**Experiment-1**

**Student Name:** YANA SRIVASTAVA **UID: 20BCS2279**

**Branch: CSE Section/Group: 906/B**

**Semester: 5th Date of Performance: 18/08/2022**

# Subject Name: ML Lab Subject Code: 20CSP-317

**1. Aim/Overview of the practical:** Implement Exploratory Data Analysis on any data set.

1. **Task to be done:**

* **Load dataset.**
* **Create a deep copy of dataset so that your original dataset remain same.**
* **Analyse column of dataset & if necessary then rename it .**
* **Check datatype of each column & check for NAN also.**
* **If there is NAN value then check the count of such number & fill or drop it accordingly.**
* **To get more insights about dataset create more columns i.e. Feature Scaling.**
* **Show changes that you made so far.**

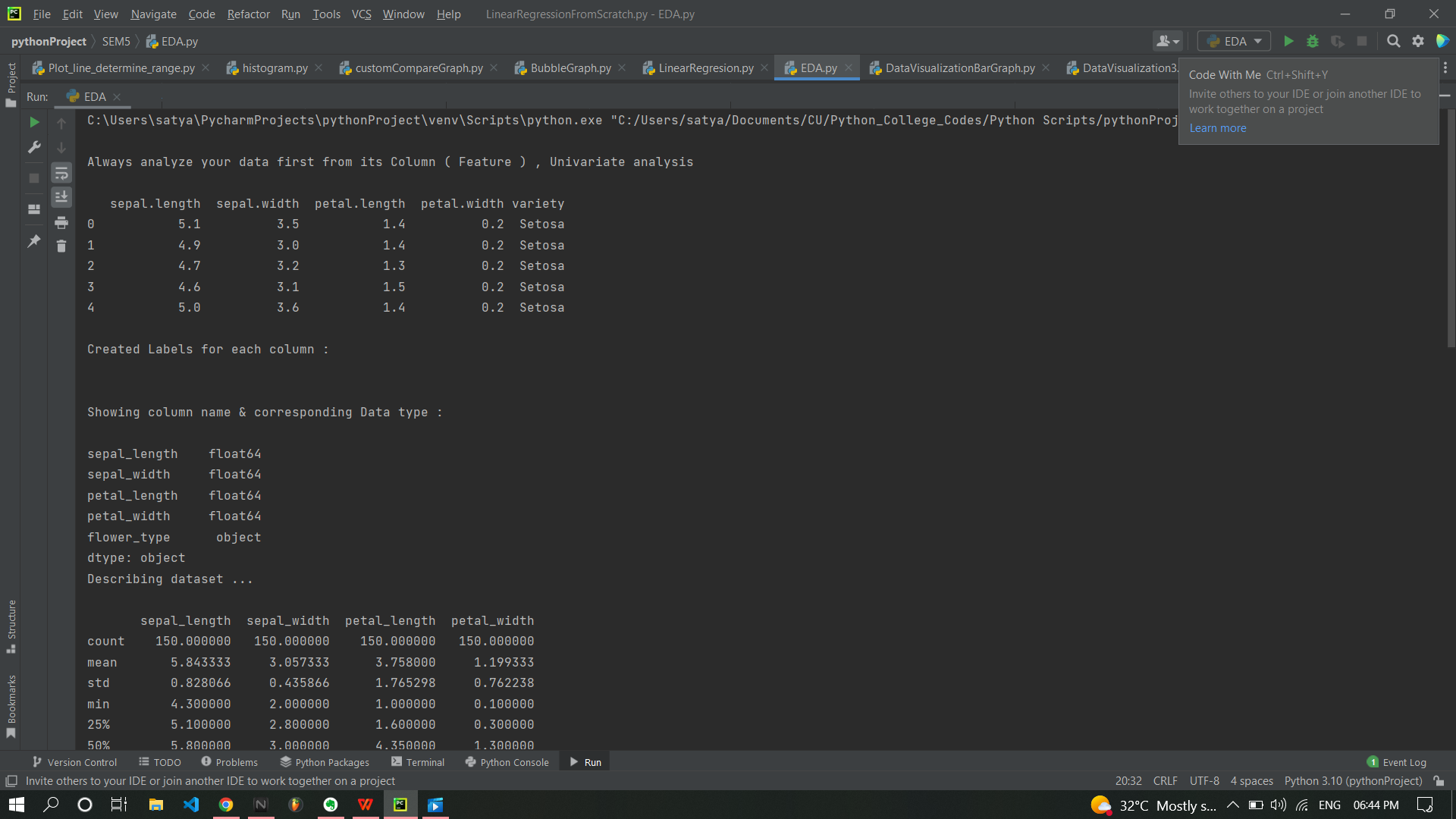
**3. CODE:**

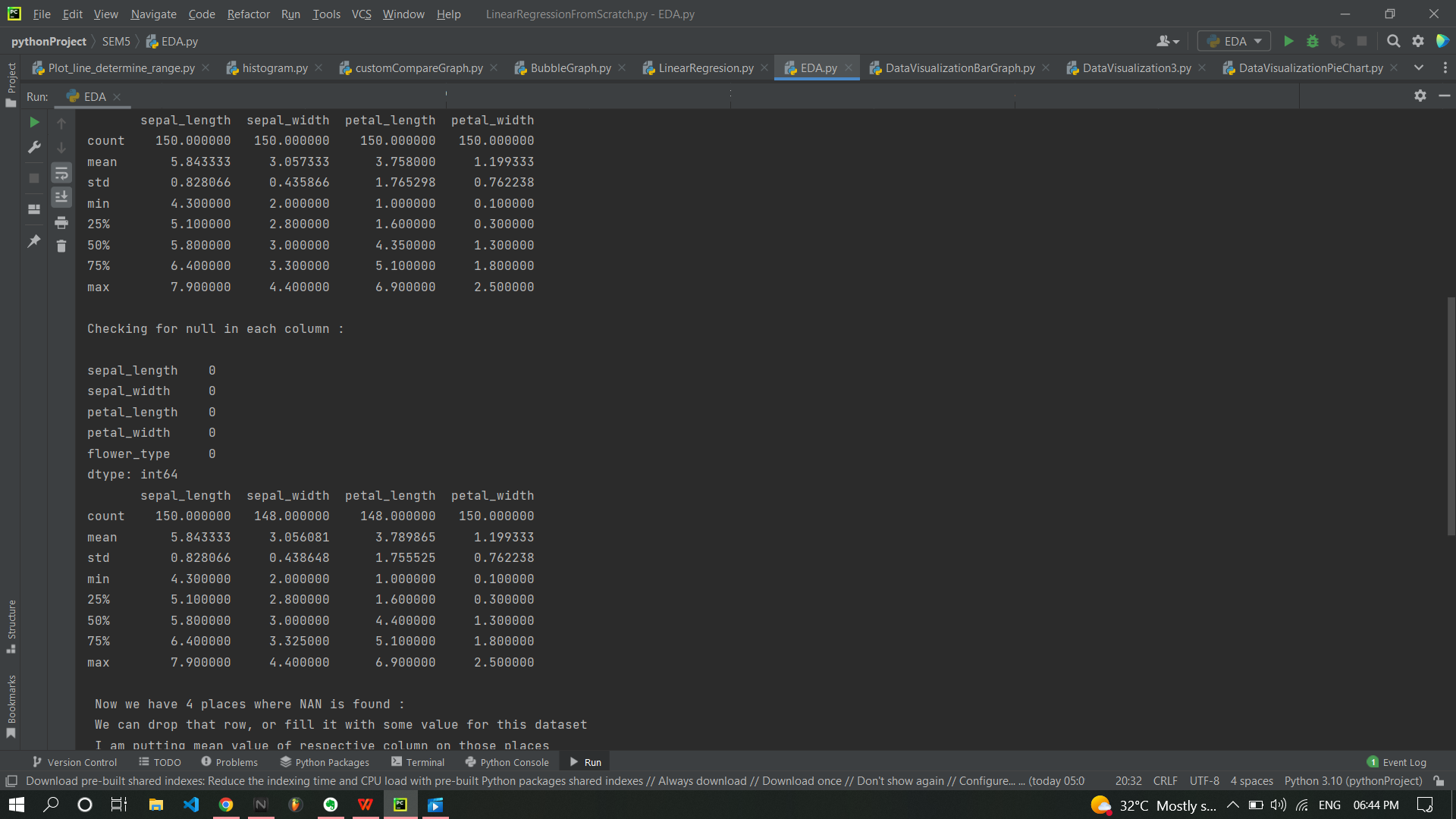
#CHIRAG BITHER

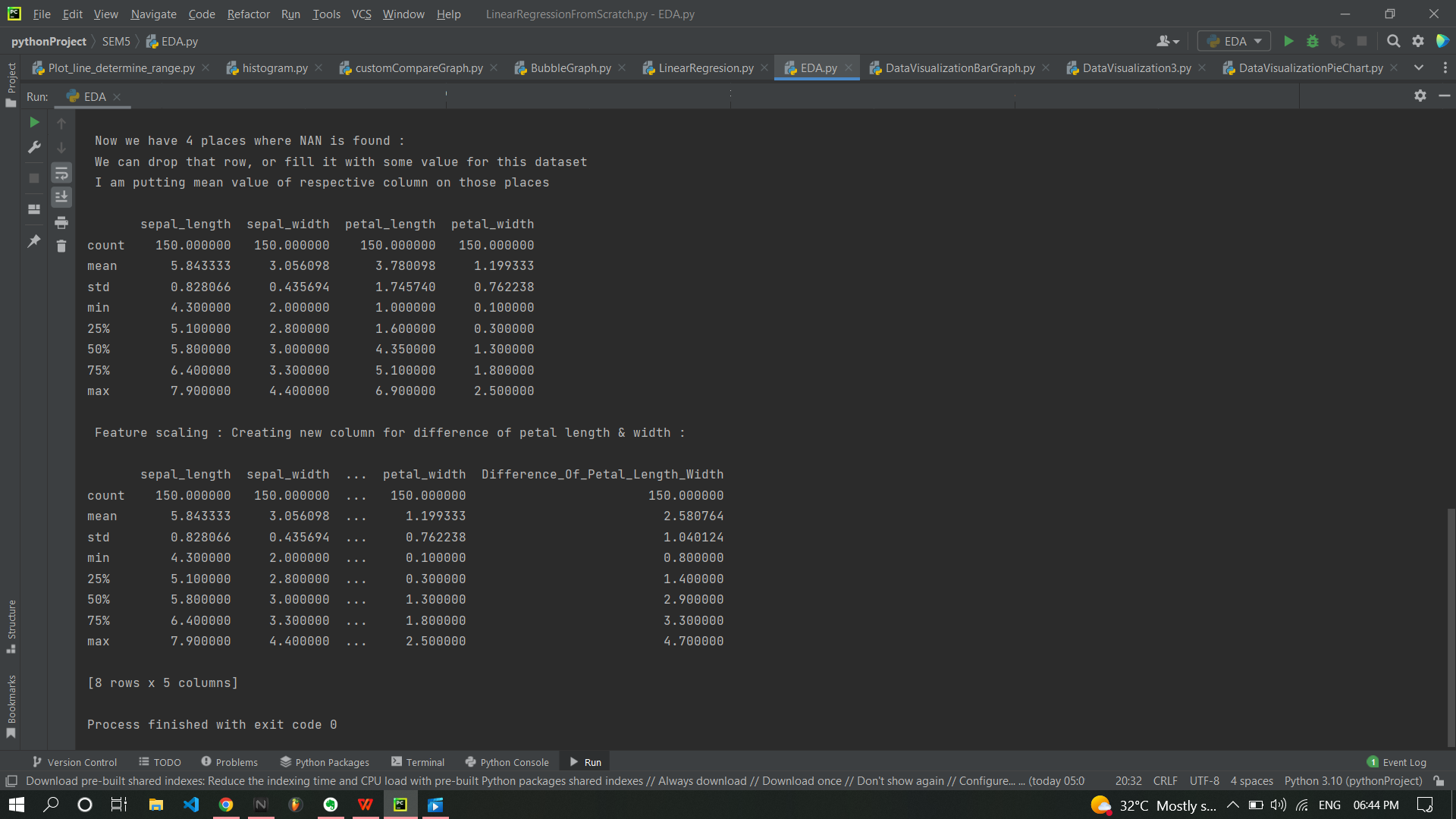
#20BCS1838

#Pandas :  
import pandas as pd  
import numpy as np  
# Dataset is loaded into an object i.e : iris  
  
iris = pd.read\_csv("C:\\Users\\satya\\Downloads\\iris.csv")  
  
# By doing this you will get another object of iris with the same value but different reference  
# Deep copy :  
dataFrame = iris.copy()  
