CS251 OUTLAB - 3
Python, Numpy and Scipy
(Autumn-2021)

(Max Marks : 100)

General Instructions:

- Make sure you know what you write, you might be asked to explain your code at a later point in time.
- The submission will be graded automatically, so stick to the naming conventions and follow directory structure strictly.
- In cases where output is required to be submitted in the form of .csv/.txt/.png make sure that your **code** should **generate** the file at desired location at runtime.
- The deadline for this lab is **Saturday, 21th August, 11:59 PM.**
- After creating your directory, package it into a tarball:**rollno1-rollno2.tar.gz**
- **Command to generate tarball:**
  tar -czvf rollno1-rollno2.tar.gz rollno1-rollno2/
- **Incorrect directory structure will lead to marks penalty**. **Please make sure we see the exact directory structure as shown above when we decompress the archive.**
- Test this on an SL2 machine if you wish.
- Don't put rubbish like hidden files, temporaries or backup files in the archive.
- Please understand that with great ease you can make grading assignments a nightmare for the TAs. Be responsible and considerate -- this is a two way street.
- Submit once only per team from the moodle account of the smallest roll number.
- **The directory structure should be as follows (nothing more nothing less).** Quote major references in `references.txt`.
- **Please Note that your program should create the output file as and when stated in the problem statement and make sure that your submission should not contain any such output file i.e. when we will test your program, your program should create and save it at required location at that time only.**

Directory Structure:

```
rollno1-rollno2/
    |- python
        |-q1_a
            |-data
                |-testcase1
                    -2008.txt
                    -2012.txt
```

```
                              -2016.txt
                  |-testcase2
                              -2020.txt
                              -2024.txt
              -olympics.py
        |-q1_b
              |-data
                    -uid_list.txt
                    -web_list.txt
              |-output
                    -winner.txt (to be created by your program at run time)
              -win.py
   |- numpy
        |-q2_a
              |-data
                    -matrix.csv
              -operations1.py
              -operations2.py
              -operations3.py
              -operations4.py
              -operations5.py
        |-q2_b
              |-data
                    -pca_data.txt
              |-output
                    -out.png        (to be created by your program at runtime)
                    -transform.csv(to be created by your program at run time)
              -pca.py
   |- scipy
        -q3_a.py
   - references.txt
```

NOTE: All the commands given below considers that terminal has been launched from inside of rollno1-rollno2 directory.

1. Python

a. Olympics: (20 Marks)

The Olympics had just wrapped up with India winning it's record highest number of medals . So, let us solve a problem related to medal counts. We are providing you the files with the names like **year.txt** i.e. **2008.txt** , **2012.txt**, **2016.txt**, etc. You are supposed to read these files one by one which contains data in the following format:

CountryName-#Goldmedals-#Silvermedals-#Bronzemedals
.
.
.
CountryName-#Goldmedals-#Silvermedals-#Bronzemedals

Read **#Goldmedals** as the **Number of gold medals** won by that country,and the same applies for other notations as well. Below given image represents the data from one of the test cases.

```
australia-14-15-17
japan-9-8-8
italy-8-9-10
kenya-6-4-6
poland-4-5-2
romania-4-1-4
canada-3-9-8
usa-36-39-37
```

Every information that you will read from these files, is going to be in string format. So, please do consider appropriate type conversions. Once you are done with this read part then for every country, you need to store the count of medals separately i.e. for **Gold**,**Silver** and **Bronze** for all the **year.txt** files at one place.

Now we want you to sort the country names based on their Gold medal count(**in decreasing order**) and in case of same number of gold medals between two countries tie breaking should be done based on the country names (**Just to avoid any sort of confusion between capital letters and small letters , we are providing you the data in small letters only**).

For example :
Say, **japan** won **20 Gold medals** and **India** won **20 Gold medals** (Yeah day will definitely come: **APNA TIME AYEGA**)

Note: Medal count on which sorting needs to be done is the combined sum from all the year.txt files.

Then output should be like :
india(should come first)
japan(should come second)
i.e. output should be in **lexicographic order**.In final output, you should output your dictionary as shown in the **output of sample run** below.

We want you to provide the relative address path through command prompt for better understanding you can look into the following link for how to provide arguments through command prompt Argument_parser_reference.

