**The Oxford College of Science**

**Department of Computer Science and Applications**

Machine Learning Lab Manual

**NEP- BCA Sixth Semester LAB- (CA-C29L)**

**(NEP Scheme) 2023-2024**

**R. Deepa**

**Assistant Professor**

***In***

***R.DEEPA***

Program 1:

**Install and Set up Python and Essential Libraries Like NumPy and Pandas.**

**Installation of Python**

**Step 1 : Search for Python**

**Python download**” in the Google search bar and press Enter key. In the list of links shown, select the very link or click on the official website link: <https://www.python.org/downloads/>



**Step 2: Select Version to Install Python**

Choose the correct link for your device from the options provided:

either **Windows installer (64-bit)** or **Windows installer (32-bit)** and proceed to download the executable file.



**Step 3 : Downloading the Python Installer**

Once you have downloaded the installer, open the .exe file, such as python-3.11.5-amd64.exe, by double- clicking it to launch the Python installer.

Choose the option to install the launcher for all users by checking the corresponding checkbox, so that all users of the computer can access the Python launcher application.

Enable users to run Python from the command line by checking the Add python.exe to PATH checkbox.

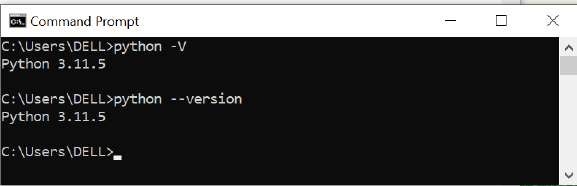
After Clicking the **Install Now Button** the setup will start installing Python on your Windows system.



**Step 4: Verify the Python Installation in Windows**

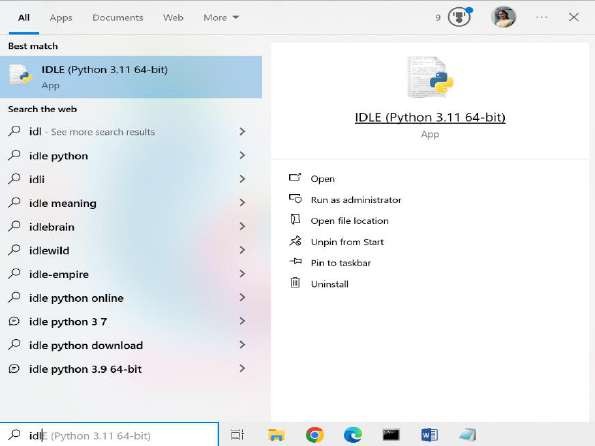
After successful installation of Python, close the installation window. You can check if the installation of Python was successful by using either the command line or the Integrated Development Environment (IDLE), which you may have installed. To access the command line, click on the Start menu and type **“cmd”** in the search bar.

Then click on **Command Prompt**, type the command “**python –V**” or “**python –version**”. You can see installed version of Python on your system.



Go to Python Integrated Development Environment (IDLE). In Windows search bar, type IDLE and you can see “**IDLE (Python 3.11.64- bit)**”.

Open IDLE on the IDLE screen itself you can see version. This gives the conformation of successful installation of python.

**Installation of essential packages NumPy and Pandas.**

**Install Numpy Package**

**NumPy** is an open-source Python library that facilitates efficient numerical operations on large quantities of data.

There are a few functions that exist in **NumPy** that we use on pandas Data Frames. The most important part about NumPy is that pandas is built on top of it which means NumPy is required for operating the Pandas.

It is defined as a Python package used for performing the various numerical computations and processing of the multidimensional and single-dimensional array elements.

