ANNAMALAI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

B. E. (COMPUTER SCIENCE & ENGINEERING) DATA SCIENCE

SEMESTER – IV

DSCP409. DATA SCIENCE LAB

LABORATORY MANUAL

(JANUARY 2023 – MAY 2023)

LAB INCHARGE:

Dr. R. Ragupathy, Associate Professor, Dept. of CSE, A.U

ANNAMALAI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DSCP 409. DATA SCIENCE LAB

COURSE TEACHER: Dr. R. Ragupathy, Associate Professor, Dept. of CSE, AU

LIST OF EXPERIMENTS

CYCLE - I

1. STUDY OF PYTHON DATA SCIENCE ENVIRONMENT
2. OPERATIONS ON PYTHON DATA STRUCTURES
3. ARRAY OPERATIONS USING NUMPY
4. OPERATIONS ON PANDAS DATAFRAME
5. DATA CLEANING AND PROCESSING IN CSV FILES
6. HANDLING CSV FILES
7. HANDLING HTML AND EXCEL FILES

CYCLE - II

8. PROCESSING TEXT FILES
9. DATA WRANGLING (PIVOT TABLE, MELT, CONCAT)
10. GENERATING LINE CHART AND BAR GRAPH USING MATPLOTLIB
11. DISPLAY DATA IN GEOGRAPHICAL MAP
12. DISPLAY DATA IN HEATMAP
13. NORMAL AND CUMULATIVE DISTRIBUTION
14. HYPOTHESIS TESTING

ADDITIONAL EXERCISES

1. GENERATION OF FACTOR PAIRS OF A GIVEN INTEGER
2. AVERAGE POOLING ON A GIVEN *n x n* MATRIX WITH A *m x m* KERNEL

Ex. No. 1

STUDY OF PYTHON DATA SCIENCE ENVIRONMENT

AIM:

To study the Python Data Science Environment (NumPy, SciPy, Pandas, Matplotlib).

PROBLEM DEFINITION:

Study the features of Python, packages required for data science operations and their installation procedure required for Data Science programming.

a) PYTHON DATA SCIENCE ENVIRONMENT

Data Science is a branch of computer science that deals with how to store, use and analyze data for deriving information from it. Analyzing the data involves examining it in ways that reveal the relationships, patterns, trends, etc. that can be found within it. The applications of data science range from Internet search to recommendation systems to customer services and Stock market analysis. The data science application development pipeline has the following elements: Obtain the data, wrangle the data, explore the data, model the data and generate the report. Each element requires skills and expertise in several domains such as statistics, machine learning, and programming. Data Science projects require a knowledge of the following software:

PYTHON: Python is a high-level, interpreted, interactive and object-oriented scripting language that provides very high-level dynamic data types and supports dynamic type checking. It is most suited for developing data science projects.

NUMPY: NumPy provides n-dimensional array object and several mathematical functions which can be used in numeric computations.

SCIPY: SciPy is a collection of scientific computing functions and provides advanced linear algebra routines, mathematical function optimization, signal processing, special mathematical functions, and statistical distributions.

PANDAS: Pandas is used for data analysis and can take multi-dimensional arrays as input and produce charts/graphs. Pandas can also take a table with columns of different datatypes and may input data from various data files and database like SQL, Excel, CSV.

MATPLOTLIB: Matplotlib is scientific plotting library used for data visualization by plotting line charts, bar graphs, scatter plots.

b) INSTALLATION OF PYTHON AND DATA SCIENCE PACKAGES

The following documentation includes setting up the environment and executing programming exercises targeted for users using Windows 10 with Python 3.7 or later version. Steps should work on most machines running Windows 7 or 8 as well.

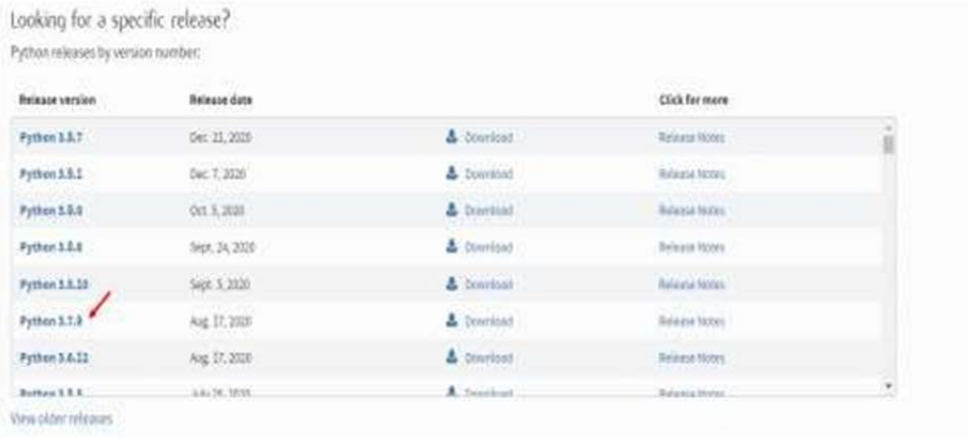
Sections that are indicated as optional are marked with **[Optional]**. Though optional, students are strongly encouraged to try out these sections.

