SENG 265 Term Portfolio Project

Daniel Oyeniyi

1. What is the core functionality of Jupyter notebooks

Jupyter notebook is a tool for advanced documentation. You can use Jupyter notebook to present code, visual data, and explantions. It has it's own markup language for customizing and adding different elements to your notebook. Jupyter notebook is popular amongst researchers and scientist for presenting their data and reasearch.

The following is an example of a Jupyter notebook on Machine learning by Yuan Zhao: Machine learning Exercise 1 - Linear Regression

2. Jupyter notebook markdown

Jupyter notebook markdown lets you customize the visual elements of your notebook. Below are examples of what you can do with Jupyter notebook markdown.

Headings: You can use # to add a heading of different size

Heading 1

Heading 5

Indents: You can use > to add indents

> one indent
>
> > two indents
> >
> > > and so on

Bullets: You can use - to add bullet points

- point 1
  - point 1.1
  - point 1.2
- point 2
  - point 2.1
  - point 2.2

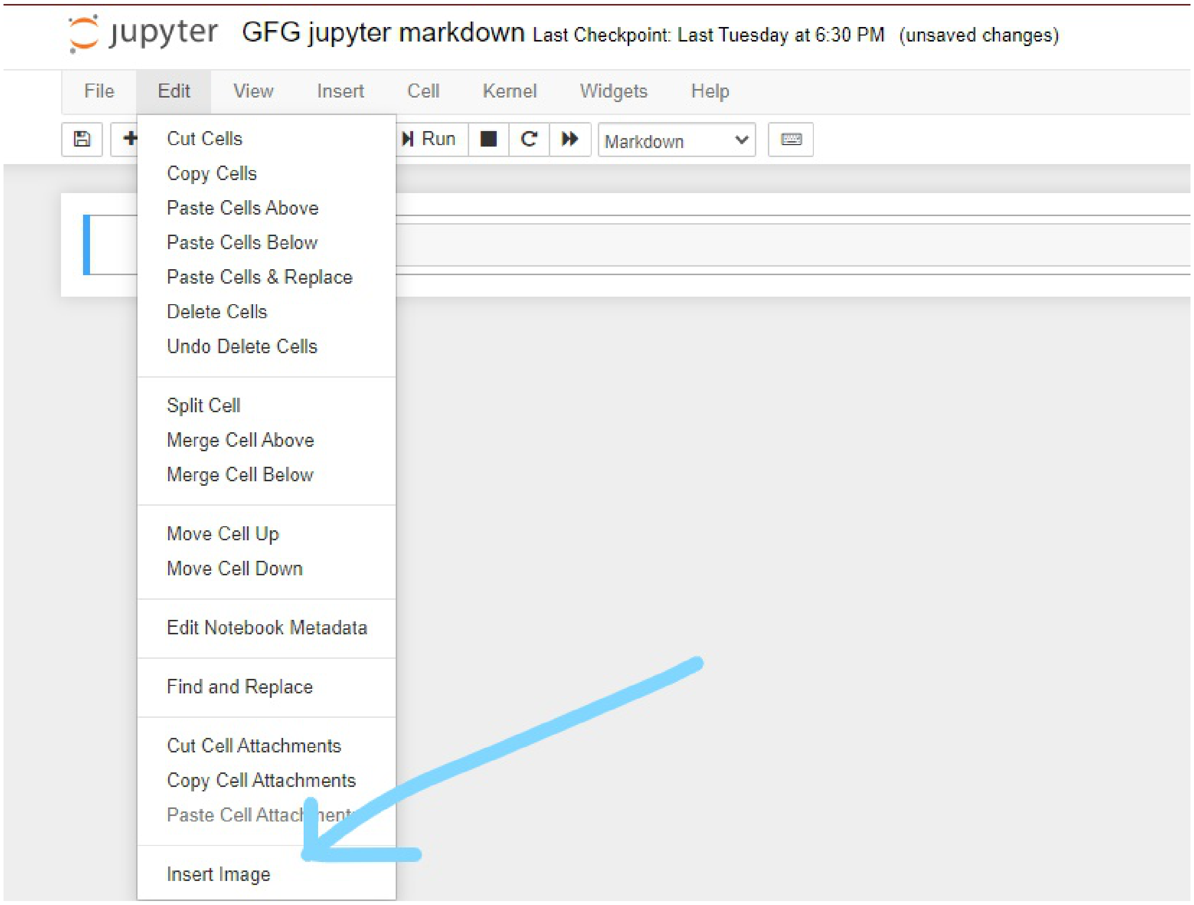
Links: There are several ways to add links. Here is one way.

