Exercise Session 1

CS 233 - Introduction to Machine Learning

February 19, 2019

1 Introduction

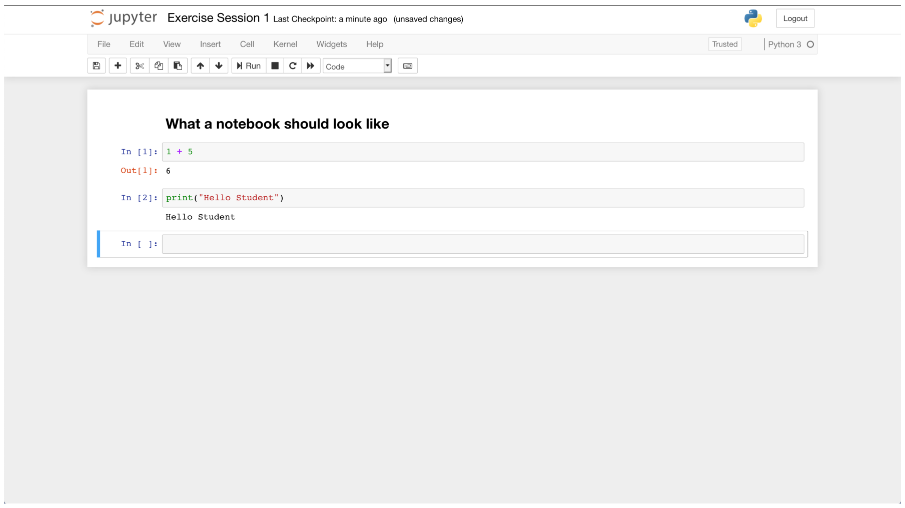
Welcome to the CS233 - Introduction to Machine Learning class. In this course, we will be using Python as our main programming language. For operations in machine learning applications, it is very beneficial to use NumPy arrays together with vectorized commands, instead of explicit *for* loops. This is more computationally efficient. Therefore, we will be using the NumPy library as well. Finally, we will also be using the Jupyter notebook environment.

2 Getting Started

2.1 Installing Jupyter Notebook

Jupyter notebook is a very nice web environment made up of ”cells” that you can run separately using Python, a powerful high-level, object-oriented programming language that is widely used for tasks in machine learning and data science. In order to install it, you can follow the simple instructions in the following link: `https://jupyter.org/install.html`. We strongly recommend you to install Jupyter using Anaconda.

If you have any installation issues, do not hesitate to call one of the assistants. After installing it and running the command *jupyter notebook* in your command line, you can create a notebook by clicking on “New” and then clicking on “Python 3”. A new window will then appear. The figure below shows you what a Jupyter notebook looks like (after running a couple of cells):



What a notebook should look like

```
In [1]: 1 + 5
Out[1]: 6

In [2]: print("Hello Student")
        Hello Student

In [ ]:
```

Figure 1: Example of Jupyter notebook using Python

In the figure above, you notice that there are different “cells” which brings us back to the definition earlier when we stated that the Jupyter notebook is a collection of “cells”. Every cell can be executed separately, making it very efficient if you have a very long piece of code. In order to write something other than Python code, such as the title “What a notebook should look like” in the image above, simply change the cell from “Code” to “Markdown” or to “Heading” in the top bar.

If you wish to exit the notebook, simply click on “Logout” in the top right of your screen or click Control+C in your command line.

2.2 Useful Commands

We give a short overview over some commands that prove useful for writing vectorized code. In order to use NumPy, you need to execute a cell containing the following command in your Jupyter notebook: `import numpy as np`. This allows you to then run the NumPy commands using the `np` shortcut.

- `a * b, a / b`: element-wise multiplication and division of matrices (arrays) **a** and **b**
- `a.dot(b)`: matrix-multiplication of two matrices **a** and **b**
- `a.max(0)`: find the maximum element for each column of matrix **a** (note that NumPy uses zero-based indices like Java)
- `a.max(1)`: find the maximum element for each row of matrix **a**
- `np.means(a), np.std(a)`: compute the mean and standard deviation of all entries of **a**
- `a.shape`: return the array dimensions of **a**

- `np.sum(a, axis=k)`: sum the elements of matrix **a** along dimension **k**

Make sure to test these commands by yourself in the notebook to practice and get an intuition. They will be extremely useful throughout the course. If you are interested in learning more, a more compelte tutorial is available at
`https://engineering.ucsb.edu/~shell/che210d/numpy.pdf`

When you are ready, have a look at the exercise session by running the corresponding notebook.

3 Probability Exercises

3.1 Spam filter 1

You have just installed a system to detect junk email. The system can identify junk messages in 99% of cases. Nevertheless, the system says that a message is junk when it isn't in 2% of cases. Given that 10% of received emails are junk:

(a) What is the probability that a message is not junk?

(b) Given that a message is junk, what is the probability that this message is detected as junk ?

(c) Given that a message is junk, what is the probability that this message is wrongly detected as not junk?

(d) What is the probability that a message is detected as junk?

(e) (Challenging question) What is the probability that a message truly is junk when the system says it is?

3.2 Spam filter 2

Assume you have collected a dataset of emails out of which 432 are considered to be **spam** and 2170 to be **legit** emails. Furthermore, you have selected three words, namely *exercise*, *fun* and *viagra* and counted how many emails contained each of them:

word	appearances in spam	appearances in legit
exercise	6	39
fun	59	9
viagra	39	19

You intend to use a naive Bayes classifier to classify new emails. Assume that you have received an email which contains the following words: $\{exercise, fun\}$. Do you classify it as **spam** or **legit**?