CS-345/M45 Big Data and Machine Learning Lab Class

The lab classes consist of tasks that are designed to be completed during the lab sessions and during your own time. If you do not complete the tasks during the lab class then you should do them at home and have them ready to be marked off in the next lab class.

Every two weeks you will be given a new lab class sheet with new tasks. You should aim to get all tasks completed and marked off by staff in the lab class. You can only get the lab sheet signed off in the lab classes (not by email, etc.).

All lab tasks must be uploaded to Canvas before the deadline stated on each lab sheet. You will be required to show your submission to Canvas in order to be signed off for the lab.

Marks will be awarded if the tasks are marked off by staff during the session they are handed out; or if they are marked off by staff during the following session.

Each whole task is worth 1 mark. The challenge tasks are optional and are not worth extra marks. However, it is a good idea to attempt these as completing them will help strengthen your programming and problem solving skills. You may need to do a little research to find out how to complete the challenge tasks.

CS-345/M45 Lab Class 1 — Release date: 10am 28/09/2020 Due date: 10am 12/10/2020

This lab is about getting familiar with Python syntax and packages commonly found in computer vision and machine learning applications. We will look at basic numerical operations and data manipulation techniques, and go further into the use of mathematical and scientific orientated packages, including `jupyter` and `numpy`.

☐ Task 1.1

This task is about familiarizing yourself with basic Python syntax in order to facilitate the coming lab sessions. You are provided with a handout which details an introduction to programming in Python, read through this document and use it to answer the following tasks.

1. Download and study the Introduction to Python document from Canvas.
2. Create a new Python Notebook in Jupyter, `MyNotebook.ipynb`, which does the following in a single cell:
   - create a variable, `x`, which contains the value 345
   - calculate: $2x + 5$ and print the output value to the command line
3. Print the string "Hello World" to the command line.
4. Define a function $f(x, w, b) = wx + b$ and call it from another cell with $x = 345$, $w = 2$ and $b = 5$.
5. Comment the top of `MyNotebook.ipynb` with your name, date and student number **in a marked up cell**.

Please turn over

☐ Task 1.2

This task is about list structures, dictionaries, loops and conditionals within Python. Continue to add to `MyNotebook.ipynb`.

1. Write a loop which prints elements of a list, `myList` to the command line **in reverse**.
2. Define the function `equals100`(x) = `True if` $x = 100$, `else False` and call it from `MyNotebook.ipynb`.
3. Create a dictionary, `myDictionary`, which recreates the following information:

   `data_name` = "myData"
   `data` = ["cat", "dog", "fish"]
   `labels` = a list of 3 numbers.
4. Print out the contents of each key from `myDictionary`.

☐ Task 1.3

This task is about importing packages and using `numpy` for multi-dimensional arrays. Continue to add to `MyNotebook.ipynb`.

1. Import the `numpy` package to `MyNotebook.ipynb`. Imports should be made at the start of a file, following the Python style guide.
2. Create two randomly initialized 2D numpy arrays of integers with size 2×3 and 3×4 respectively. Hint: numpy.random.randint will provide the functionality you require here.
3. Perform a **matrix multiplication** of these two matrices and print the result to the command line.
4. Print the **first column** of the matrix to the command line.

☐ Task 1.4

This task will look at loading and plotting data in Python using the `numpy` and `matplotlib` packages.

1. Download the .npy file `Iris_data.npy` from Canvas and load the data in using the numpy `load` function.
2. Check you have a 2D numpy array by querying the shape of the loaded data.
3. Select 2 feature dimensions and plot the data using `matplotlib.pyplot`'s scatter function.
4. Give the axes suitable labels. It may be worth looking up the Fisher Iris dataset online to find names of the features you have selected.

☐ Challenge Task 1.5

Read more into the PEP-8 style guide for Python. This is a style convention which will help you to produce well-structured Python code. Go back through your `MyNotebook.ipynb` notebook and ensure it conforms to the style guide as much as possible.