The calculations using NumPy arrays are faster than the normal Python array. It is also capable of handling a vast amount of data and convenient with Matrix multiplication and data reshaping. Steps to

**Install NumPy is,**

**Step 1:** Open command prompt, CMD. **Step 2:** Type the command,

# **C:\Users> py –m pip install numpy**

Or

**C:\Users\>pip3 install numpy**

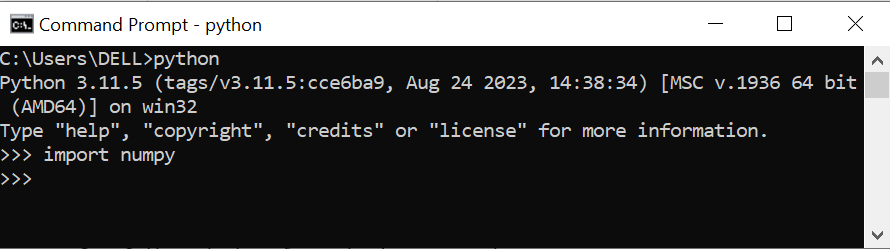
Step 3: Upgrade the software by the command

**C:\Users\DELL> py –m pip install - -upgrade pip Or**

**C:\Users\DELL> pip3 install - -upgrade pip**

**Step 4:** On the successful installation, you can type following two commands at command prompt. If python prompt “>>>” appears then package is successfully installed.

1. **C:\Users\DELL>python**
2. **>>>import numpy**



**Install Pandas Package**

**Pandas** is a very popular library for working with data (its goal is to be the most powerful and flexible open- source tool, and in our opinion, it has reached that goal).

**Data Frames** are at the center of pandas. A Data Frame is structured like a table or spreadsheet. The rows and the columns both have indexes, and you can perform operations on rows or columns separately.

It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., load, manipulate, prepare, model, and analyze. Steps to install pandas is,

Step 1: Open command prompt, CMD.

Step 2: Type the command,

**C:\Users\> py –m pip install Pandas**

Or

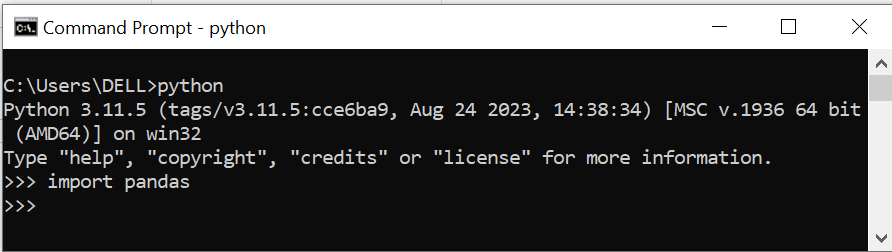
**C:\Users\DELL>pip3 install pandas**

**Step 3:** Upgrade the software by the Command

##### **C:\Users> py –m pip Install - -upgrade pip**

Or

**C:\Users\DELL> pip3 install - -upgrade pip**



**step 4:** On the successful installation, you can type following two commands at command prompt. If python

Prompt  **“>>>”** appears then package is successfully installed.

**C:\Users\DELL>python**

**>>>import pandas**

**AB- (CA-C29L)**

**(Neme) 2023-2024**

***R.DEEPA***

**Faculty in Computer Science**

**Lab Program 3: Install and set up scikit-learn and other necessary tools.**

PIP is a **package manager for Python**, which means it **allows you to install and manage libraries** and dependencies that are supplemental to the standard library. (A package contains all the files you need for a module, and modules are Python code libraries that you can include in your projects.)

PIP3 is also a package manager, **designed to replace PIP** to solve few problems caused by it. Latest versions of python 3.x allows the use of pips command for installing python libraries.

**Scikit-learn (Sklearn) Library:**

Scikit-learn is the most useful machine learning library. It provides modules for data analysis and statistical modelling. It provides a wide range of efficient **tools such as classification, regression, and clustering and dimensionality reduction** via a consistence interface in Python. This library, which is largely written in Python, **is built upon following essential libraries: NumPy, Pandas, SciPy and Matplotlib libraries.**

