**DATA MINING PROJECT**

TITLE: **DIABETES PREDICTION**

TEAM MEMBERS:

1. SARANG RAJ(RA1911003010101)
2. S AJAY(RA1911003010102)
3. TARUUN P(RA1911003010103)
4. S KARTHIKEYAN(RA1911003010120)
5. SAIPAVAN TIPPANU(RA1911003010121)
6. SRIRAM BALASUBRAMANIAN(RA1911003010124)
7. VAGEESH DIXIT(RA1911003010133)
8. K PAVITHRAN(RA1911003010135)
9. BENAK RAJ(RA1911003010136)
10. RAJ SRIVASTAVA(RA1911003010139)

DATASET: DIABETES CSV

[https://github.com/legendslayer01/Diabetes-supervised/blob/main/diabetes.csv](about:blank)

Ipynb file:

[https://github.com/legendslayer01/Diabetes-supervised/blob/main/diabetes.ipynb](about:blank)

CODE:

**import** pandas **as** pd

data**=**pd**.**read\_csv('diabetes.csv')

data**.**head()

data**.**shape

data**.**Outcome**.**unique()

data**.**describe()

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

plt**.**figure(figsize**=**(20,25))

plotnumber**=**1

**for** column **in** data:

**if** plotnumber**<=**9:

ax**=**plt**.**subplot(3,3,plotnumber)

sns**.**histplot(data[column])

plt**.**xlabel(column,fontsize**=**20)

plotnumber**+=**1

plt**.**show()

data**.**BMI**=**data**.**BMI**.**replace(0,data**.**BMI**.**mean())

data**.**Age**=**data**.**Age**.**replace(0,data**.**Age**.**mean())

data**.**Insulin**=**data**.**Insulin**.**replace(0,data**.**Insulin**.**mean())

data**.**Glucose**=**data**.**Glucose**.**replace(0,data**.**Glucose**.**mean())

data**.**BloodPressure**=**data**.**BloodPressure**.**replace(0,data**.**BloodPressure**.**mean())

data**.**SkinThickness**=**data**.**SkinThickness**.**replace(0,data**.**SkinThickness**.**mean())

X**=**data**.**drop(columns**=**['Outcome'])

y**=**data['Outcome']

**from** sklearn.preprocessing **import** StandardScaler

scaler**=**StandardScaler()

X\_Scaled**=**scaler**.**fit\_transform(X)

**from** sklearn.model\_selection **import** train\_test\_split

x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(X\_Scaled,y,test\_size**=**0.2,random\_state**=**100)

print(x\_train**.**shape)

print(x\_test**.**shape)

print(y\_train**.**shape)

print(y\_test**.**shape)

**from** sklearn.neighbors **import** KNeighborsClassifier

knn**=**KNeighborsClassifier()

knn**.**fit(x\_train,y\_train)

knn**.**score(x\_train,y\_train)

knn**.**score(x\_test,y\_test)

prediction**=**knn**.**predict(x\_test)

prediction

**from** sklearn.metrics **import** accuracy\_score,confusion\_matrix

print(accuracy\_score(prediction,y\_test))

print(confusion\_matrix(prediction,y\_test))

str((80**+**27)**/**(80**+**26**+**21**+**27))**+**" is the Accuracy"

str((80**/**(80**+**26)))**+**" is the precision"

str((80**/**(80**+**21)))**+**" is the recall"

str((2**\***80**/**(2**\***80**+**26**+**21)))**+**" is the f1 score"

**from** sklearn.model\_selection **import** GridSearchCV

param\_grid**=**{'algorithm':['ball\_tree','kd\_tree','brute'],

'leaf\_size':[18,20,22,23,25,26,28,30,34],

'n\_neighbors':[3,5,7,9,11,13,15,17,19]}

gridsearch**=**GridSearchCV(knn,param\_grid)

gridsearch**.**fit(x\_train,y\_train)

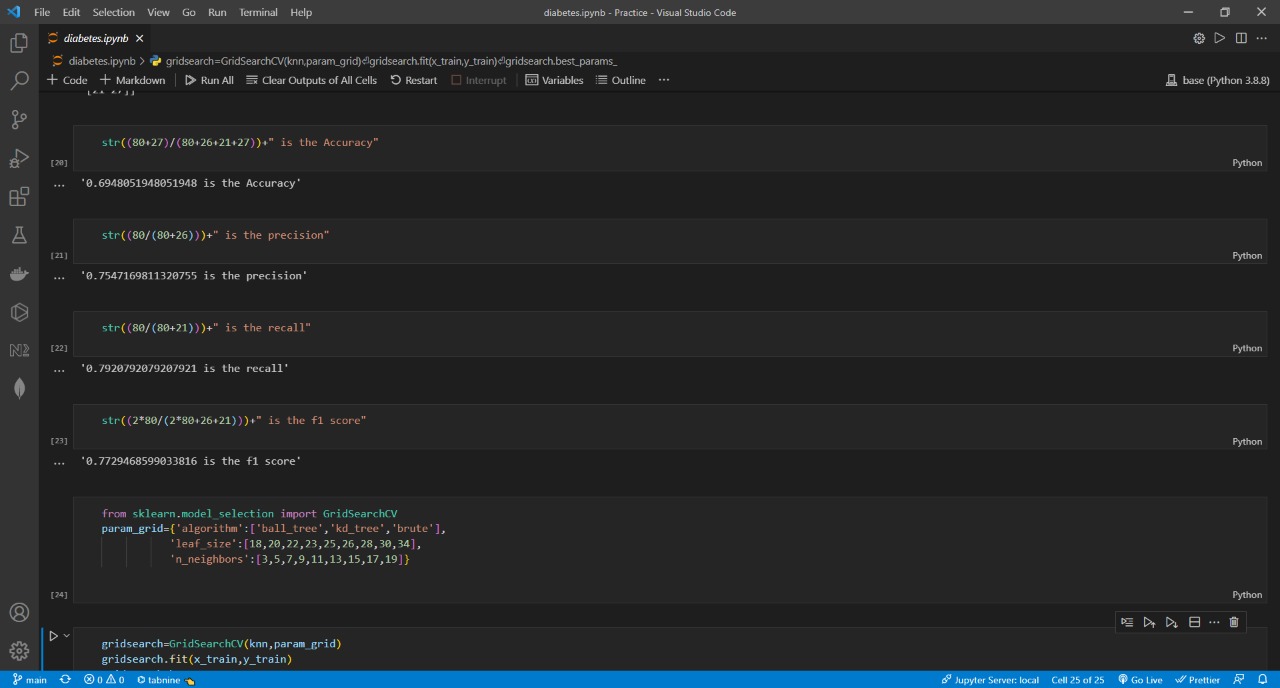
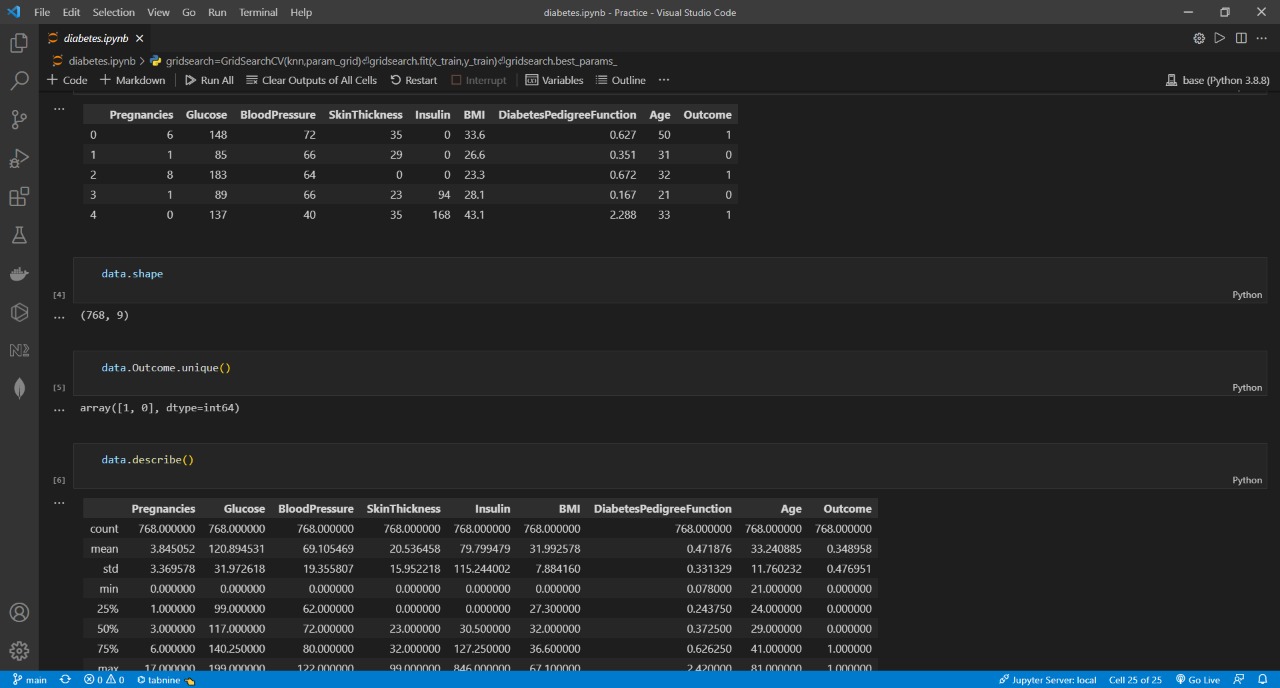
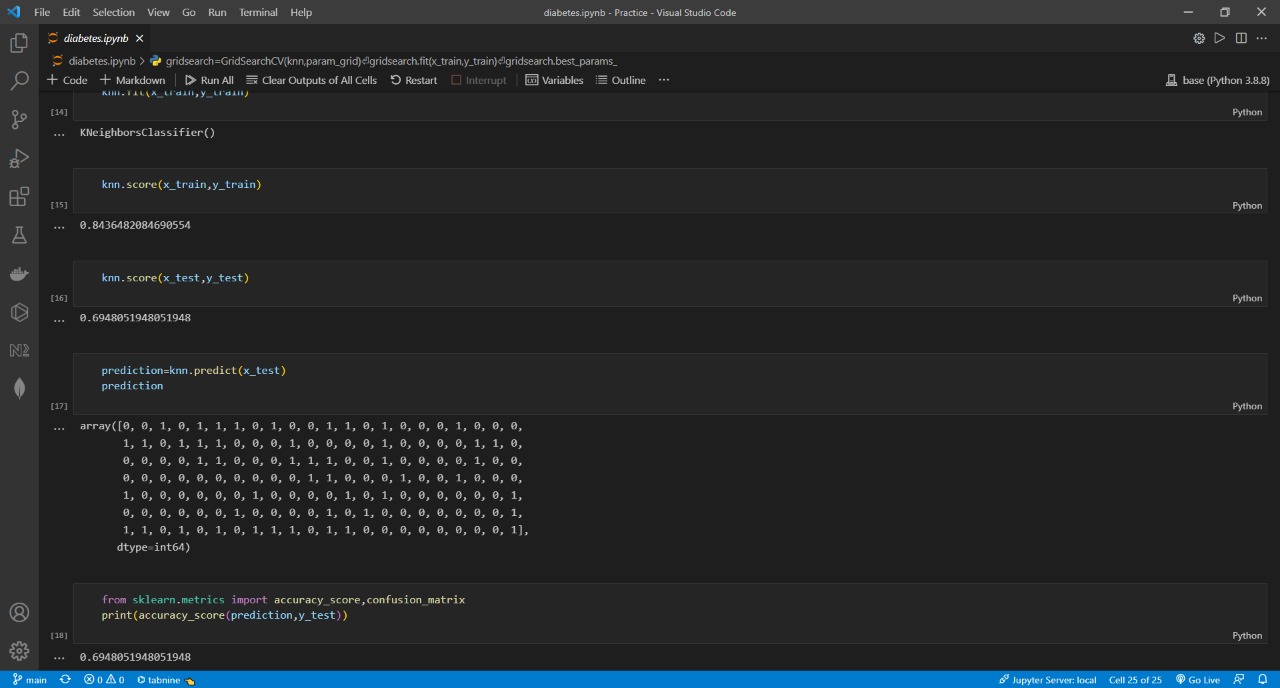
gridsearch**.**best\_params\_

SCREENSHOT:

Chart, histogram

Description automatically generated

**A picture containing text, screenshot, electronics, display

Description automatically generated**