Assignment L03 Lab ITAI 1378

Houston Community College

Mary Balemba

ITAI 1378: Computer Vision-Artificial Intelligence

Professor Anna Rachapudi

February 8th, 2025

Link to GitHub Repository: https://github.com/MaryBalemba/jupyter-exploration

Journal Summary on Exploring GitHub and Jupyter Notebooks

What I Did

Setting Up GitHub

I started by creating a GitHub account, which involved signing up with my email and setting a secure password. After verification, I explored the GitHub interface, including the dashboard, repositories, pull requests, issues, and actions. This helped me understand the basic functionalities of GitHub.

Next, I created a new repository named "jupyter-exploration." I made it public and initialized it with a README file. This step was important because it served as the foundation for storing and managing my Jupyter Notebook files.

To familiarize myself with the commit process, I edited the README file by adding a brief description of my project. I then committed the changes with a descriptive message, ensuring that my modifications were documented in the repository's history.

Working with Jupyter Notebooks

To begin exploring Jupyter Notebooks, I had two options: using an online environment like Google Colab or installing Jupyter Notebook locally. I chose installing Jupyter Notebook locally because I had it installed already.

I created a new notebook and experimented with both Markdown and code cells. In a Markdown cell, I wrote a simple heading and description to document my work. In a code cell, I executed a basic Python script: print("Hello, World!"). This code successfully displayed "Hello, World!" in the output, confirming that my Jupyter Notebook was functional.

Saving and uploading to GitHub

After completing my notebook, I saved it and downloaded it as an .ipynb file. To upload it to my GitHub repository, I used GitHub Desktop. I cloned my repository to my local machine, moved

the .ipynb file into the repository folder, and committed the changes. Finally, I pushed the notebook to GitHub, making it accessible through my repository's online interface. For this last step, I am not sure it my commit went through successfully but I was able to see it in my Github.

What I Learned

Understanding GitHub and Version Control

Through this process, I gained hands-on experience with GitHub and version control. I learned how repositories help manage project files, track changes, and collaborate efficiently. The commit process reinforced the importance of documenting modifications with clear messages. Additionally, I understood the significance of branches and commits in version control. While I only worked on the main branch for this exercise, I now see how creating separate branches for feature development can enhance collaboration and prevent conflicts.

Working with Jupyter Notebooks

Jupyter Notebooks provide an interactive coding environment, making it easy to combine text, code, and visualizations. I learned how to use Markdown to document my work effectively and how to execute Python code in a structured manner.

One of the key takeaways was the difference between Markdown and code cells. Markdown is useful for adding explanations, while code cells execute Python commands. Understanding this distinction improves the readability and organization of Jupyter Notebooks.

Uploading Files to GitHub

I learned how to use GitHub Desktop to simplify the file upload process. Instead of manually uploading through the GitHub web interface, cloning a repository locally and using GitHub Desktop provides better version control and flexibility.

Questions or Comments

One challenge I faced was the inability to commit to main and push to origin so it can be uploaded to GitHub. Every other step was successful. I would like to explore advanced GitHub features, such as creating branches and pull requests, to improve my workflow.

Overall, this exercise provided valuable insights into GitHub and Jupyter Notebooks, helping me build foundational skills for future technical projects.