**Install numpy library**

**Step 1:** Open command prompt, CMD. **Step 2:** Type the command, **C:\Users\DELL> py –m pip install numpy** Or

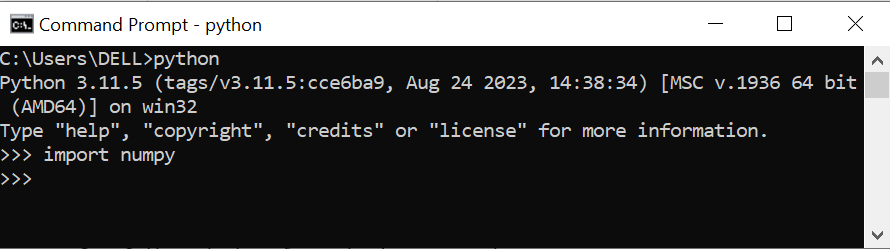
**C:\Users\DELL>pip3 instal numpy**

**Step 3:** Upgrade the software by the command **C:\Users\DELL> py –m pip install - -upgrade pip** Or

**C:\Users\DELL> pip3 install - -upgrade pip**

**Step 4:** On the successful installation, you can type following two commands at command prompt. If python prompt “>>>” appears then package is successfully installed.

1. **C:\Users\DELL>python**
2. **>>>import numpy**



**Install pandas library**

**Step 1:** Open command prompt, CMD. **Step 2:** Type the command, **C:\Users\DELL> py –m pip install pandas** Or

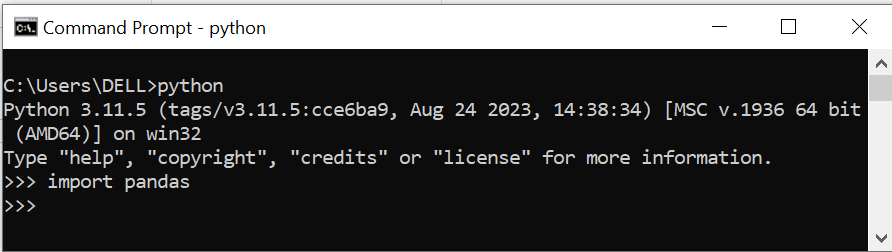
**C:\Users\DELL>pip3 install pandas**

**Step 3:** Upgrade the software by the command **C:\Users\DELL> py –m pip install - -upgrade pip** Or

**C:\Users\DELL> pip3 install - -upgrade pip**

**Step 4:** On the successful installation, you can type following two commands at command prompt. If python prompt “>>>” appears then package is successfully installed.

1. **C:\Users\DELL>python**
2. **>>>import pandas**



**Install Sciy library**

**Step 1:** Open command prompt, CMD. **Step 2:** Type the command, **C:\Users\DELL> py –m pip install SciPy** Or

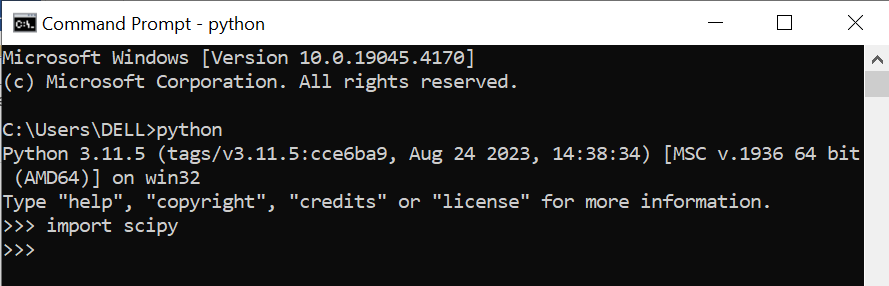
**C:\Users\DELL>pip3 install SciPy**

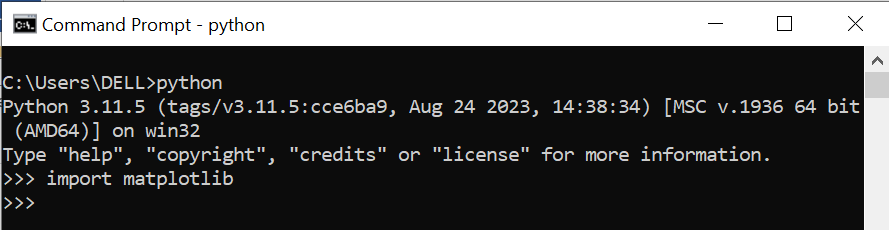
**Step 3:** Upgrade the software by the command **C:\Users\DELL> py –m pip install - -upgrade pip** Or