Sample run:
Files:
---2020.txt
india-1-2-4
japan-1-40-23
usa-20-30-40

--2024.txt
japan-1-20-50
indonesia-1-30-40
india-1-23-45
China-2-3-4

Command line **query to run program**:
python3 ./python/q1_a/olympics.py --path ./python/q1_a/data/testcase2/

Output for the above files(2020.txt and 2024.txt):
{'usa': [20, 30, 40], 'china': [2, 3, 4], 'india': [2, 25, 49], 'japan': [2, 60, 73], 'indonesia': [1, 30, 40]}

Explanation :
One can clearly see that in the above output **usa has the maximum number of gold medals** so **usa has been ranked first** and likewise others but in case of **ties**

like for the case between china, india and japan, they all have equal number of gold medals. So, they have been **ranked lexicographically**. Try looking for the lambda functions for the implementation. Even new methods are also accepted but do not use any library here other than os,argparse.

b. Lakshmi Chit Fund !!!   (20 Marks)

Lakshmi Chit Fund is running as one of the most successful online lucky draws. They have got various sponsors from different websites. The sponsors have posted their advertisement links on the Lakshmi Chit Fund website. Now the sponsors want to award prizes to users who click on the entertainment and gaming link.

You will be given a **web_list.txt** file. This file contains the websites URL that are selected for lucky draws. One URL per line in the format "https://site_path". (without double quotes)
(Example : **https://www.hotstar.com**)

Also you will be given a **uid_list.txt** file. This file has one entry per line. Each entry has "**username**" followed by "**password**" followed by "**IP Address**" followed by "**Website URL**" , separated by "**||**" (double pipe symbol) as shown below :

 "**user_name||pass_wd||ip_address||url**"   (without double quotes)
(Example : **pranshuiitb||you will never know||10.12.1.2||hotstar.com**)

Example:**uid_list.txt**

```
 1 pransh9 ||hipassword||10.25.55.220||yahoo.com
 2 94sheet||loveangel||10.34.45.203||ymail.com
 3 Paroni01||ruhanika||10.22.11.218||http://www.pandora.com/
 4 Ishiru10 ||firewall ||10.12.33.188||http://www.youtube.com/
 5 Karishma45||nexa123||10.33.12.102||www.allmusic.com/newreleases
 6 Arya01||hero001||10.22.32.112||www.myntra.com||
 7 66Swarna||classic350||10.22.33.201||allmusic.com
 8 Eva5||thunder50||10.23.55.222||www.gmail.com||
 9 Namit87||bulletlover||10.12.14.140||gamespot.com/games/pc/
10 Shiv01||hfdlx666||10.20.30.160||www.Shockwave.com
11 00Tarishi||whitehat66||10.33.12.203||ymail.com
12 Taris34||gungarrage||10.21.31.245||http://www.tvtattle.com/
13 Urv||urkarsh888||10.22.24.211||www.pogo.com/games/a-way-with-words
14 Vansh77||pranvi143||10.21.43.190||https://www.Agar.Io
```

Example : **web_list.txt**

```
1   https://www.gamespot.com/games/pc/
2   https://www.allmusic.com/newreleases
3   https://www.pogo.com/games/a-way-with-words
4   https://www.Agar.Io
5   https://www.A10.Com
6   https://www.Tankionline.Com
7   https://www.Newgrounds.Com
8   https://www.Bigfishgames.Com
9   https://www.Girlsgogames.Com
10  https://www.BoredGames.com
```

Task 1 : Read "**web_list.txt**" file and print in terminal by removing "**https://**" and other extra paths.   **(6 Marks)**
Note : 'web_list.txt' is guaranteed to have a complete url starting from https .
Example :- "https://www.hotstar.com/entertainment/movies"

should print only "www.hotstar.com".

```
www.gamespot.com
www.allmusic.com
www.pogo.com
www.Agar.Io
www.A10.Com
www.Tankionline.Com
www.Newgrounds.Com
www.Bigfishgames.Com
www.Girlsgogames.Com
www.BoredGames.com
```

Example :

Task 2 : Read user_name, ip_address, url from "**uid_list.txt**" and check if that URL belongs to "**web_list.txt**", if yes then print :   **(7 Marks)**

print(f"user_name<space>-<space>{user_name}<space>:<space>Winner<space>- Lucky draw!!!<space>-<space>{url_complete})".

