CODE MODULE – 5

(PRACTICLE ASSESSMENT)

REGISTRATION NUMBER - ADIT22AP02849

NAME - ROSHNI VERMA

SET-A

QUESTION NUMBER-3

HARDWARE/SOFTWARE

1.COMPUTER

2.WINDOWS 10

3.EXCEL

4.MS WORD

STEP 1. Open the excel and put the first name,last name,salary in coloumn vise.

STEP 2. Put the formula Mean,median,and standard deviation .

STEP 3. SUM= SUM(C3:C11)

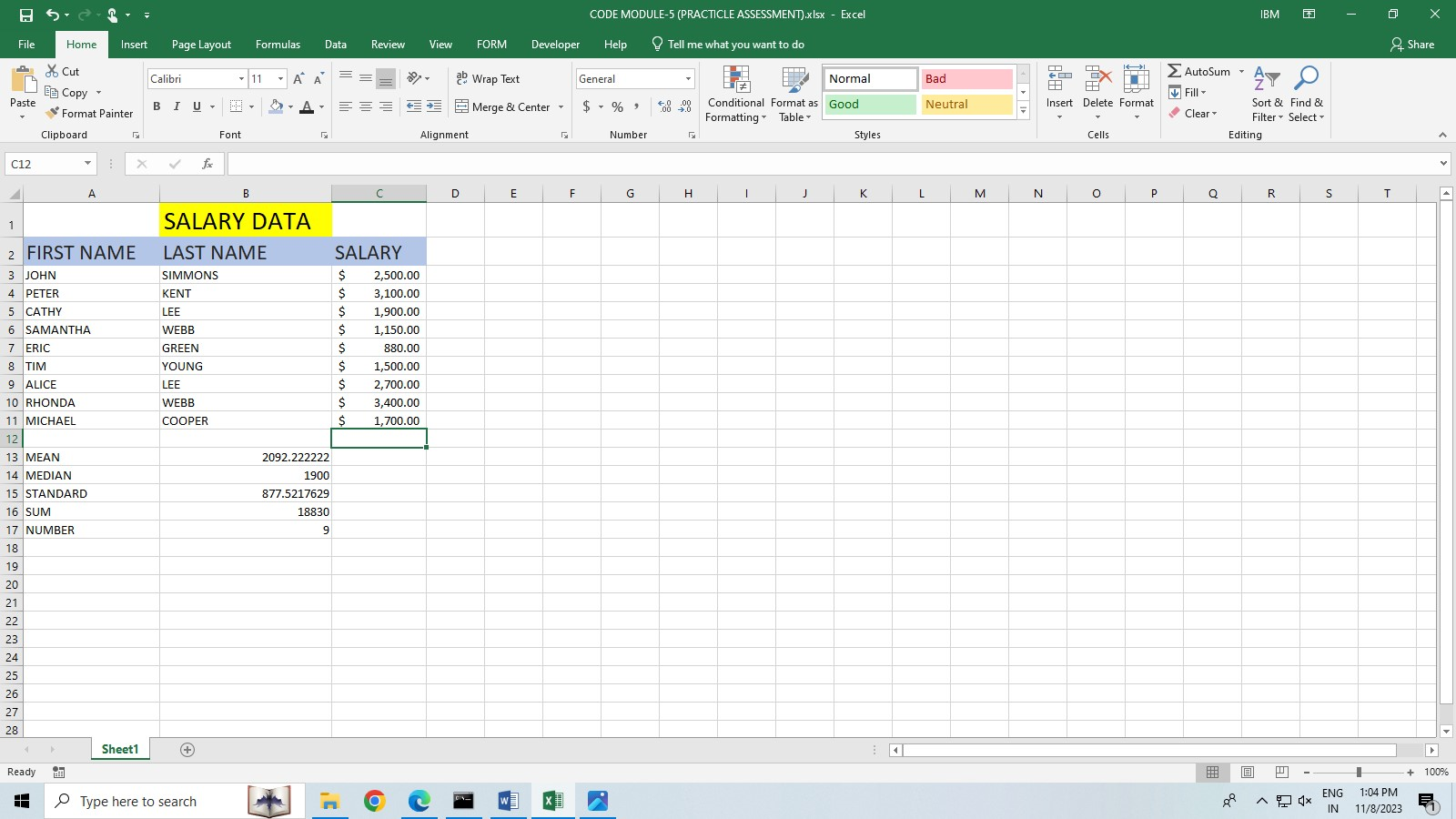
NUMBER= =COUNT(C3:C11)

MEAN =B16/B17

=SUM OF TOTAL NO/TOTAL FRQUENCY

STEP 4.MEDIAN= MEDIAN(C3:C11)

STEP 5.STANDARD DEVIATION =STDEV(C3:C11)



QUESTION -1.

HARDWARE/SOFTWARE

1.COMPUTER

2.WINDOWS 10

3.JUPYTER NOTEBOOK

4.MS WORD

5.INTERNET CONNECTION

STEP 1. Open the cmd and install jupyter notebook, pandas,matplotlib,sklearn.

STEP 2. IRIS.csv file download from Kaggle. And upload jupyter notebook.

