National University of Singapore
School of Computing
CS5242: Neural Networks and Deep Learning

Tutorial 1:
PYTHON 2

Third-Party Python Libraries
a. NumPy (for low-level math operations)
b. Pandas (for data loading and manipulation)
c. Matplotlib and Seaborn (for data visualization)

NumPy

The purpose of this tutorial is to familiarize with NumPy array creation, its operations like slicing, indexing, transposing and some important mathematical functions.

- NumPy is an N-dimensional array type called `ndarray`.
- It describes the collection of items of the same type. Each element in `ndarray` is an object of data-type object (called `dtype`)
- Images are loaded into NumPy array

1. Create a 1-D NumPy array `‘A’`, from list `[5 8 10 5 4 3 2 1]`, check its parameters (`ndim, shape, itemsize, dtype`)

2. Create a 2-D NumPy array `‘B’`. Check its parameters

```
 B = [[ True False]
      [False True]]
```

3. Create a 1-D NumPy array `‘C’` of size `4 * 1`, consisting of all ones i.e.

```
  C = [[1.0]
       [1.0]
       [1.0]
       [1.0]]
```

4. Create a 2-D NumPy array, `‘D’` of size `4 * 5`, consisting of number from `0,1,2,3,4….20`

```
  D = [[ 0  1  2  3  4]
       [ 5  6  7  8  9]
       [10 11 12 13 14]
       [15 16 17 18 19]]
```

5. Add array `‘C’` to array `‘D’`. (You will be doing broadcasting!!)

6. Transpose the 2-D array `‘D’`

7. From array `‘D’` use NumPy 2-D slicing to extract below slice

```
  [[ 0  2  4]
   [10 12 14]]
```

8. From array ‘D’ extract the corner elements using NumPy Indexing
```
    [[ 0  4]
     [15 19]]
```
9. Modify the ‘D’ array to make center elements 0 i.e.
```
        D = [[ 0  1  2  3  4]
             [ 5  0  0  0  9]
             [10  0  0  0  14]
             [15 16 17 18 19]]
```

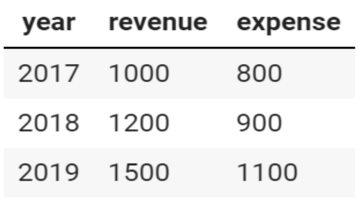
10. Initialize a `5 * 4` numpy array ‘E’ with random numbers. Now find the maximum number in all columns and its position. Find the `mean` and `std` of the matrix. Now multiply the matrix ‘E’ and matrix ‘D’.

11. Load an image (Tutorial-1-sample.jpg provided) into numpy array. Commands to load an image to numpy array is ‘`import cv2; img = cv2.imread('Tutorial-1-sample.jpg')`’. Now extract a `100*100` matrix from the image and show it with matplotlib. Command to show an image:‘`from matplotlib import pyplot as plt; plt.imshow(img)`’

Pandas

- There are two central data structures of Pandas are **Series** and **DataFrame**
- Foundation of a DataFrame is a Series. The docstring of DataFrame defines a DataFrame as “`Can be thought of as a dict-like container for Series objects`”
- So what is a **series**? A Pandas series can be conceptualized in two ways.
  - It can be envisioned as a single column of tabular data.
  - It can also be envisioned as a single row of tabular data

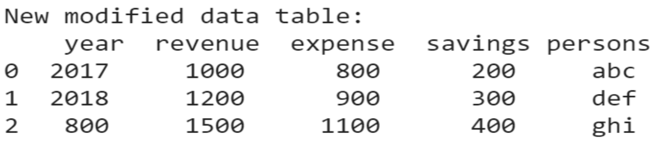
Let’s assume there is a database table called accounting which stores revenue and expenses across different years.



year	revenue	expense
2017	1000	800
2018	1200	900
2019	1500	1100

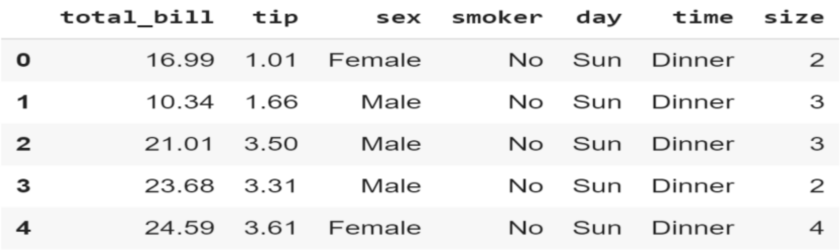
1. Create the table above with Pandas dataframe object.

2. Modify the above table to add ‘savings’ and ‘persons’ as shown below.
‘saving’ = ‘revenue’ - ‘expenses’



```
New modified data table:
    year  revenue  expense  savings persons
0  2017     1000      800      200     abc
1  2018     1200      900      300     def
2   800     1500     1100      400     ghi
```

3. Load a CSV file (Tutorial-2-tips.csv provided) to load a CSV file into a pandas dataframe. Now display ‘`.head()`’ of dataframe. Then display `.description()` of dataframe. Name this dataframe as `‘tips_table’`. Sample `.head()` of tips table.



	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

MATPLOTLIB and SEABORN

- ❖ Seaborn is built on top of Python’s core visualization library Matplotlib.
- ❖ It is meant to serve as a complement, and not a replacement.
- ❖ Seaborn works well with NumPy and Pandas data structures

1. Write a program to draw a scatter_plot with `sb.scatterplot` taking `‘total_bill’` as x-axis,`‘tip’` as y-axis and `hue` as `‘time’`.

2. Write a program to plot a pie chart of given `‘days’` data in `‘tips_table’` table.

3. Plot a bar plot with `sb.barplot` taking `‘sex’` in x-axis and `‘total_bill’` in y-axis.

4. Plot the univariate distribution of various data in the `‘tips_table’` table with `sb.distplot`.

5. Plot the bivariate distribution of data `‘total_bill’ vs ‘tip’` with `sb.jointplot()`

6. To analyze relation between each data of the table to classify `‘time’` class try using `sb.pairplot(tips_table, hue=’time’)`

7. To analyze the relation of categorical data try to plot using `sb.stripplot` with `‘day’ as x`, and `‘total_bill’ as y`