Ex : user_name - Karishma45 : Winner - Lucky draw!!! - https://www.allmusic.com/newreleases

Note : In place of **<space>** you need to **replace it** with **single space.**

At last print the number of **lucky winners.**

Note : URLs in "uid_list.txt" may have truncated URLs like www.google.com **or just** google.com **, you need to check both cases.**

Example :

```
user_name - Karishma45 : Winner - Lucky draw!!! - https://www.allmusic.com/newreleases
user_name - Namit87 : Winner - Lucky draw!!! - https://www.gamespot.com/games/pc/
user_name - Urv : Winner - Lucky draw!!! - https://www.pogo.com/games/a-way-with-words
user_name - Vansh77 : Winner - Lucky draw!!! - https://www.Agar.Io
4
```

Query : if **'web_list.txt'** contains **https://www.gamespot.com/games/pc** , then **'uid_list.txt'** urls should **completely match the later part** , i.e. **gamespot.com/games/pc** is matching with entry in **web_list.txt** so it should be selected and hence to be printed and copied to **'winner.txt'** and other entries if any in **uid_list.txt** like **gamespot.com/games** should not be printed nor added to **winner.txt** file.

Task 3 : User code should generate a "**winner.txt**" inside output directory, which will have one entry per line containing: "**user_name||ip_address||complete_url**" of users who won lottery awards. **(7 Marks)**
Truncated_url : **"www.allmusic.com/newreleases"**
complete_url : **"https://www.allmusic.com/newreleases"** .(Since urls in
uid_list.txt may contain truncated urls, you need to report complete_url appended with appropriate 'https://' and 'www' symbols)
Example :

```
1  Karishma45||10.33.12.102||https://www.allmusic.com/newreleases
2  Namit87||10.12.14.140||https://www.gamespot.com/games/pc/
3  Urv||10.22.24.211||https://www.pogo.com/games/a-way-with-words
4  Vansh77||10.21.43.190||https://www.Agar.Io
5
```

Note:
● user_name, pass_w, ip_address, url are separated by **||** symbol so it will not be present in above attributes in test cases.
● Your program should create a winner.txt file when you run your program. Please make sure that you do not submit winner.txt as a file in your submission.

Command to run win.py file:

python3 ./python/q1_b/win.py --path1 ./python/q1_b/data/web_list.txt --path2 ./python/q1_b/data/uid_list.txt --output ./python/q1_b/output/

2. Numpy

a. Play with Numpy   (25 Marks)

Your friend Rockstar is too good at coding and playing with arrays. He always makes fun of you at coding assessments, since you are a backlogger. You decided to take revenge with your friend. You know that Rockstar only knows C programming language and you are comfortable with python. You know that "**Numpy**" is a magic wand in python and can work on different functions very easily. So you decided to challenge your friend and one who completes , task will be considered "**Coding God**".

Task : You will have a .csv file (**matrix.csv**) which is a two dimensional matrix, you have to read the csv file and convert it into a numpy array and perform various operations to be coded in these python files named as **opertions1.py,opertions2.py…..   Operations5.py** for the tasks **Task1, Task2….Task5** respectively.

Example : Given matrix :
```
[[ 5  5 84  3  9]
 [ 6 11  1 55 58]
 [ 1 20 48 12 36]
 [ 8  4 41 93 98]
 [ 6 17 64  0 13]]
```

Task 1 : Print only elements belonging to the upper diagonal (including diagonal element) in column wise fashion. Code in **operations1.py.**   (5 Marks)

Command to run operations1.py file:

python3 ./numpy/q2_a/operations1.py --path ./numpy/q2_a/data/matrix.csv

Example output :
```
5
5 11
84 1 48
3 55 12 93
9 58 36 98 13
```

Task 2 : Print **mean, median, standard deviation along x axis with precision = 2, determinant and inverse of matrix in order**. (Note : You need to print the inverse if the determinant is non-zero, but if the determinant is zero there is something in numpy called **pseudo inverse** (Moore Penrose Pseudo inverse link) which needs to be printed .) Code in **operations2.py**.   **(5 Marks)**