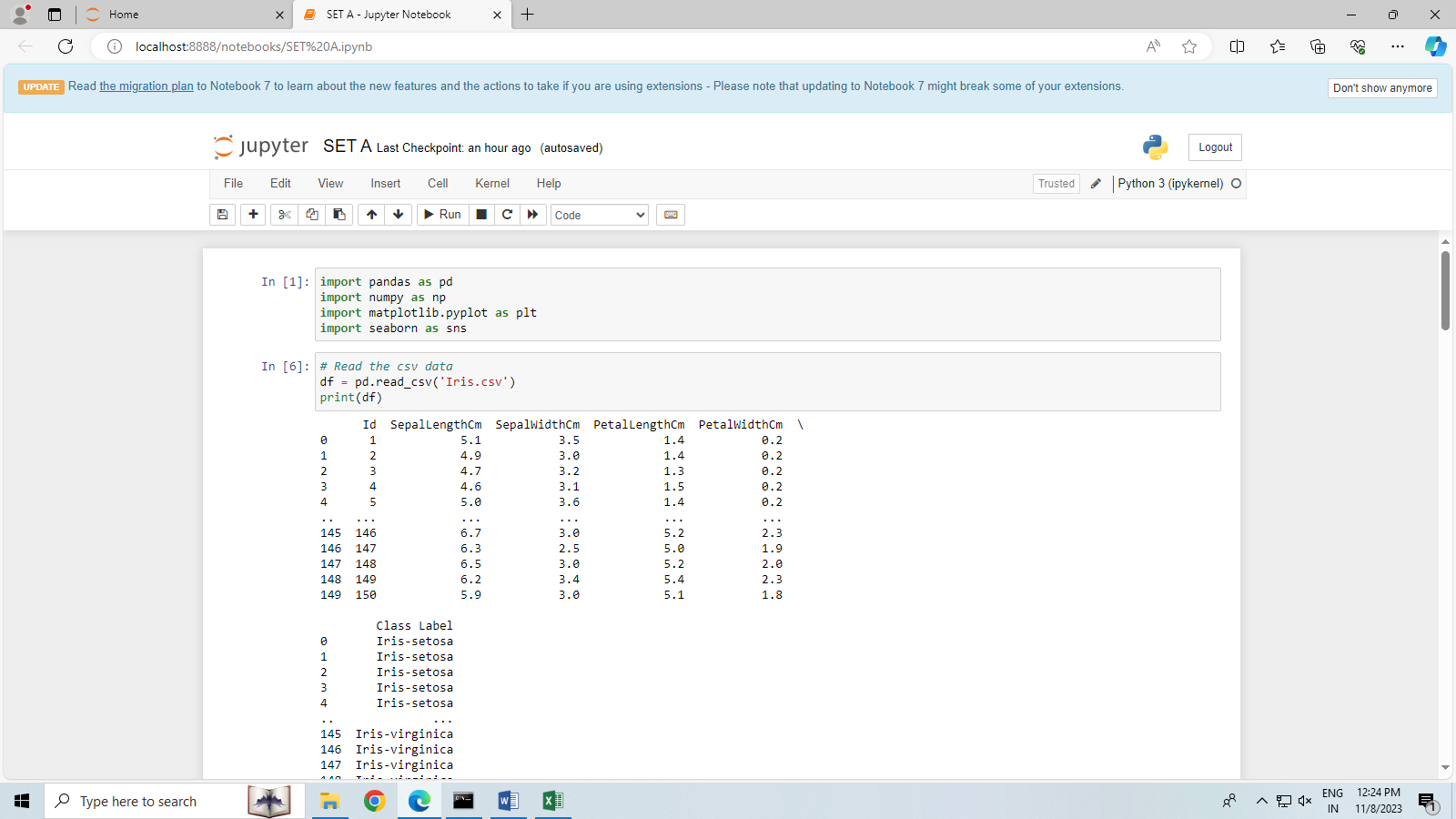
Put the code and run

STEP 3. import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

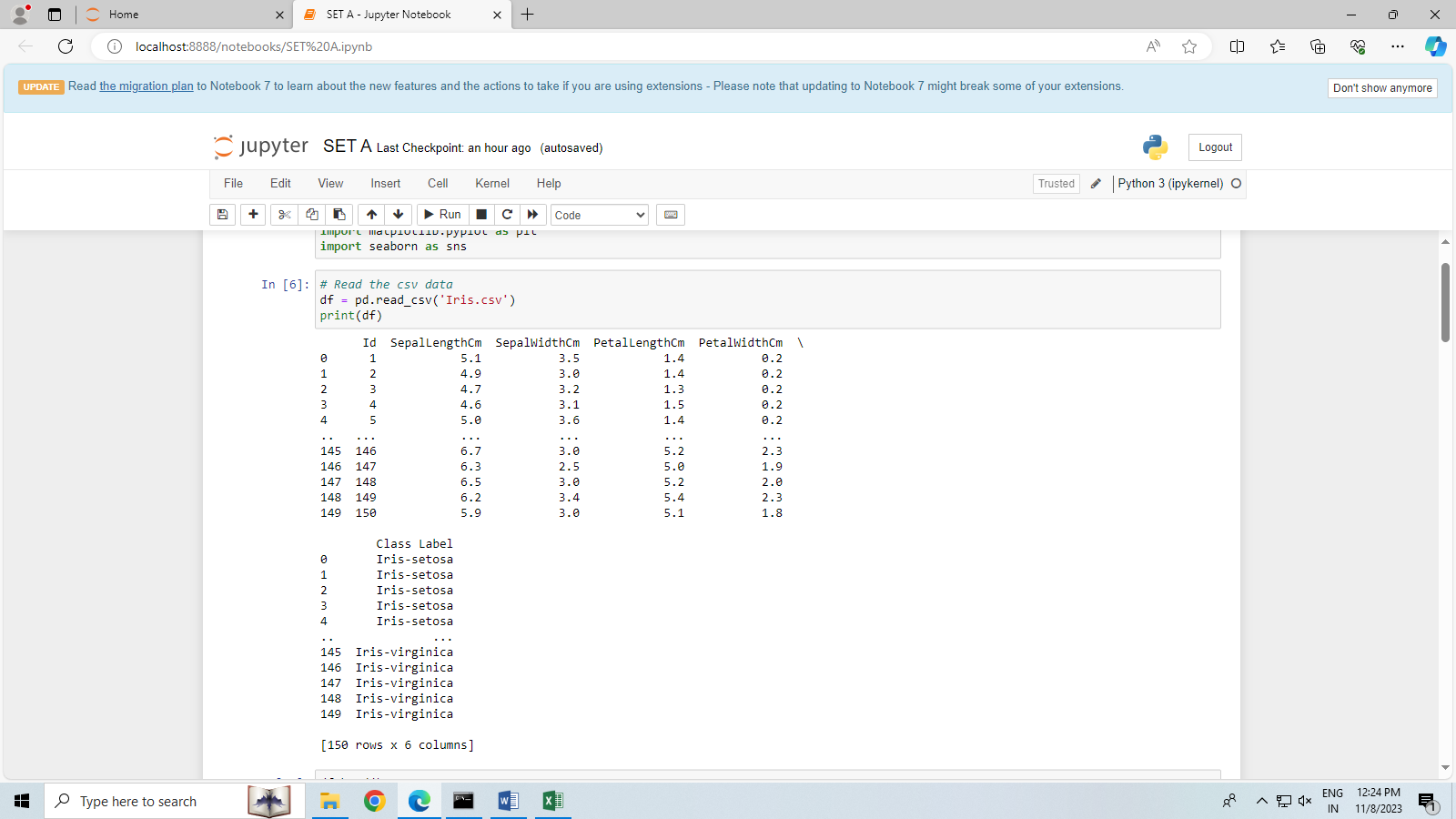
