**Assignment 2**

**NumPy, Pandas, and Matplotlib Basics**

**Question 1:**

Create a 2-D numpy array with following entries, and print the values.

1 4 7

2 6 12

3 8 17

1. Save the array into a text file named *dummy2Darray.txt*
2. Read the text file into another 2D array using numpy.
3. Alter the values of the second row as the square of the corresponding entries of the first row, i.e., the second row will be the square of
4. Plot the values of the first and second rows of the modified matrix as a line graph using Matplotlib, such that the first row will be the x-values, and the second row will be the corresponding y-values. ( so, the line graph will look like a parabola)

**Question 2:**

Consider the 2D array provided in question 1. Read three integer values from user, and check if these three values match with any row or any column of the given 2D array. If yes, print the row/column number, else print -1.

**Question 3:**

Create two n\*n 2D arrays with random integer values using numpy, *consider n=5 for this first case*. Compute their inner product using *for loops*. Print the output and the time taken by the computation. Next, use the numpy built-in function, if any, to compute the same and print the time. Also print the time difference.

Increase the value of n upto 50, in steps of 5, and do the same job. In each step, print the difference between time taken by the for-loop and numpy built-in function to compute the inner product.

**Question 4:**

Create one n\*n 2D array and one 1\*n vector with random values using numpy. Perform an operation that creates a n\*n matrix as output, such that,

the ith column of the output matrix = the ith column of the input matrix \* the ith entry of the vector. Use numpy built-in function.

For example:

Input matrix: Input vector: Output matrix:

2.9 3.9 [ 3 8] 8.7 31.2

5 10 15 80

**Question 5:**

Create a random n\*n square matrix, where the value of n to be provided by user using numpy. Without using numpy built-in function, create a *boundary* on all sides of the 2D matrix with zeroes as shown in the example:

Original matrix: Modified matrix:

2 7 0 0 0 0

8 17 0 2 7 0 0 8 17 0

0 0 0 0

Now search if there is any numpy built-in function which will do the same job.

**Question 6:**

Create a n\*n 2D array with random values, such that the values are drawn from a gaussian distribution, using numpy. Compute row-wise mean, column-wise mean and the mean of the entire matrix using numpy and print the outputs. Also compute the row-wise and column-wise standard deviation and print the outputs.

**Question 7:**

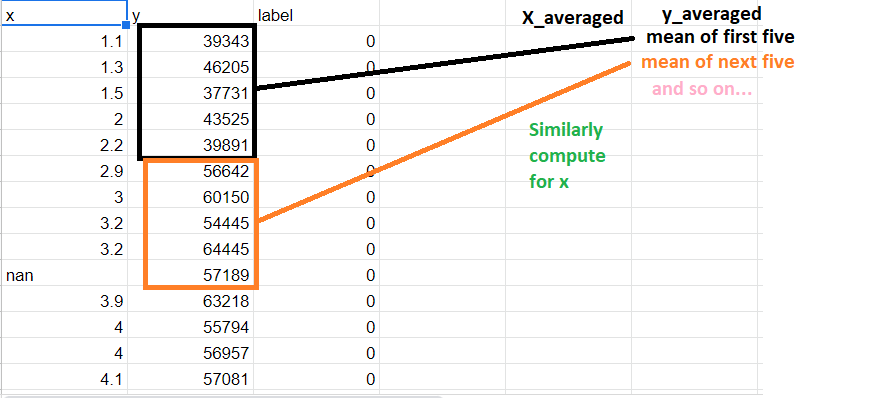
Read the dataset named ‘dataset\_2\_duplicate.csv’ in a dataframe using pandas and print the first 5 rows. Now, modify the entries of the last column, such that ‘yes’ is replaced by the integer 1, and ‘no’ is replaced by the integer 0. Save the modified dataset in .xlsx format.

**Question 8:**

Read the dataset named ‘dataset\_1\_duplicate.xlsx’ in a dataframe using pandas. Check if any column contains ‘nan’ values. If yes, then compute the mean of the remaining entries of that column and replace the ‘nan’ entries with this column value. Save the modified dataset named as “dataset\_1\_processed.csv”.

**Question 9:**

Read the modified dataset named “dataset\_1\_processed.csv”, generated in the Question 8. Compute the mean of each 5 consecutive entries of the columns named as x and y, and store them in a numpy array, as shown in the picture below:



Now draw a scatter plot using Matplotlib for x\_averaged and y\_averaged columns.

**Question 10:**

The heights of the people of south-east Asia can be assumed as normally distributed with a mean height of 172 cm, and standard deviation of 11 cm. Create a numpy array, the entries of which indicates the height of 500 persons, and these heights are normally distributed with mean equals 172, and standard deviation of 11 cm. Now plot a histogram of this data and observe the nature of plot.