[Example text](your link)

Example text is what the reader will see as the hyperlink, your link is the actuall link you want the reader to go to.

Here's the result: Example text

Images: Jupyter notebook lets you insert from the header



There are several more options ranging from bolded text to tables.

Information primarily obtained from geeksforgeeks: Jupyter notebook markdown

3. Typestting mathematical formulas using LaTeX Markdown

You can display mathematical formulas in Jupyter notebook by using latex. Surround your expression by $ as you would with quotations.

For example if we type $y=a^2+b^2+c^2\$ then we get:

$y = a^2 + b^2 + c^2$

Aside: In order to display special characters we can use \ or \\ before the character.

4. History of Unix

Unix was the first portable operating system. It was developed by AT&T Corporation's Bell Laboratories around 1970. The team leads were Ken Thompson an Dennis Ritchie. Initially Unix was good at text processing and filesharing but had issues with security and perfomance. Currently Unix is used by Android, MacOS, Microsoft Windows, etc...

Information obtained from: Britannica and SENG 265 Lecture Slides by Hausi Müller

5. Functionality of Bash

Bash, also known as Bourne Again Shell, is a scripting language but also a program. In simple terms Bash is a command interpreter. Bash allows users to manipulate filesystems through the use of bash commands. Bash is especiialy useful when it comes to file i/o through the use of pipes and filters.

Here are some Bash commands:

ls - list directory contents

echo - print text to terminal window

touch - creates or updates a file

> - redirects the output from the preceeding command

| - Pipe the output of one command as the input of the other

Information obtained from: Educative

6. Version control and configuratin managements in software development

Version control is neccessary for ensuring a good workspace environment when working on a project. There needs to be logs of past versions and changes. This allows developers to access previous versions when they need to. Version Control is also important when multiple people are working on the same project. Members need to be working on simillar versions of the software to prevent conflicting code.

7. Functionalities of Git, Github, and Gitlab

Git, Github, and Gitlab are all version control systems. These version control systems provide different way for people to work on the same project. It lets them access the same repository or filesystem and work with simillar files localy. These software also allow users to easily deal with conflicts from merging different versions.

Here are some git commands:

add - adds files to the local repository status - shows the current status of the local repository compared to the main repository, as well as changes staged for commit

commit - change the files in the local repository with the added files, store a log and message of state of the repository

push - replace the main repository with the current version of your local repository

And many more!

8. My Background with C

Prior to SENG 265 I had no experience with the C programming language. However I am now familiar with some rudimentary concepts related to C. I have used it for assignments related to file I/O so I have some knowledge in that aspect. C is currently mainly used in systems programming. Python, Java, Swift and several other languages are written in C. Aside from systems programming C can be used for a plethora of other things. C is popular because it is a middle-level language that is one of the fastest progamming languages.

9. My Background with Python 3

I have been using python since my first year in university. It is by far my favourite language for its simplicity and versatility. Outside of assignments it is my preffered language for side projects. Python is one of the most popular programming languages used today. It is used for data science, machine learning, web development, applications, etc... Its uses are seemingly endless.

One of my python projects: Battlesnake Spring 2023

10. Most popular programming languages

According to Northeastern University these are the top programming languages in 2020:

1. Python

Python is easy to use and easy to read because of its enforced syntax. It is used in a wide variety of applications.

2. JavaScript

When it comes to web developments javascript provides developers with different tools. Devopers have access to several different frameworks that can help aid the web development process.

3. Java

Java is found in a large number of devices. It is used primarily in client-server applications. It has high compatability when run on other platforms.

11. My future with C and Python

We are learning both C and Python in SENG 265 so it would be nice to use it more in the future. However I am open to learning other languages since there is no guarantee that I will be working with C or Python in the future.