We use the default python package management system - pip to install packages through one may prefer to install using conda.

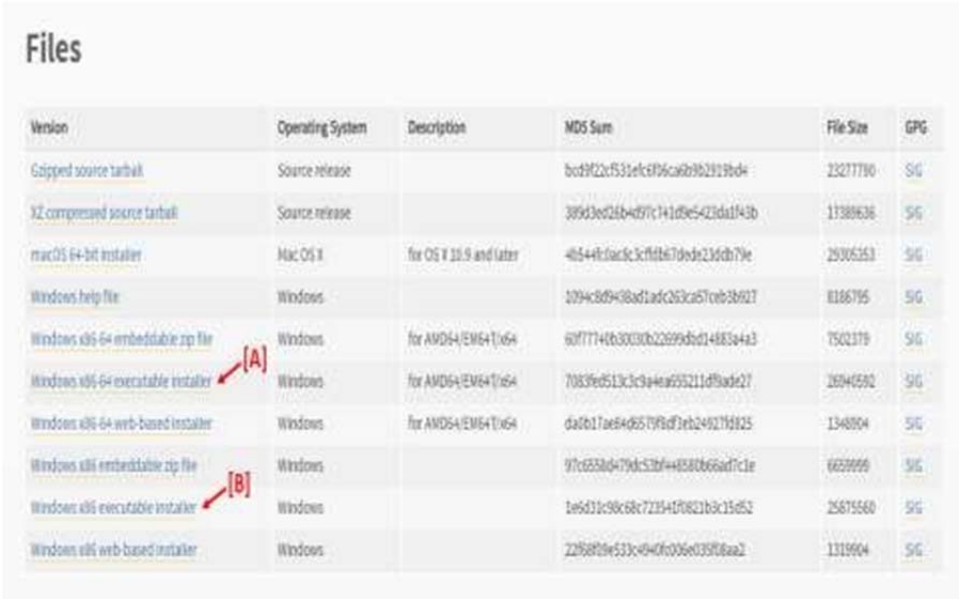
Setting up Environment:

Python:

1. To install Python 3 on Windows, navigate to https://www.python.org/downloads/ on your web browser, download and install the desired version.
2. For example to install Python 3.7.9:
    1. Navigate to https://www.python.org/downloads/
    2. Scroll down to “Looking for a specific release?” section and click on Python 3.7.9 as shown below:



    c. Scroll down to “Files” section and click on “Windows x86-64 executable installer” (Indicated [A]) if running a 32 bit machine or “Windows x86 executable installer” (indicated [B]) if running a 64 bit machine. If not sure if your machine is 32 or 64 bit, we recommend installing the 32 bit version.



    d. Double click the downloaded exe to run the installer. Follow the prompts on the screen and install with default options.

3. To verify installation, go to Start->Command Prompt. Type in “python --version” and hit Enter key. This will display “Python 3.7.9” or similar in the next line. If you do not see this or see any other error, please revisit the above steps.
4. Advanced Windows users or users facing issues can refer to https://docs.python.org/3/using/windows.html
5. To install Python on other distributions refer to:
a. Macintosh OS: https://docs.python.org/3/using/mac.html
b. Unix distros: https://docs.python.org/3/using/unix.html

Additional Resource:
https://docs.python.org/3/installing/index.html#basic-usage

pip

Python installation comes with a default package management/install system (pip - “pip installs Package”). Make sure to verify this by:

1. Start->Command Prompt.
2. Type in “pip --version” and hit Enter key.
3. This will display “pip 20.0.2 from “c:\users\DELL\appdata\local\programs\python\python37\lib\site-packages\pip (python 3.7)” or similar in the next line.

Virtual Environment (venv) [Optional]

Follows steps from here to install/use virtual environment:
https://docs.python.org/3/tutorial/venv.html#creating-virtual-environments

Jupyter Notebook [Optional]

Jupyter Notebook is a web based interactive development environment, usually preferred for quick prototyping.

To install:

1. Start->Command Prompt.
2. Type in “pip install jupyter” and hit Enter key.

To use:

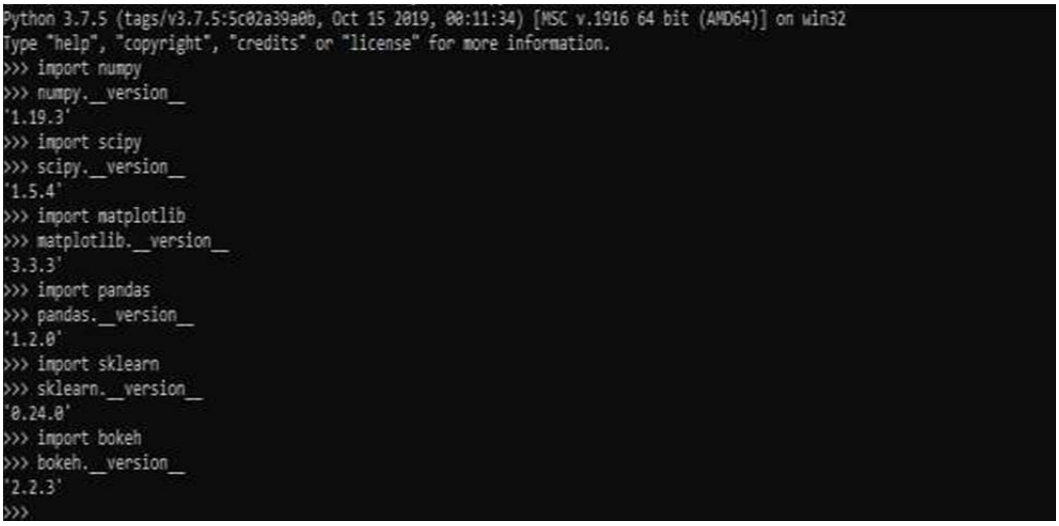
1. In Command Prompt, type “jupyter notebook” and hit Enter key.
2. By default a web browser tab with jupyter notebook will open. If not, type in the following URL to open - http://localhost:8888/tree
3. Do not close this Command Prompt opened in Step 1.
4. Click on New -> Python 3 (right top) to open a new Notebook.
5. To close (also called as “Shut down Jupyter”), close all newly created notebook tabs and click on “Quit”.

More on Jupyter Notebooks at https://jupyter.org/

Packages

We will install the following packages: numpy, scipy, matplotlib, pandas, scikit-learn (sklearn), bokeh.

1. Start->Command Prompt.
2. Type in “pip install numpy” and hit Enter key**.
   **If one encounters issue with installing/using numpy, try “pip install numpy==1.19.3”
3. Type in “pip install scipy matplotlib pandas sklearn bokeh” and hit Enter key.
4. To verify installation:
   a. Type in “python”, hit enter.
   b. Type in
      import <package_name>
      <package_name>.__version___
   c. This will display the desired package with it’s version number if properly installed as indicated below:



RESULT:

A study on the Python Data Science environment was carried out to understand and install the software packages required for Data Science experiments.

Ex. No. 2

OPERATIONS ON PYTHON DATA STRUCTURES

AIM:

To develop Python programs to perform operations on Python Data Structures such as String, List, Tuple, Dictionary, and Set.

(a) STRINGS

PROBLEM DEFINITION:

Check if the given pair of words are anagram using sorted() function. Print “True” if it is an anagram and “False” if not.

CODE:

```
def fn_test_anagram(string1, string2):
    string1_sorted = sorted(string1.lower())
    string2_sorted = sorted(string2.lower())
    if(string1_sorted == string2_sorted):
        return True
    else:
        return False


if __name__ == "__main__":
    input1 = "Binary"
    input2 = "Brainy"
    print(fn_test_anagram(input1, input2))
```

TEST CASE:

CASE 1: INPUT: Listen, Silent OUTPUT: True

CASE 2: INPUT: Chin, Inch OUTPUT: True

CASE 3: INPUT: Binary, Brainy OUTPUT: True

CASE 4: INPUT: About, Other OUTPUT: False

(b) DICTIONARY, LIST

PROBLEM DEFINITION:

Generate a dictionary of words and the corresponding number of times it occurred in a given sentence. Print the occurrence when the user enters a word and 0 if a word is not found. (Ignore ‘,’, ‘.’ and ‘?’)

CODE:

```
def fn_clean_string(test_string, list_to_remove):
    test_string = test_string.lower()
    for item in list_to_remove:
        test_string = test_string.replace(item, "")
    return test_string

def fn_word_frequency(test_string):
    word_list = test_string.split()
    word_count = []
    for word in word_list:
        word_count.append(word_list.count(word))
    word_freq_dict = dict(list(zip(word_list, word_count)))
    return word_freq_dict

def fn_display_count(test_word, word_freq_dict):
    test_word = test_word.lower()
    if test_word in word_freq_dict.keys():
        return word_freq_dict[test_word]
    else:
        return 0

if__name__== "__main__":
    input_string = "She sells seashells on the sea shore. The shells she sells are seashells, I'm
sure. And if she sells seashells on the sea shore, Then I'm sure she sells seashore shells."
    list_to_remove = [".", ",", "?"]
    clean_string = fn_clean_string(input_string, list_to_remove)
    word_freq_dict = fn_word_frequency(clean_string)
    test_word = "Shells"
    print(fn_display_count(test_word, word_freq_dict))
```

TEST CASE:

CASE 1: INPUT: Shells OUTPUT: 2

CASE 2: INPUT: The OUTPUT: 3

CASE 3: INPUT: Sea shell OUTPUT: 0

CASE 4: INPUT: Shore. OUTPUT: 0