**30thjan ’21 DECISION TREE CLASSIFICATION MODEL**

**NAME:**SARIKA MOHANRAJ

**AIM:**

To build the classification for whether a bank note is authentic or fake depending upon the four different attributes of the image of the note. The attributes are Variance of wavelet transformed image, curtosis of the image, entropy, and skewness of the image.

**VARIABLES USED :**

Variance - variance of Wavelet transformed image

Skewness - Skewness of Wavelet transformed image

Curtosis - Curtosis of Wavelet transformed image

Entropy - Entropy of image

Class – integer(1 or 0) ie authentic or fake

**MODEL:**

**IMPORTING THE LIBRARIES:**

import numpy as np

import pandas as pd

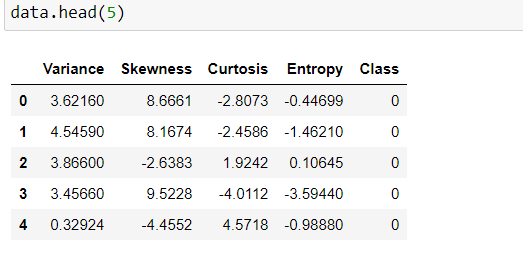
import matplotlib.pyplot as plt

import seaborn as sns

**IMPORTING DATASET:**

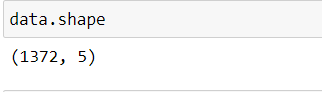
data=pd.read\_csv("D://datasets//bill\_authentication.csv")

**VISUALIZATION OF DATASET:**

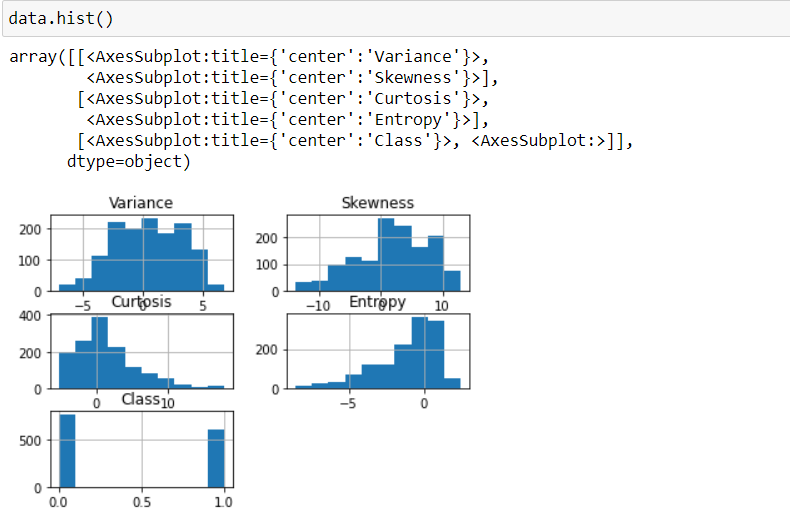


**DATA ANALYSIS:**

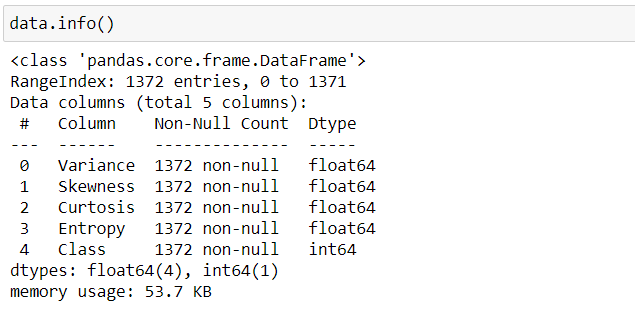
**SHAPE OF THE DATASET**



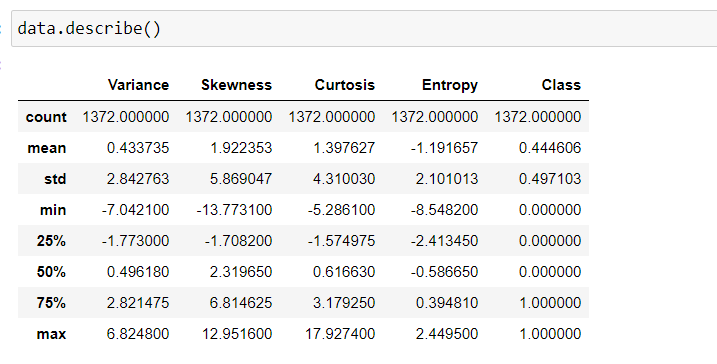
**HISTPLOT OF THE DATASET:**



**INFORMATION ABOUT THE DATASET**



**STATISTICAL DSECRIPTION ABOUT THE DATASET:**

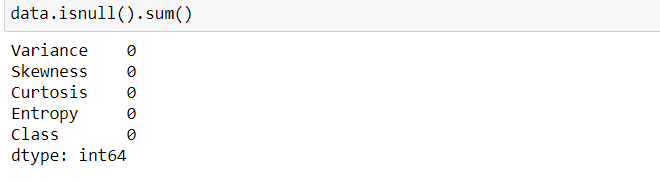


**COUNTPLOT ON NO.OF AUTHENTIC AND FAKE NOTES**

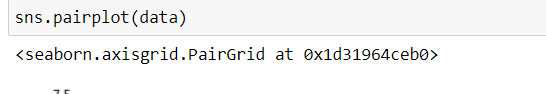


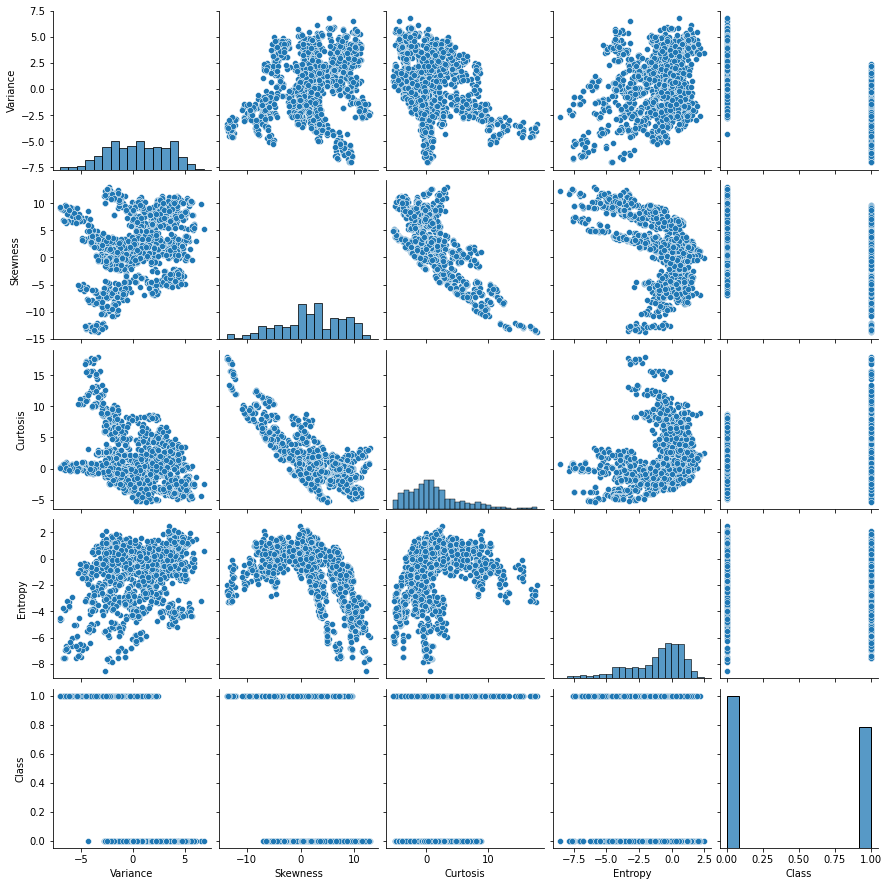
From the above plot we can interpret dataset contains more fake notes

**CHECKING FOR THE MISSING VALUES:**



**PAIRPLOT:**



****

Pairplot is a way to visualize relationships between each variable in matrix form

**PREPARING THE DATA FOR MODEL:**

x=data.drop(['Class'],axis=1)

y=data['Class']

**LIBRARIES NEEDED FOR DECISION TREE CLASSIFIER:**

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

from sklearn.tree import DecisionTreeClassifier

from sklearn import metrics

from sklearn.metrics import classification\_report, confusion\_matrix

from sklearn import tree

**TRAIN-TEST SPLIT:**

Here I have taken training data set size as 75 % and remaining 25% for testing the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.25)

**DECISION TREE CLASSIFIER MODEL:**

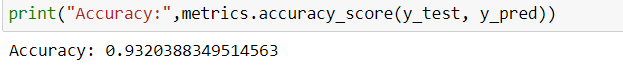
classifier = DecisionTreeClassifier(criterion='gini',max\_depth=3, random\_state=40)

classifier.fit(X\_train, y\_train)

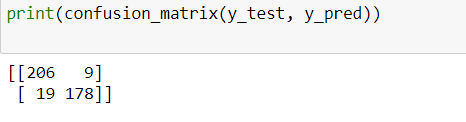
y\_pred = classifier.predict(X\_test)

**CLASSIFICATION METRICS**

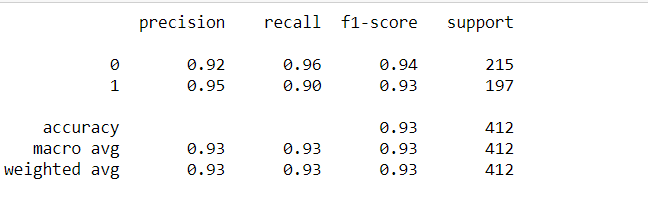
print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))



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



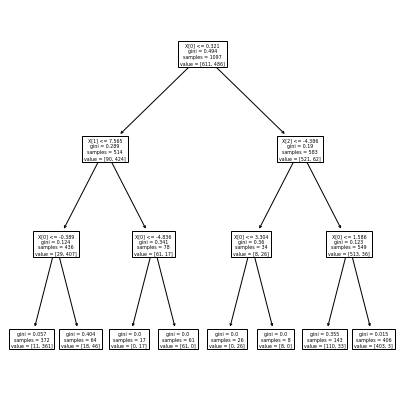
print(classification\_report(y\_test, y\_pred))



**TREE VISUALIZATION:**

plt.figure(figsize=(7,7))

tree.plot\_tree(classifier.fit(X\_train,y\_train))



**PREDICTIONS:**

input1=[[9.23432,6.32433,-8.4452,-3.53653]]

output=classifier.predict(input1)

print(output)

 🡪 fake

input2=[[-5.4352,3.54647,3.34563,-8.3241]]

output=classifier.predict(input2)

print(output)

🡪authentic

input3=[[1,1,-1,-1]]

output=classifier.predict(input3)

print(output)

🡪fake

**CONCLUSION:**

* Accuracy of model while using CART ALGORITHM as 93.2%
* Accuracy ( true values / total ) of confusion matrix came as 92%