12. Fundamental differences between Python and C

Python is an object oriented language while C uses functional programming. In python everything is an object the same can't be said for C. C does have structs which can serve a simmilar purpose but at its core C is not an object oriented language. Also Python is interpreted at runtime whereas C is compiled. When we run python code it is executed line by line, this is why variables in python can be dynamic and change types. For C it is compiled use a C compiler that creates an executable file when run. Since C programms are compiled their variables are static and cannot change type. There are also syntatical differences between the two. The most notable syntax difference is the lack of semicolons and the use of indentation in python when compared to C. There are several other differences between the two from memory managment to logical operators.

13. Assignment 1

Getting started in assignment 1 was a bit hard since I had no experience with C. I was also using vi(m) so that added to the process. However the farther I got into the assignment the smoother the process was. Once you have some bearing on some of the C fundamentals and what tools you have access to than C doesn't seem so bad.

14. Assignment 2

Assignment 2 was a lot easier than assignment 1 since I was fimilliar with python and I was used to using vi(m). The main task was understanding how to use the pandas to manipulate data and how to use matplotlib to create graphs.

15. Epiphanies the from first part of SENG 265

The biggest epiphanie by was when we discussed interview questions in class. I learned that answering the question isn't always the goal. The most important part is the dialogue between you and the interviewer. They want to know if you take the time to carefully interpret and understand what is being asked of you and how well you communitcate your ideas.

For example: You are given the head pointers of two linked lists where each linked list represents an integer number (i.e., each node is a digit). Add them and return the new linked list.

If you miss the fact that each node is a digit then you are already headed in the wrong direction. If you catch that you still need to ask for more information. Is this a singly or doubly linked list? Is the head the least significant digit or most significant digit?

Upon further discussing with a colleague they showed told me about the Mars Climate Orbiter. A situation where an assumption and misscommunication led to the loss of $193.1 million. Mars Climate Orbiter

16. How useful was SENG 265 for building my resume

Overall I think the course was a big help. SENG 265 covered a lot of differnt topics that are relevant in the current job market. I now have more knowledge of things like object oriented programing, modles, bash, fstring, etc... Now if I could get a response from one of the co-op jobs that would be nice, but I don't have 9 years of experience yet.

17 Which skills from SENG 265 are the most valuable to me

Learning bash was by far the most usefull skill, at least for this semester. It allowed me to see the beauty of how file systems are managed. Another skill was learning regular expressions. I was suprised to see that they are standard amongst diffrent languages. So once you learn them it's easy to apply them to other languages.

18 Python Type Hints

In my understanding the main benefit of type hints is clarity. Since python allows you dynamically change the type of a variable, unlike c. I can understand how it could be confusing to track the different variable types. Type hints make it easier to track exactly what the variable type is.

19 Python list slicing and comprehension

List slicing and comprehensions are easy way to copy lists in python.

Slicing

Allows you to copy a section of a list, for example:

In [6]:

```
x: list = [1, 2, 3, 4, 5]
y: list = x[2:4]
print(y)
```

```
[3, 4]
```

Comprehension

Allows you to copy only the desired items in a list, for example:

In [7]:

```
x: list = [2,8,3,5,10,1]
y: list = [num for num in x if num % 2 == 0]
print(y)
```

```
[2, 8, 10]
```

Some infromation obtained from W3schools

20 Objects

The purpose of creating objects makes it easier to manage certain elements of programing. Objects allows us to have the same class of items and methods but with different values. If I made a ball class, each ball object created would have it's own volume, air pressure, weight, and color. However each ball would still have an inflate method and a deflate method. This allows us to manage sever different instances of data while still maintaining the same functionality.

21 Deep Copy vs Shallow Copy

Shallow copy will copy the reference to the object whereas deep copy will copy each element of the object. So if you only perform a shallow copy then modifications to the reference object will still change both copies. You won't run into this problem with deep copy but it takes longer.