Command to run operations2.py file:

python3 ./numpy/q2_a/operations2.py --path ./numpy/q2_a/data/matrix.csv

Example output :

```
[ 5.2 11.4 47.6 32.6 42.8]
[ 6.  11.  48.  12.  36.]
[ 2.32  6.34 27.6  36.09 32.73]
1821200.9999999788
[[-0.58 -0.53 -0.26  0.37  0.74]
 [ 0.23  0.29  0.08 -0.19 -0.24]
 [ 0.05  0.03  0.01 -0.02 -0.05]
 [ 0.32  0.34  0.06 -0.21 -0.34]
 [-0.29 -0.31 -0.05  0.2   0.29]]
```

Task 3 : Sort the original array along the x-axis, y-axis and flatten array, and print them in order. Code in **operations3.py**.   **(3 Marks)**

Command to run operations3.py file:
python3 ./numpy/q2_a/operations3.py --path ./numpy/q2_a/data/matrix.csv

Example output :

```
[[ 1  4  1  0  9]
 [ 5  5 41  3 13]
 [ 6 11 48 12 36]
 [ 6 17 64 55 58]
 [ 8 20 84 93 98]]
[[ 3  5  5  9 84]
 [ 1  6 11 55 58]
 [ 1 12 20 36 48]
 [ 4  8 41 93 98]
 [ 0  6 13 17 64]]
[ 0  1  1  3  4  5  5  6  6  8  9 11 12 13 17 20 36 41 48 55 58 64 84 93
 98]
```

Task 4 : Flatten 2D array into 1D array, **print list of unique elements of array in sorted order**(as shown in the **first line of output**) with their respective **frequencies in next line**(as shown in the **second line of output**) and report **frequency of second largest element**(here 93 is the second largest element with frequency 1) in next line(as shown in the **third line of output**). Note : **Only** using **numpy** operations **(No looping allowed).**   **(6 Marks)**

Command to run operations4.py file:
python3 ./numpy/q2_a/operations4.py --path ./numpy/q2_a/data/matrix.csv

Example output :

```
[ 0  1  3  4  5  6  8  9 11 12 13 17 20 36 41 48 55 58 64 84 93 98]
[1 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
1
```

Task 5 : Padding is a common operation in image processing. You will be given **num (int)** as a command line argument and you need to pad value 0 in top,bottom,right and left that will lead to increase in the row and column size by **2*num**. Ex: if num = 1 (given below), count of rows as well as columns increase by two one each for the top and bottom. Similarly do this on the left and right. Code in **operations5.py**.   **(6 Marks)**

Command to run operations4.py file:
python3 ./numpy/q2_a/operations5.py --path ./numpy/q2_a/data/matrix.csv --num 1

Example output :

```
[[ 0  0  0  0  0  0  0]
 [ 0  5  5 84  3  9  0]
 [ 0  6 11  1 55 58  0]
 [ 0  1 20 48 12 36  0]
 [ 0  8  4 41 93 98  0]
 [ 0  6 17 64  0 13  0]
 [ 0  0  0  0  0  0  0]]
```

Note : Use only numpy,pandas packages only. (NumPy,Pandas)

b. Principal Component Analysis (PCA)   (25 Marks)

PCA, is a popular statistical **dimensionality reduction** technique in Machine Learning.It uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. Necessary formulas for conversion are given here in the link.

Input : You will be given a txt file (**pca_data.txt**) which will have N lines and each line has D comma separated values. Read the .txt file and generate the N x D matrix from it.

Now, for applying PCA on this N x D matrix. You can follow the below given set of steps for PCA :-

Step 1: Standardize the N x D matrix.

Step 2: Calculate the covariance matrix for the D (dimensions) in the matrix .

Step 3: Calculate the eigenvalues and eigenvectors for the covariance matrix.

Step 4: Sort eigenvalues and their corresponding eigenvectors and **print sorted eigenvalues** on **terminal**.

Step 5: Pick K=2 (K<D) eigenvalues and form a matrix of eigenvectors.

Step 6: Transform the original matrix and store N x K transformed matrix in the directory "**output**" with the name '**transform.csv**'

Step 7: Plot the projected data (using a scatter plot from matplotlib) and save the plot to the output directory as **'out.png'** [This saving to file should be automatically done by the code]. While plotting, ensure both the x and y axis have the same aspect, and show values from [-15,15].

