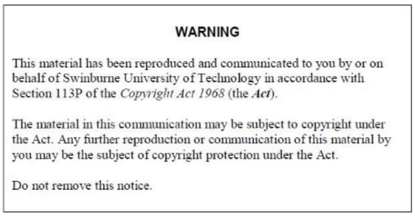
Tutorial & Lab – Week 2



Welcome to **COS80027 – Machine learning**!

Each week you will receive a tutorial & lab sheet to work through in your tutelab class.

You can work through the tasks with your study buddies in the class. You can also ask your tutor for help if you are stuck.

Note: 🧑‍🏫 stands for tutorial exercises and 🖥 stands for lab exercises.

Warning: Whenever you log out, all files you've created and saved in the lab computer will be **ERASED**!!! Therefore, so do please remember to save and copy all your files from the lab computer to your own USB, hard drive or cloud drive before you log out.

🧑‍🏫 Task 1 – Review Some Basic Concepts in Python Programming

- What are the characteristics of Python compared to other programming languages that you're aware of? How do you understand "*Python is a dynamically typed and strongly typed programming language*"?
- Name the basic data types in Python.
- What is the major difference between `list` (data type) and `tuple` (data type)?
- What's the major difference between a variable in Python and that in the other programming language(s) you are familiar with, if any?
- List different ways of defining a string literal and describe the uniqueness of different ways, if any.
- How to define a function in Python? What is the difference between the way to define a function in Python and that in the other programming language(s) you are familiar with, if any?
- How can we make some input parameter(s) of a function to become optional?
- How to check the class of an object? How to use the attributes and methods of an object in Python, e.g., in the context of a string object `"Knowledge"`?

Task 2 – Working with Vectors, Matrices and Numpy arrays

In this task, you review working with vectors and matrices in python3 and using basic operations such as summation, subtraction, multiplication, division, dot product, and transpose. Try to create vectors and matrices by Numpy arrays.

Step 1: Make the preparation

- Launch the Jupyter Notebook in the lab computer.
- Create a folder named "`ML`" in the default working directory "`C:\Users\xxx`".
- Create a sub-folder named "`tutelab02`" in "`C:\Users\xxx\ML`" and enter this sub-folder under the “Files” tab in the Jupyter Notebook.
- Create (and open) a new notebook file via “`New` → `Python 3`” and rename this file to "`Task2`".

Step 2: Type the following practice codes in your notebook and see the outputs. If you get an error for any of those, try to figure out the reason and solve the issues by discussing it with your classmates and tutor.

```
>>> import numpy as np
>>> V = np.array([3, 1, 8, 24, 6])
>>> RowV = np.array([[3, 1, 8, 24, 6]])
>>> ColV = np.array([[3], [1], [8], [24], [6]])
>>> M = np.array([10, 5, 7, 14, 2, 9, 4, 8, 20, 17, 1, 3, 36, 15,
11]).reshape(3,5)
>>> print(M + V))
>>> print(M – V))
>>> print(M / V))
>>> print(M ∗ V))
>>> print(np.multiply(M , V))
>>> np.dot(M,V)
```

Step 3: Now repeat these operations for M and ColV and for M and RowV and interpret the results.

Step 4: Practice using the transpose command using the following codes:

```
>>> print(M.T)
>>> print(M + M)
>>> print(M – M)
>>> print(M ∗ M)
>>> print(M / M)
>>> print(np.multiply(M,M)))
>>> print(np.multiply(M,M.T)))
>>> print(np.dot(M,M)))
>>> print(np.dot(M,M.T)))
```

Task 3 – Define and Call a Function in Python (Interactive Mode)

Step 1: Create (and open) a new notebook file via "`New → Python 3`" and rename this file to "`Task3`".

Step 2: Define a function `DaysToNextBirthday(birthmonth, birthday)` in a code cell, where, `birthmonth, and birthday` must be provided as an arguments of the function.

This function will calculate the remained days to a person's next birthday `and print it as a result.` Furthermore, it will display the result in months and days. For the time being, let's assume no existence of leap years and there are 365 days in a year.

You should handle the invalid values for birthmonth and birthday, an error message indicating the input value is invalid will be displayed, and the program will terminate accordingly.

Step 3: Call the function within the same code cell as the function definition via the following statements:

```
DaysToNextBirthday (11,24)
DaysToNextBirthday (13,1)
DaysToNextBirthday (5,32)
```

and observe the outputs.

Task 4 – Class and Object

Many of you may be more familiar with how to write code in the OOP paradigm. If this is your case, you are encouraged to write a class named `Person` which contains a method named `DaysToNextBirthday` to do the same thing as `DaysToNextBirthday` (defined in Tasks 3) does. You are expected to run the following two statements to obtain the same output as those of Tasks 2 and 3:

```
p = Person(7,15)
p. DaysToNextBirthday()
```

If you are unfamiliar with the OOP paradigm, you may choose to follow Week 2's lecture slides to try to work out this task or just skip over it.

Task 5 – Python vs. Non-Python Programming Languages (Optional)

If you are familiar with some programming language other than Python, please work out Tasks 3 and 4 by using that programming language. Then, you may thoroughly explore the difference between the code in Python and the code in the programming language you are familiar with.

If you are very new to programming, you may just skip over this task.

Task 6 – Probability distribution and statistics

In this task, you are asked to sample a normal distribution and generate histogram each time. The steps for writing the code:

- sample 15 random numbers (float16) from a normal distribution with **mean 4** and standard deviation 2.
- generate a histogram with 50 bins from the above generated numbers and plot the generated histogram.
- repeat the above steps using increased sampling size (100, 150, 200, …), until the plotted histogram looks like a bell curve
- calculate the mean and standard deviation of the finally sampled numbers and represent their differences from 4 and 2, respectively.
- you can optionally consider sampling from another distribution such as binomial and chi-square distributions to repeat the above steps.

You may to refer to the following links to find out how to use the relevant functions in NumPy to complete this task:

- https://docs.scipy.org/doc/numpy-1.16.1/reference/routines.random.html
- https://docs.scipy.org/doc/numpy-1.16.1/reference/routines.statistics.html

You may discuss with your study buddies or contact your tutor for help should you encounter any problems.

--END--

REFERENCES

- https://learnxinyminutes.com/docs/python/
- https://static.thenounproject.com/png/108284-200.png
- https://image.flaticon.com/icons/png/512/69/69045.png