**C:\Users\DELL> pip3 install - -upgrade pip**

**Step 4: On the successful installation, you can type following two commands at command prompt. If python prompt “>>>” appears then package is successfully installed.**

1. **C:\Users\DELL>python**
2. **>>>import scipy**





**Install scikit- learn (sklearn) library**

**Step 1:** Open command prompt, CMD.

**Step 2:** Type the command,

**C:\Users\DELL> py –m pip install sklearn**

Or

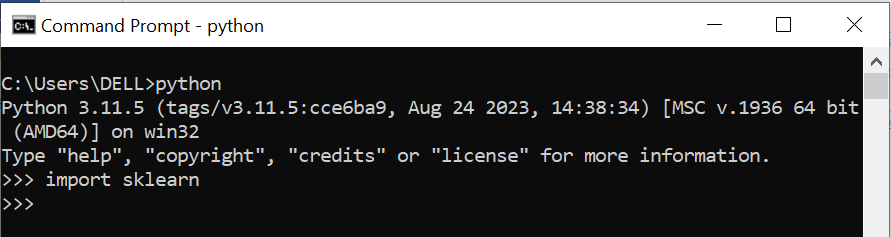
**C:\Users\DELL>pip3 install sklearn**

**Step 3:** Upgrade the software by the command **C:\Users\DELL> py –m pip install - -upgrade pip** Or

**C:\Users\DELL> pip3 install - -upgrade pip**

**Step 4:** On the successful installation, you can type following two commands at command prompt. If python prompt “>>>” appears then package is successfully installed.

1. **C:\Users\DELL>python**
2. **>>>import sklearn**



1. **Introduce sci-kit-learn as a machine learning library.**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

**Installation**

If you already installed NumPy and Scipy, the following are the two easiest ways to install scikit-learn −

**Using pip**

The following command can be used to install sci-kit-learn via pip

**pip install -U scikit-learn**

**Features**

Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modelling the data. Some of the most popular groups of models provided by Sklearn are as follows −

**Supervised Learning algorithms** − Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.

**Unsupervised Learning algorithms** − On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

**Clustering** − This model is used for grouping unlabelled data.

**Cross Validation** − It is used to check the accuracy of supervised models on unseen data.

1. **Install and set up scikit-learn and other necessary tools.**

scikit-learn, a powerful Python library for machine learning. Here are the steps to set it up:

Install Python: If you haven’t already installed Python, download and install the latest version of Python 3 from the official Python website.

Install scikit-learn using pip: Open your terminal or command prompt and run the following command:

pip install -U scikit-learn

To verify your installation, you can use the following commands:

python -m pip show scikit-learn

# To see which version and where scikit-learn is installed

python -m pip freeze

# To see all packages installed in the active virtual environment

import numpy import pandas import SciPy import sklearn

import matplotlib

print("numpy library version is: ")

print(numpy. version ) #please type two underscore symbols.

print("numpy library is successfully installed")

print(" ")

print("pandas library version is: ") print(pandas. version )

print("pandas’ library is successfully installed")

print(" ")

print("SciPy library version is: ")

print(SciPy. version )

print("SciPy library is successfully installed")

print(" ")