Shallow Copy

In [22]:

```
import copy
original: list = [1, 2, 3, 4, 5, 6, 7]
shallow: list = original # creates a shallow copy
print(original)
print(shallow)

original[2] = 1000
original.append(44)
print(shallow)
print(original)
```

```
[1, 2, 3, 4, 5, 6, 7]
[1, 2, 3, 4, 5, 6, 7]
[1, 2, 1000, 4, 5, 6, 7, 44]
[1, 2, 1000, 4, 5, 6, 7, 44]
```

Some information obtained from geeksforgeeks

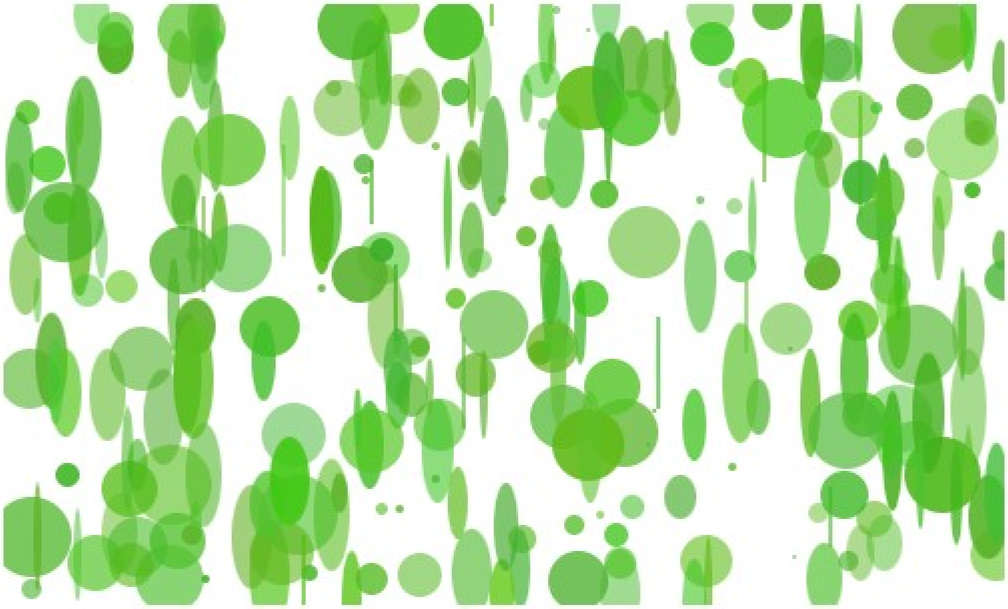
22 Assignment 3

Assignment 3 was somewhat challenging. We had to manipulate linked lists in C so getting everything set up was difficult. However after the initial setup was done it was fairly simple to complete the assignment. Overall it was a lot easier than assignment 1 since I had more experience with C.

23 Assignment 4

Like Assignment 3, Assignment 4 took a while to setup. Assignment 4 had 3 parts but the bulk of the work was in the first part. In the first part you had to set up most of the classes that you would need for parts 2 and 3. Part 2 was a bit difficult as well but most of the setup was already completed in part 1. Part 3 however was just implementing parts 1 and 2 so it didn't take long to finish. By far part 3 was the most fun since you got to play around with the finished product.

Here's one of the outputs from Assignment 4, not to shabby, if I do say so myself! I didn't share it with any family members but at least the people reading this portfolio will get to see it.



24 Epiphanies from the second part of SENG 265

Before the second part of the course I didn't understand the value of Regular Expression but I can now appreciate a laguage format that is widly used regardless of the programing language.

I also understand dunder variables a bit more. I now have a better idea of what this code snippet means

In [23]:

```
if __name__ == "__main__":
    print("hello world")
```

```
hello world
```

I also learnt the value of docstrings and how we can use pythons doctest module to promote incremental development. Incremental development as a whole is something that I've subconsiously employed before but now I'm more aware of what is is and why it's important.

25 Citations and References

1. https://nbviewer.org/github/jdwittenauer/ipython-notebooks/blob/master/notebooks/ml/ML-Exercise1.ipynb
2. https://www.geeksforgeeks.org/markdown-cell-in-jupyter-notebook/

3. https://www.britannica.com/technology/UNIX
4. https://www.educative.io/blog/bash-shell-command-cheat-sheet
5. https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/
6. https://en.wikipedia.org/wiki/Mars_Climate_Orbiter#:~:text=The%20Mars%20Climate%20Orbite
7. https://www.w3schools.com/python/python_lists_comprehension.asp
8. https://www.geeksforgeeks.org/difference-between-shallow-and-deep-copy-of-a-class/