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

Overview:

These labs are designed to get you familiar with the methods covered in the lectures, you will be applying a variety of techniques to a variety of different problems in order to gain a deeper practical understanding of the theoretical concepts covered.

It is important to note however they do not cover all the intricacies of the methodologies discussed, and as such you are **strongly** encouraged to read into how the methods are implemented and how various hyper-parameters may have effects on the performance of the methods. You will need to read the documentation for respective third party packages being used, and you will need to consider the problem being solved and how to approach a solution.

How the labs are delivered:

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. The release date and deadline are clearly provided at the top of each lab sheet, and this is also reflected on the Canvas page.

How the labs are solved:

In these labs you will create Python 3 Notebooks (.ipynb), using them to produce an executable notebook of code which solves the tasks outline on the lab sheets. You should really lean into the use of notebooks as a communication tool, and utilise the cell structure of a notebook to produce your solution.

Each lab sheet will have only a single .ipynb notebook submission which solves the tasks on the sheet. Do not submit a notebook per task. You can use Jupyter or Google Colab to produce your notebook, both allow you to export the notebook as an .ipynb file for submission, more details on these can be found on Canvas.

How the labs are assessed:

Each whole task is worth 1 mark and no partial marks are awarded. Each lab sheet may have a varying number of tasks to complete, but each will be clearly indicated on the sheet.

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 week's session. No marks will be awarded after the deadline. Labs will only be signed off in lab classes, not by email etc.

Important: In addition to getting your solution signed off in the lab session, you **MUST** upload your singular. ipynb file to Canvas before the deadline stated on each lab sheet. This is a precautionary measure to ensure that we have a record of your lab work, it is not for grading; so make sure you also get signed off in the lab session.

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
Total Marks: 4

Release date: 02/02/2024
Due date: 16/02/2024

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 science orientated packages, including jupyter and numpy. Although this may be a straightforward lab sheet for many, it is very important that we set a baseline for everyone. Moving forward in the labs will be more complex usage of the language and it is not feasible for us to be helping with small syntax issues.

D Task 1.1

This task is about familiarizing yourself with the notebook interface, and utilising 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. Create a new Python3 Notebook, called <student_number>_Notebook. ipynb, where the <student_number> is your student number.
2. Annotate the notebook with your name, date and student number **in a markdown cell.**
3. In a new cell, create a variable, x, which contains the value 345. Calculate $2x + 5^3$ and print the output value.
4. In a new cell, print the string "Hello World".
5. In a new cell, define a function $f(w,x,b) = wx$ +band call it from another cell with $x = 345$, $w = 2$ and $b = 5^3$.

□ Task 1.2

This task is about list structures, dictionaries, loops and conditionals within Python.

1. Create a list, my_list containing at least 5 short strings.
2. Write a loop which prints out the elements of the list **in reverse order.** Don't use List's reverse () method.
3. Define the function equals_100(x) = True if $x = 100$, else False and test the function by calling it for $x = 99$ and $x = 100$.
4. Create a dictionary, my_dictionary, which recreates the following key-value pairings:
   data_name = "Animal counts"
   label = ["cat", "dog", "fish"]
   count = [2, 5, 10]
5. Print out the content of the count key by indexing into the dictionary.
6. Reproduce the following print out, using my_dictionary and an f-string:

   Animal name: dog, count: 5

Please turn over

□ Task 1.3

This task is about importing packages and using numpy for multi-dimensional arrays. Continue to add to your existing notebook. You can check Numpy's API documentation for more.

1. Import the numpy package to the notebook. Imports should be made at the top of the notebook, following the Python style guide.
2. Create two randomly initialized 2D numpy arrays of integers with size 2 x 3 and 3 x 4 respectively. Hint: numpy.random.randint will provide the functionality you require here.
3. Perform a **matrix multiplication** of these two matrices, storing the result to a variable and printing the result.
4. Print only the **first column** of the result matrix from the last subtask.

□ Task 1.4

This task will look at loading and plotting data in Python using the numpy and matplotlib packages. Again, check the documentation in order to explore what is available to you.

1. Download the .npy file 'Iris_data.npy' from Canvas and load the data in with numpy's load function. Hint: to upload data to colab: from google.colab import files uploaded=files.upload() or login to google drive and read files from gdrive
2. Check you have a 2D numpy array by printing the shape of the loaded data. The first axis of this data should be 150 elements long, the second axis should be 4 elements long. What does this mean? (hint: look up the Fisher Iris dataset online, we'll discuss it more next lab)
3. Select 2 **feature** dimensions and create a **scatter plot** the data using matplotlib. pyplot's scatter function.
4. Give the plot a title, and the axes suitable labels. It may be worth looking up the Fisher Iris dataset online to find the names of 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 notebook and ensure it conforms to the style guide as much as possible.

Also go back and update your notebook with suitable markdown cells to annotate the document. Utilising markdown cells to really create a great tool for communication is one of the key benefits of notebooks, so lean into this functionality!