import seaborn as sns



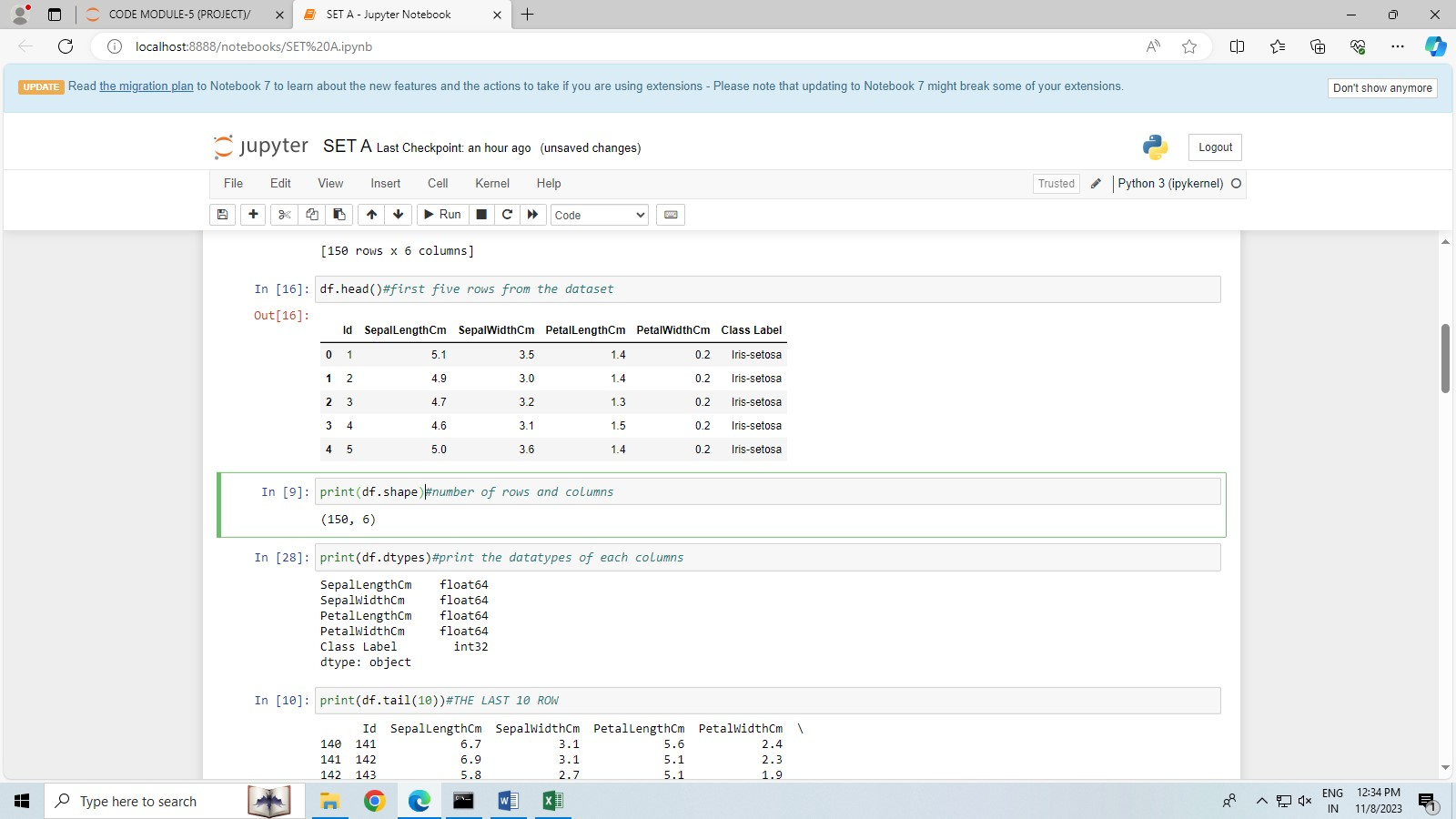
STEP 4. # Read the csv data

df = pd.read\_csv('Iris.csv')

print(df)