NOTE: Your program should save out.png and transform.csv file when you run your program i.e. at run time only. Please make sure that you do not submit out.png and transform.csv as a file in your submission. When we will test your program it should generate these files at the required place.

Command to run pca.py file:

python3 ./numpy/q2_b/pca.py --path ./numpy/q2_b/data/pca_data.txt --output ./numpy/q2_b/output/

Output : 1. Sorted eigenvalues   (5 Marks)
        2. Save Transformed matrix in 'output/transform.csv'   (10 Marks)
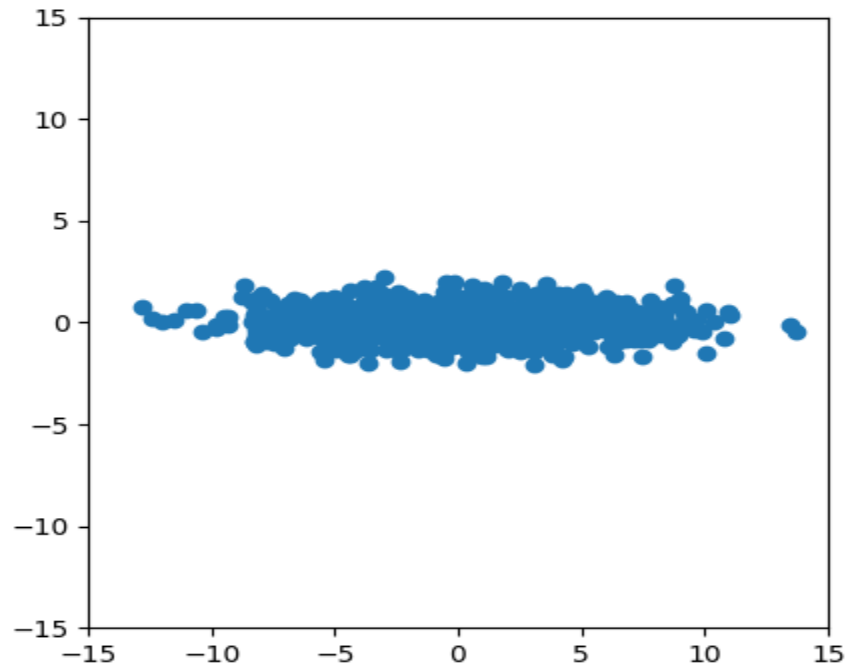   **3. Save scatter plot in 'output/out.png'   (10 Marks)**

Finally you have reduced D dimensions into K (K<D) dimensions, which carry the same information as D dimensions. **Well congratulations you have implemented the PCA algorithm from scratch!!!**

Output : Eigenvalues.

```
[1.72610170e+01 5.22514214e-01 1.64274296e-15 9.77282229e-17]
```


Note : Use only **os,argparse,numpy,pandas and matplotlib** ([link](#)).

Do not use scikit or other similar packages for implementing PCA functions. We want you to learn all the intricacies related to numpy library and this is one of the standard examples for learning.

3. Scipy

a. Integration: (10 Marks)

You all have solved a lot of calculus problems during JEE preparation. Now, Let us get back to scientific calculation but guess what not on paper but on the computer. Yeah, this question will require the use of the scipy library. You can use the following reference link for better understanding of the inbuilt library functions *(you can use other resources as well from the web, no restrictions on that but while using them please do put the reference in your reference.txt)*.

We want that you should integrate the following integral:

$$\int_0^{\frac{1}{2}} \int_0^3 \int_0^4 x^2 y^2 z^2 \, dx\,dy\,dz$$

The answer should be the only thing that you should print on the console screen(i.e. Only the integration result).Use scipy and lambda functions only.

Command to run the program:
python3 ./scipy/q3_a.py 0 0.5 0 3 0 4

NOTE:
- **We will check your program on a different set of integral limits but only on this function having triple integral.**
- **We know that the scipy function uses approximations for performing the integration like if you solve some integral and you got 15 as an answer on your notebook then this is possible that your program may give 14.9999995. So, no issues with that we will handle this difference.**