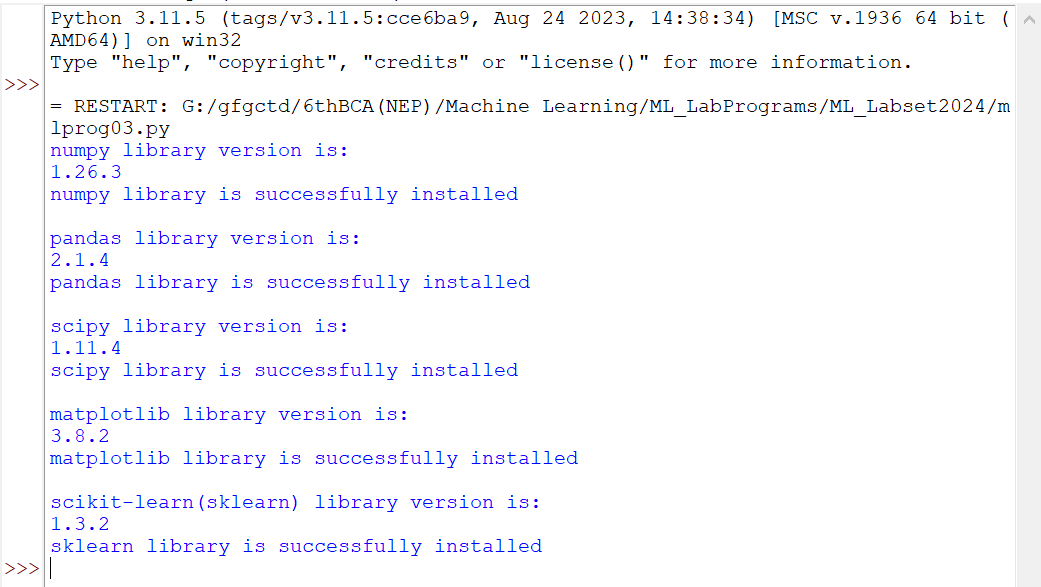
print("matplotlib library version is: ) print(matplotlib. version )

print("matplotlib library is successfully installed") print(" ")

print("scikit-learn(sklearn) library version is: ") print(sklearn. version )

print("sklearn library is successfully installed")

**Program Output:**



**4.Write a Program to Load and Explore the Dataset of .CVS and Excel files Using Pandas.**

StudentDT.csv

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| StuEnroll | UniversityCode | UniversityName | CollegeID | CollegeName | CourseID | CourseName | Admin  Year | Sub  stream | PUPresent | Obtained |
| U0001 | B101 | Bangalore University | C0001101 | Oxford | 1101 | BCA | 2021 | CS | 90 | 80 |
| U0002 | B102 | Bangalore University | C0002102 | Christ | 2102 | B.SC | 2022 | CS | 85 | 70 |
| U0003 | B103 | Bangalore University | C0003103 | AMC | 3103 | MCA | 2020 | CA | 90 | 85 |
| U0004 | B104 | Bangalore University | C0004104 | Dayanandha | 4104 | BCA | 2021 | CA | 95 | 92 |
| U0005 | B105 | Bangalore University | C0005105 | SFS | 5105 | B.SC | 2020 | CS | 90 | 85 |
| U0006 | B106 | Bangalore University | C0006106 | T.John | 1101 | BCA | 2021 | CS | 90 | 80 |
| U0007 | B107 | Bangalore University | C0007107 | Christ | 2102 | B.SC | 2022 | CS | 85 | 70 |
| U0008 | B108 | Bangalore University | C0008108 | Oxford | 3103 | MCA | 2020 | CA | 90 | 85 |
| U0009 | B109 | Bangalore University | C0009109 | Dayanandha | 4104 | BCA | 2021 | CA | 95 | 92 |
| U0010 | B110 | Bangalore University | C0010110 | SFS | 5105 | B.SC | 2020 | CS | 90 | 85 |

**Program : 4**

#python program to load csv file

import pandas as pd

import openpyxl

csv\_file = "StudentDT.csv"

excel\_file = "StudentDT.xlsx"

df\_csv = pd.read\_csv(csv\_file)

print("CSV Data:")

print(f"File Name: {csv\_file}")

print(df\_csv.info())

print(df\_csv.head())

workbook = openpyxl.load\_workbook(excel\_file)

worksheet = workbook["StudentDT"]

print("\n Excel Data:")

print(f"File Name: {excel\_file}")

first\_row = next(worksheet.iter\_rows(min\_row=1, max\_row=1))

print("Column Names:")

for cell in first\_row:

print(cell.value)

**Output**

**CSV Data:**

**File Name: StudentDT.csv**

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 10 entries, 0 to 9

Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 StuEnroll 10 non-null object