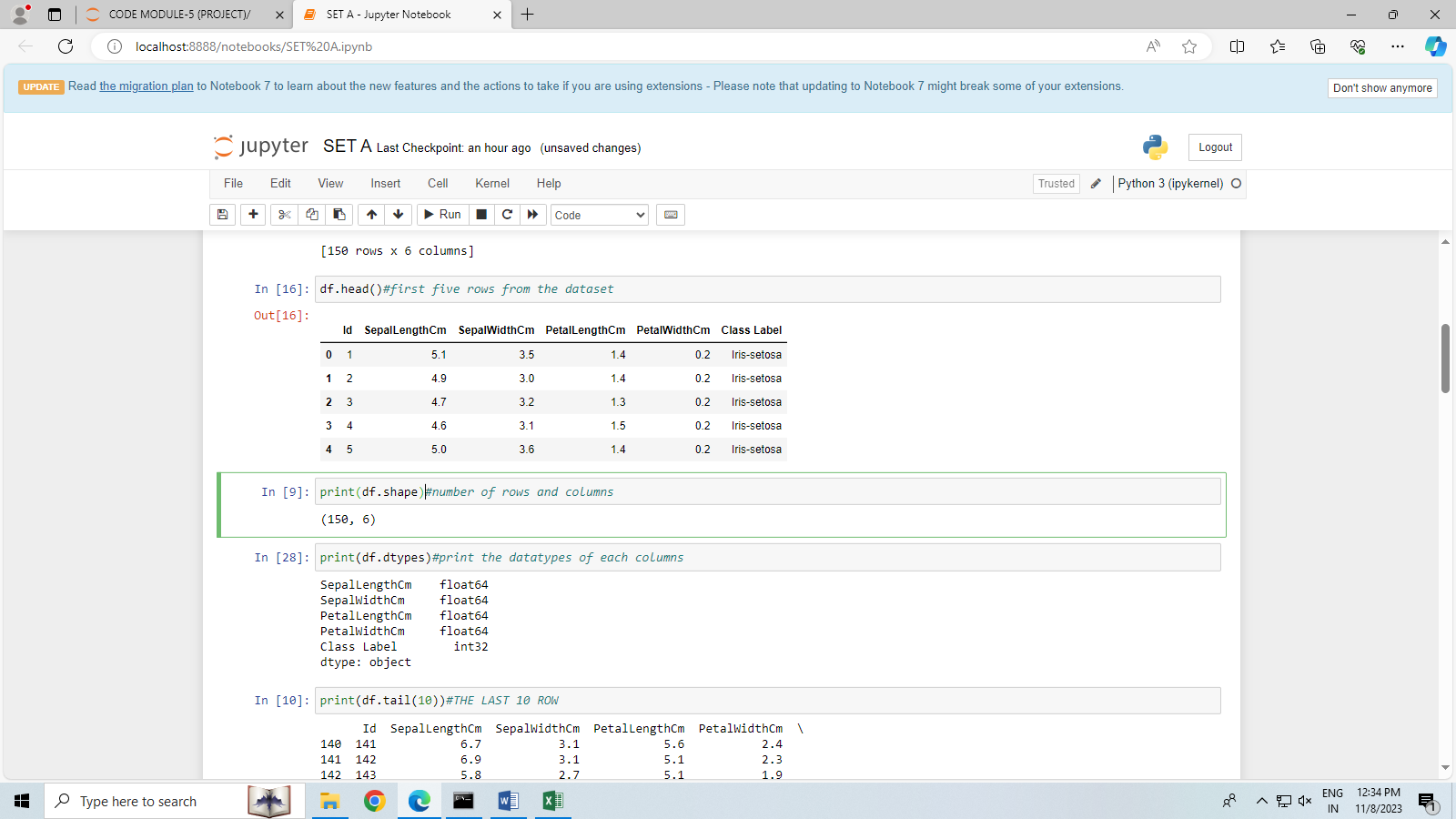


STEP 5. df.head()#first five rows from the dataset

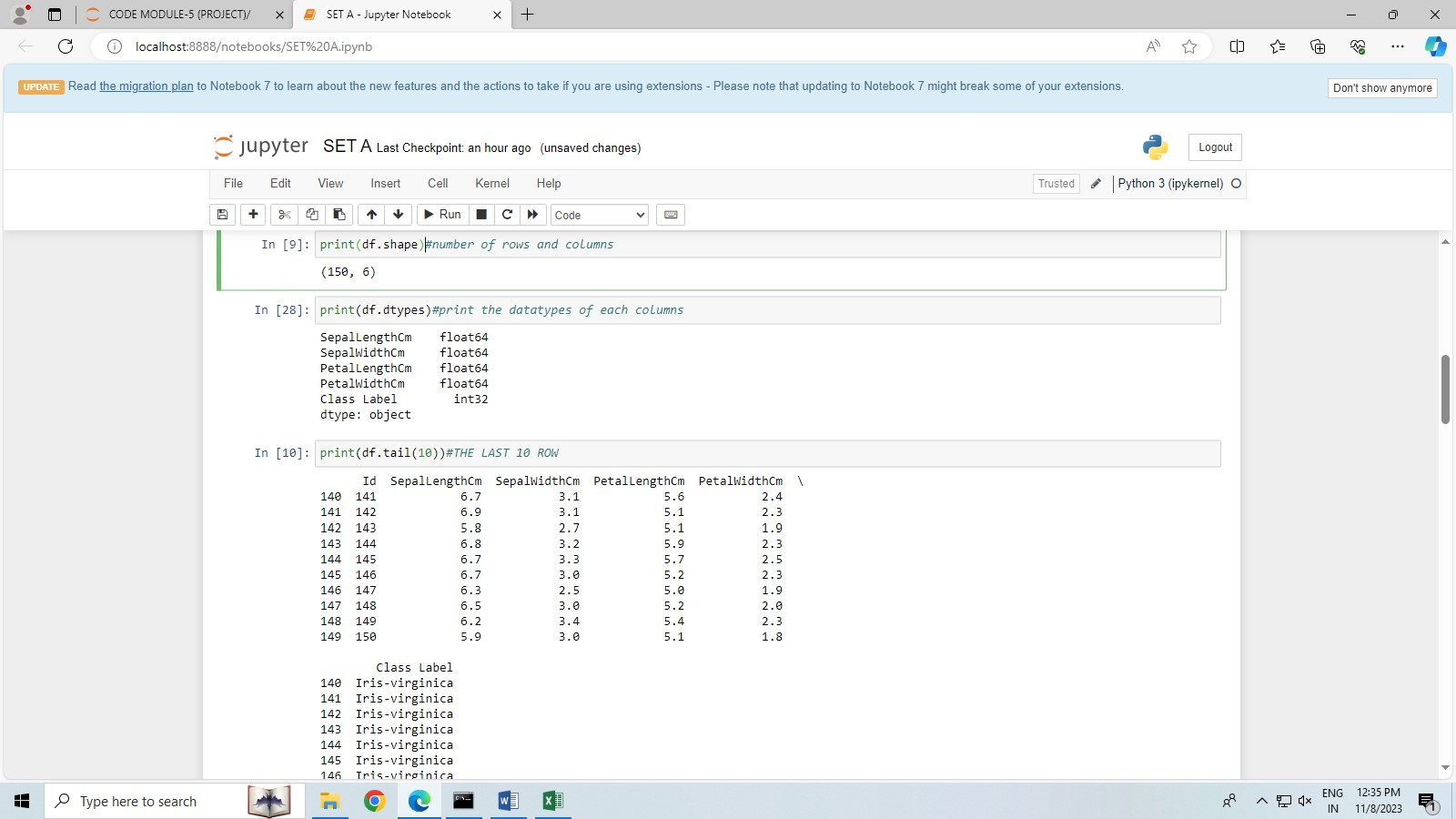


STEP 6. #number of rows and columns.

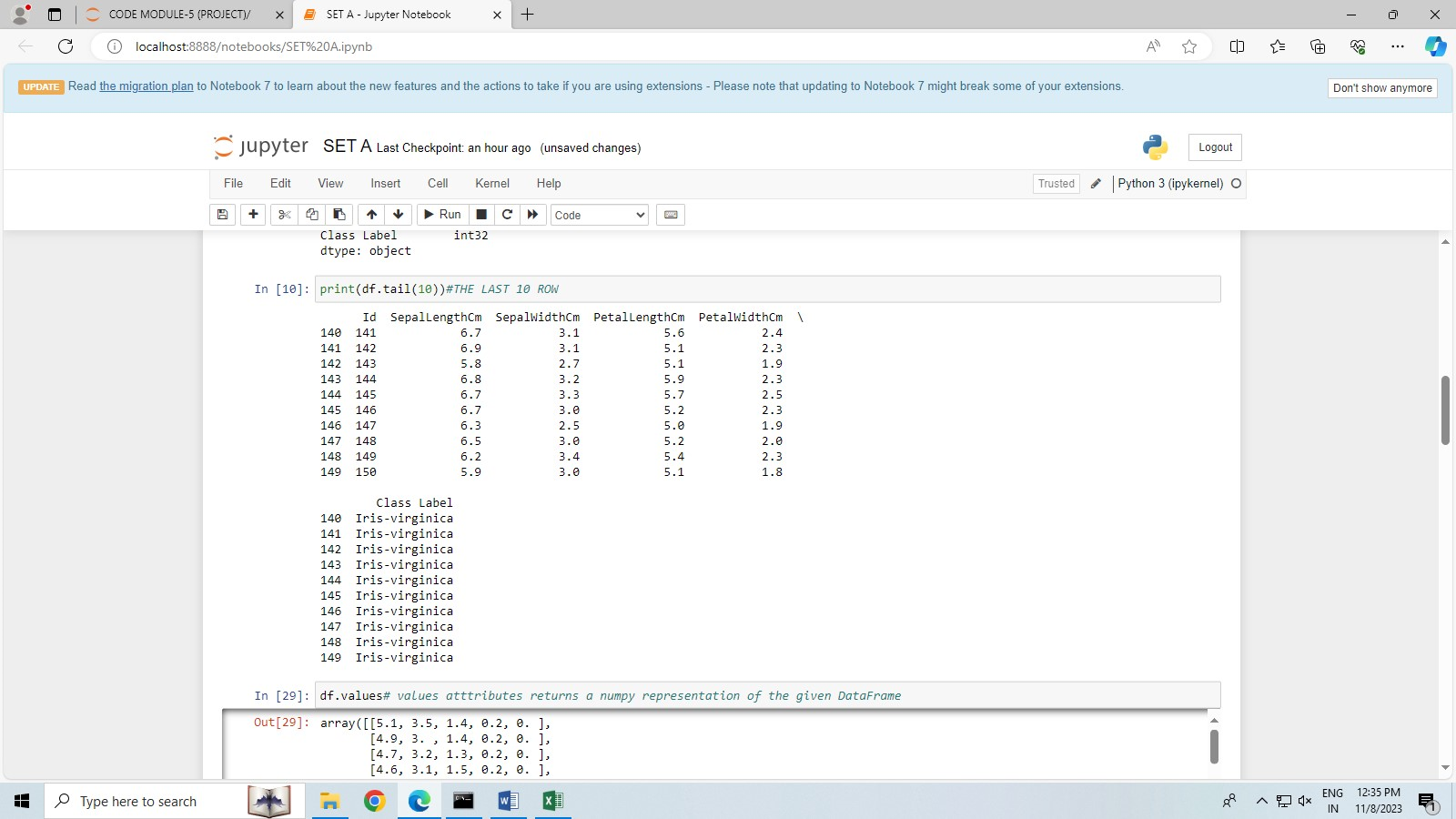
print(df.shape)



STEP 7. print(df.dtypes)#print the datatypes of each columns



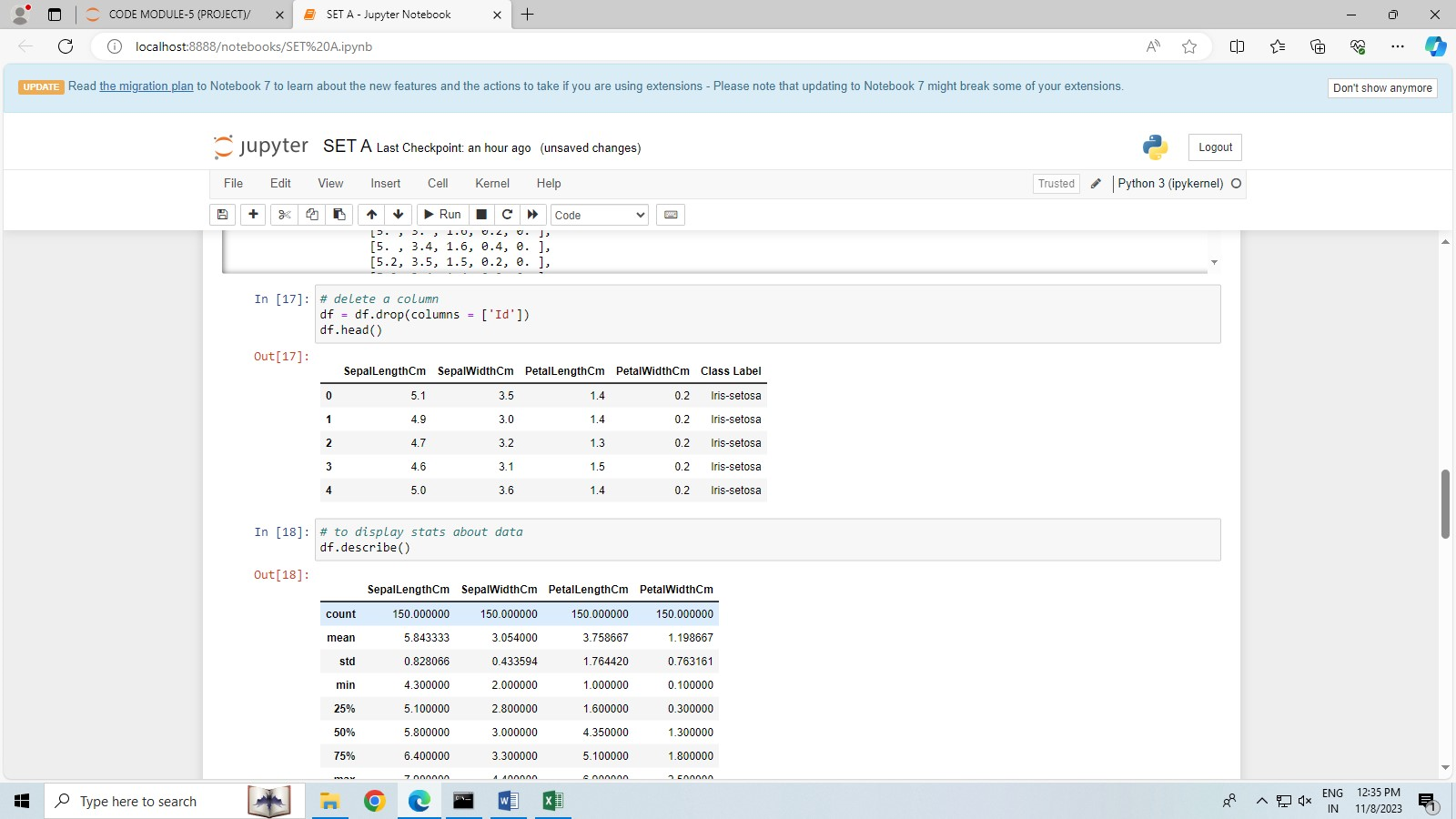
STEP 8. print(df.tail(10))# Get the last 10 row



STEP 9. df.values # values atttributes returns a numpy representation of the given DataFrame

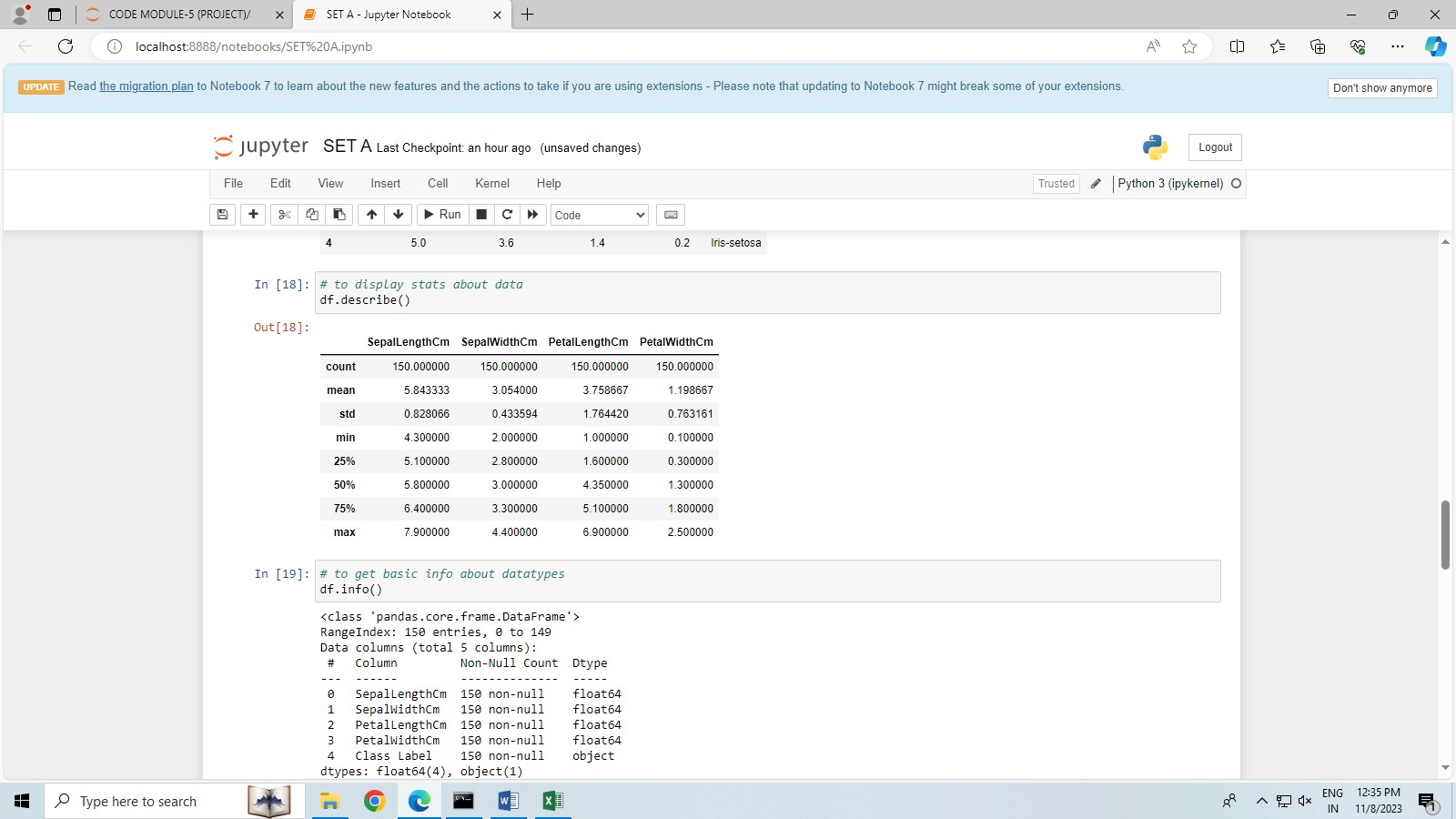


STEP 10. # delete a column

df = df.drop(columns = ['Id'])

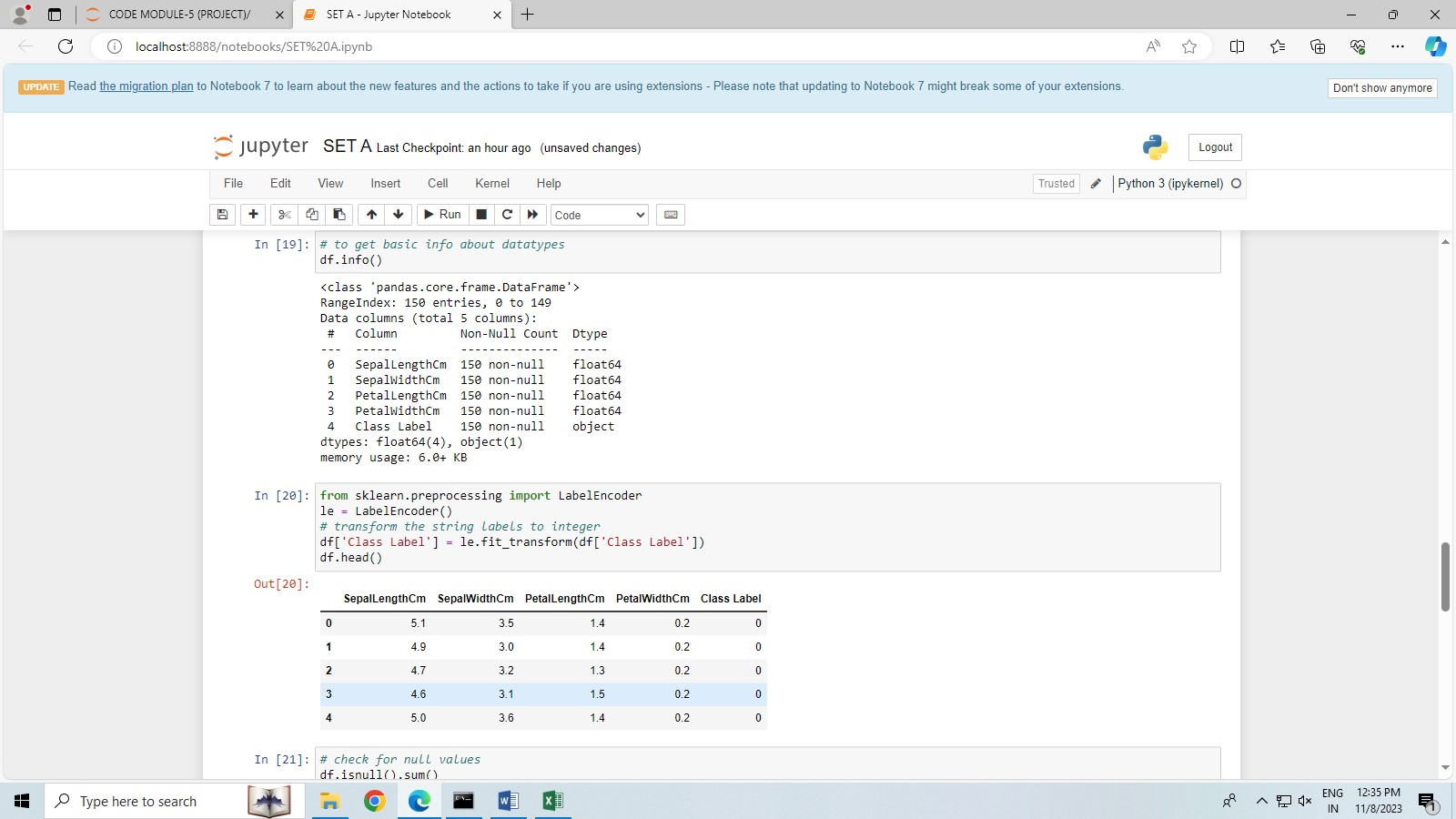
STEP 11. # to display stats about data

df.describe()



STEP 12. # to get basic info about datatypes

df.info()



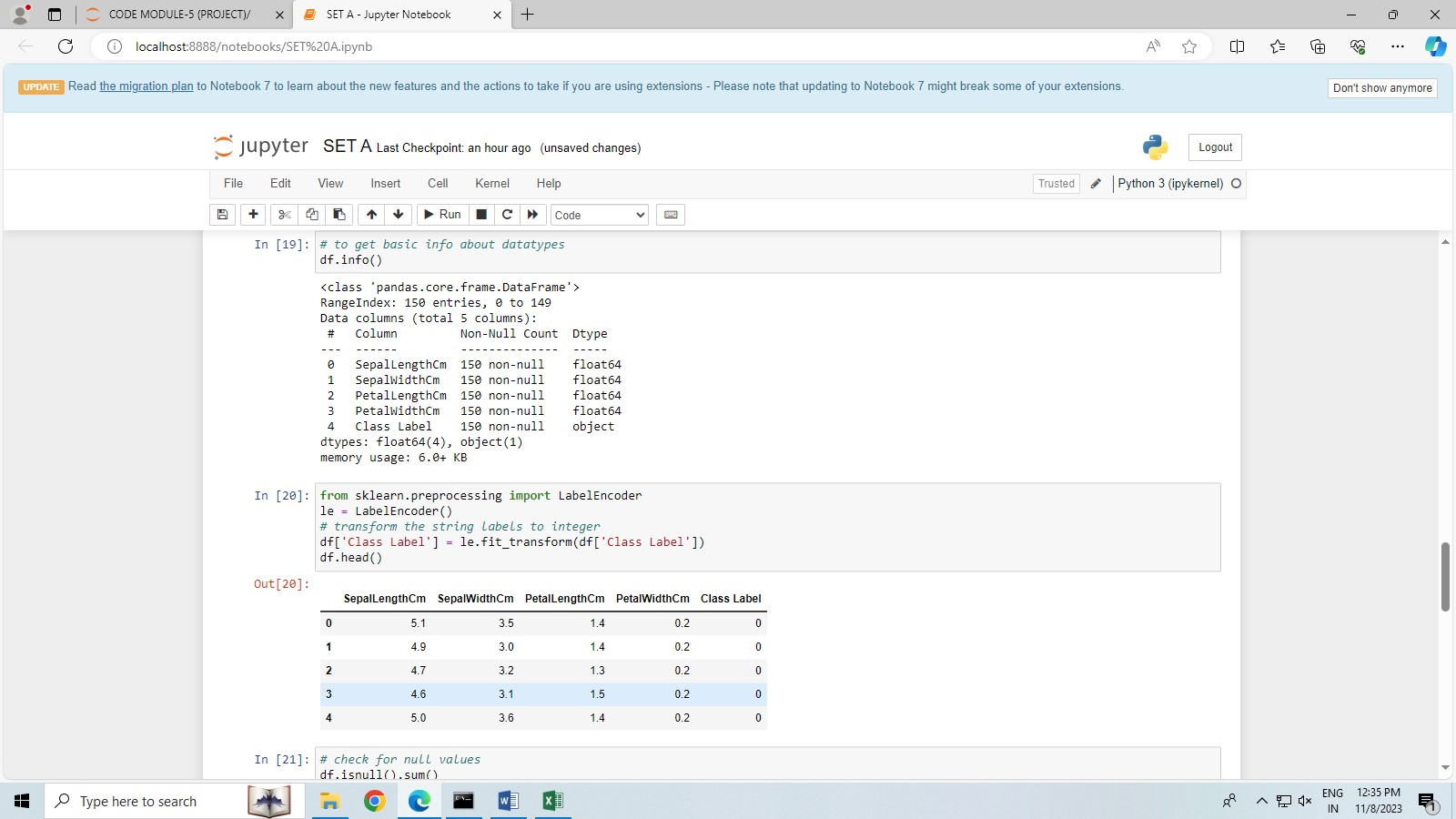
STEP 13. from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

# transform the string labels to integer

df['Class Label'] = le.fit\_transform(df['Class Label'])

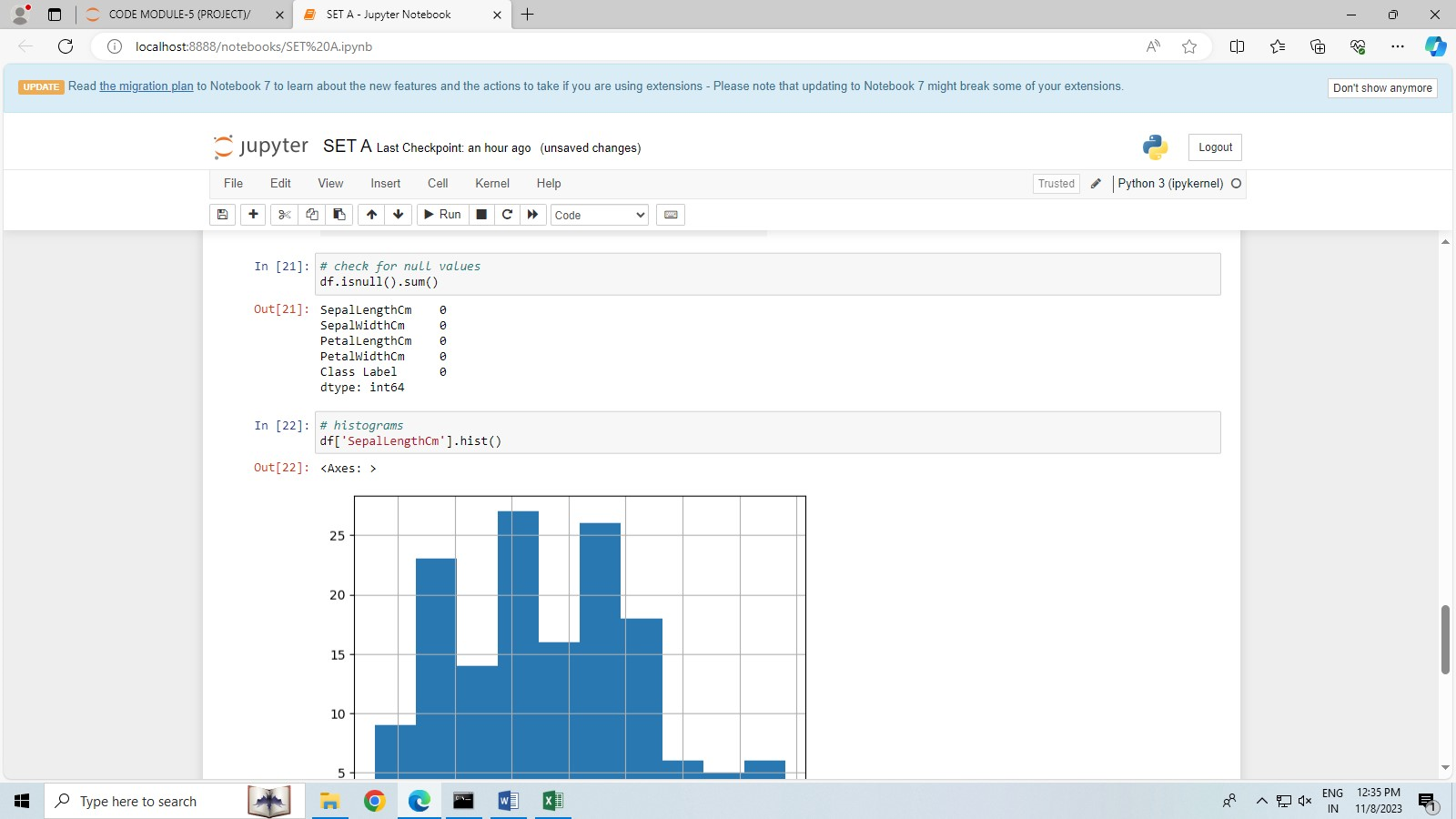
df.head()



STEP 14.

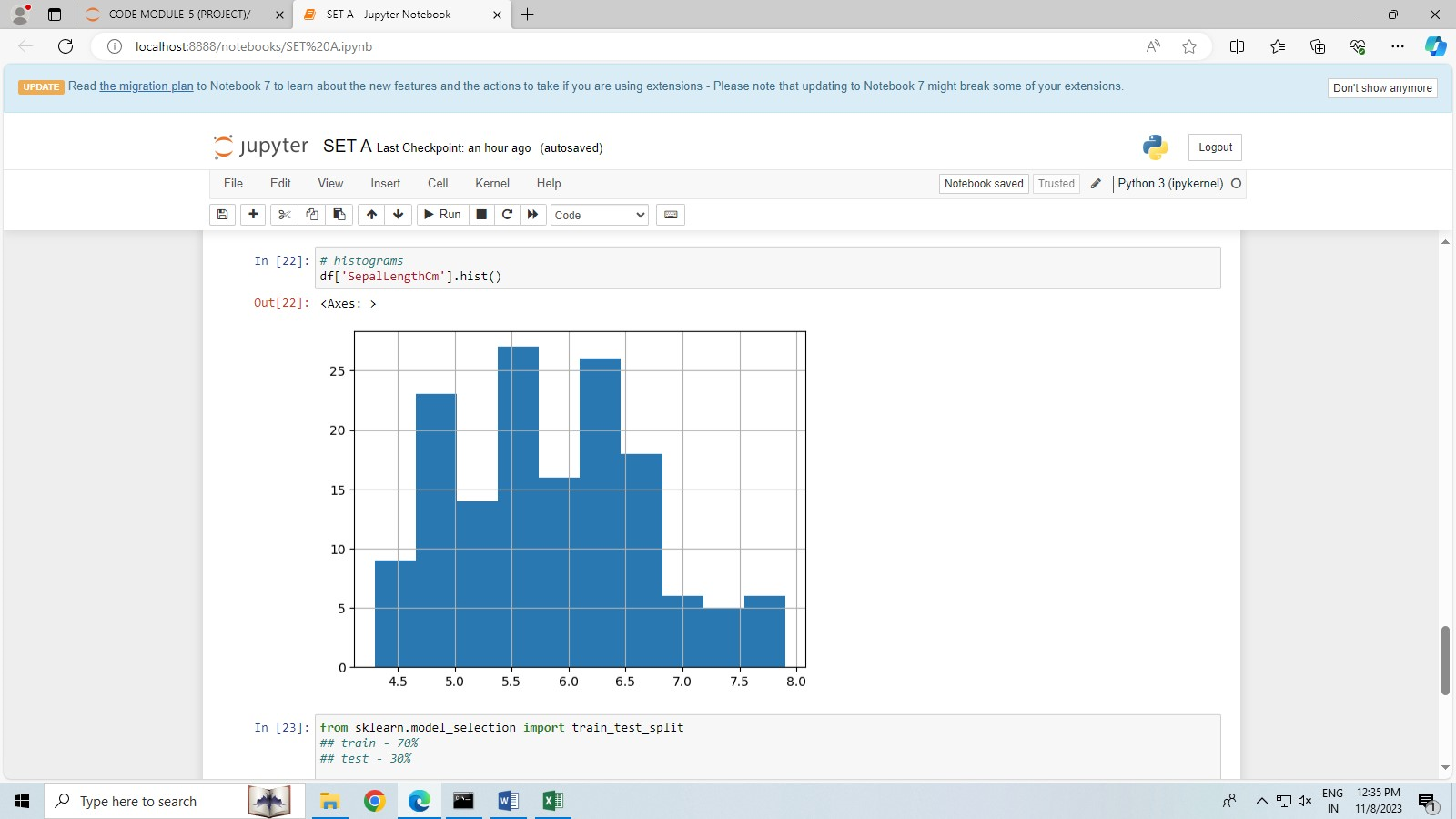
# check for null values

df.isnull().sum()



STEP 15. # histograms

df['SepalLengthCm'].hist()



STEP 16. from sklearn.model\_selection import train\_test\_split

## train - 70%

## test - 30%

# input data

X = df.drop(columns=['Class Label'])

# output data

Y = df['Class Label']

# split the data for train and test

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.30)

STEP 17 # logistic regression

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression()

STEP 18. # print metric to get performance

print("Accuracy: ",model.score(x\_test, y\_test) \* 100)