print("\nAlways analyze your data first from its Column ( Feature ) , Univariate analysis \n")  
print(dataFrame.head())  
iris.columns = ["sepal\_length","sepal\_width","petal\_length","petal\_width","flower\_type"]  
print("\nCreated Labels for each column :\n")  
dataFrame.columns = ["sepal\_length","sepal\_width","petal\_length","petal\_width","flower\_type"]  
# It will show column name & corresponding Data type :  
print("\nShowing column name & corresponding Data type :\n")  
print(dataFrame.dtypes)  
# Gives us statistical frequently used information about our data ONLY in Number columns :  
print("Describing dataset ...\n ")  
print(dataFrame.describe())  
# To access particular column you can write df.column  
# To check is there any null in column :  
print("\nChecking for null in each column : \n")  
print(dataFrame.isnull().sum())  
  
# Handling NAN :  
# Creating NAN so that we can learn how to handle this problem :  
# row & column specifically with iloc : row as slicing & column as slicing  
dataFrame.iloc[2:4,1:3] = np.nan  
print(dataFrame.describe())  
  
print("\n Now we have 4 places where NAN is found :")  
print(" We can drop that row, or fill it with some value for this dataset ")  
print(" I am putting mean value of respective column on those places\n ")  
  
# Putting mean value inplace of NAN :  
dataFrame.sepal\_width.fillna(iris.sepal\_width.mean(),inplace = True)  
dataFrame.petal\_length.fillna(iris.sepal\_width.mean(),inplace = True)  
  
print(dataFrame.describe())  
  
# Adding new feature : ( Feature scaling )  
print("\n Feature scaling : Creating new column for difference of petal length & width : \n")  
  
dataFrame["Difference\_Of\_Petal\_Length\_Width"] = dataFrame["petal\_length"] - dataFrame["petal\_width"]  
  
print(dataFrame.describe())

1. **OUTPUT:**

****

Here we can see that total 4 count is less that we made so that we can fill it with mean value. ****

****

1. **Learning outcomes (What I have learnt):**

* **Application of numpy library in loading dataset as well as creating & handling NaN values.**
* **Application of Pandas in the form of Data Frames with various functions.**
* **Learned how data scientist creates inferences out of dataset with feature scaling prediction.**
* **Learned how to deal with un-pre-processed dataset.**

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |