**print(df.head() )**

**print(df.shape)**

**print(df.describe() )**

**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)**

**df["Glucose"]**

**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)**

**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 ')**

**plt.show()**