1 UniversityCode 10 non-null object

2 UniversityName 10 non-null object

3 CollegeID 10 non-null object

4 CollegeName 10 non-null object

5 CourseID 10 non-null int64

6 CourseName 10 non-null object

7 AdminYear 10 non-null int64

8 Substream 10 non-null object

9 PUPresent 10 non-null int64

10 Obtained 10 non-null int64

dtypes: int64(4), object(7)

memory usage: 1012.0+ bytes

None

Stu Enroll UniversityCode UniversityName ... Substream PUPresent Obtained

0 U0001 B101 Bangalore University ... CS 90 80

1 U0002 B102 Bangalore University ... CS 85 70

2 U0003 B103 Bangalore University ... CA 90 85

3 U0004 B104 Bangalore University ... CA 95 92

4 U0005 B105 Bangalore University ... CS 90 85

5 rows x 11 columns]

Excel Data:

File Name: StudentDT.xlsx

Column Names:

StuEnroll

UniversityCode

UniversityName

CollegeID

CollegeName

CourseID

CourseName

AdminYear

Substream

PUPresent

Obtained

**\*\*\*\* \*\*\*\*\*\***

**iris.csv**

1. **Write a program to Visualize the dataset to gain insights using Matplotlib or Seaborn by plotting scatter plots, and bar charts.**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

def visualize\_dataset(file\_path):

# Load the dataset into a pandas DataFrame

df = pd.read\_csv(file\_path)

# Assuming it's a CSV file, change accordingly if it's an Excel file

# Plot scatter plots

sns.pairplot(df)

plt.title("Pairplot of the Dataset")

plt.show()

# Plot bar chart for categorical column (assuming the first column is categorical)

if df.iloc[:, 0].dtype == 'object':

sns.countplot(x=df.columns[0], data=df)

plt.title("Bar Chart of Categorical Column")

plt.xlabel(df.columns[0])

plt.ylabel("Count")

plt.show()

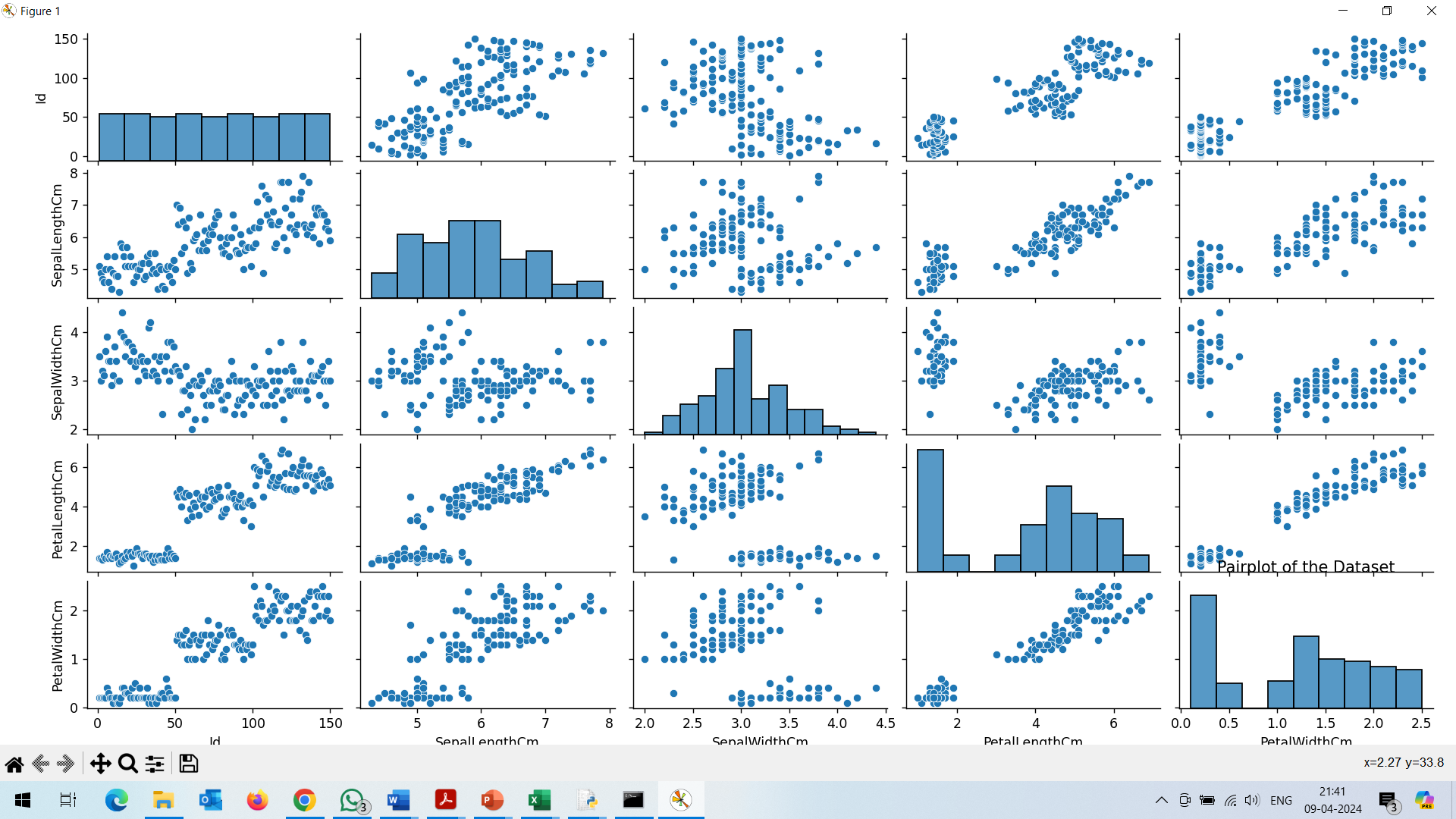
else:

print("No categorical column found to plot bar chart.")

# Example usage

file\_path = 'IRIS.csv' # Change this to the path of your CSV file

visualize\_dataset(file\_path)



1. **Write a program to implement a k-Nearest Neighbours (k-NN) classifier using scikit-learn and Train the classifier on the dataset and evaluate its performance.**

**Example : 1**

from sklearn import datasets

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

from sklearn.neighbors import KNeighborsClassifier

# Load an example dataset (iris dataset)

iris = datasets.load\_iris()

X\_train, X\_test, y\_train, y\_test = train\_test\_split(iris.data, iris.target, test\_size=0.3)

# Initialize and train a classifier (K-Nearest Neighbors) clf = KNeighborsClassifier(n\_neighbors=3) clf.fit(X\_train, y\_train)

# Evaluate the classifier

clf = KNeighborsClassifier(n\_neighbors=3)

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

accuracy = clf.score(X\_test, y\_test)

print(f"Accuracy: {accuracy}")

**Output:**

**Accuracy: 0.9777777777777777**

**Example : 2**

import numpy as np

import pandas as pd

from sklearn.datasets import load\_iris

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

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score, classification\_report

# Load Iris dataset

iris = load\_iris()

X = iris.data

y = iris.target

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize the k-NN classifier

k = 3

# Number of neighbors

knn\_classifier = KNeighborsClassifier(n\_neighbors=k)

# Train the classifier

knn\_classifier.fit(X\_train, y\_train)

# Make predictions on the testing set

y\_pred = knn\_classifier.predict(X\_test)

# Evaluate the classifier's performance

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

# Display classification report

print("Classification Report:")

print(classification\_report(y\_test, y\_pred, target\_names=iris.target\_names))

**Output :**

**Accuracy : 1.0**

**Classification Report:**

**precision recall f1-score support**

**setosa 1.00 1.00 1.00 10**

**versicolor 1.00 1.00 1.00 9**

**virginica 1.00 1.00 1.00 11**

**accuracy 1.00 30**

**macro avg 1.00 1.00 1.00 30**

**weighted avg 1.00 1.00 1.00 30**