

## L02\_Journal\_Terrance\_Hardy\_ITAI\_2373

### a. What I Did

In this lab, I worked with GitHub, Google Colab, and a Jupyter Notebook to complete a Natural Language Processing (NLP) exercise focused on text preprocessing.

I started by creating a GitHub account and a public repository called Jupyter-Exploration, initializing it with a “README” file. I committed the first changes to familiarize myself with version control.

Next, I uploaded the provided notebook:

LabBasicNLPPreprocessingTechniques\_Enhanced\_Mod02.ipynb to Google Colab and executed all the code cells in order. The notebook included practical implementations of tokenization, stop word removal, stemming, and lemmatization using both NLTK and SpaCy libraries. I also answered the reflection questions embedded in the notebook.

Once I completed the exercises, I downloaded the notebook from Colab to my Mac and uploaded it to GitHub. I verified that the notebook was correctly uploaded and visible in my repository alongside the “README file”.

### b. What I Learned

This lab helped me understand both the technical workflow and the concepts behind NLP preprocessing.

From a workflow perspective, I now understand how:

- Google Colab acts as a cloud-based coding environment for running notebooks
- My local machine stores files downloaded from Colab
- GitHub serves as a cloud repository for storing, versioning, and sharing files

From a conceptual perspective, I observed the effects of preprocessing on raw text data. The notebook showed:

- Tokenization: breaking text into smaller units like words or sentences
- Stop word removal: eliminating common words that carry little semantic meaning
- Stemming: reducing words to their root forms, i.e., “running - run”
- Lemmatization: reducing words to their dictionary form, i.e., “better - good”

By comparing NLTK and SpaCy, I could see different approaches to preprocessing and how each library represents tokens and normalizes text. I also noticed that preprocessing decisions can significantly change the structure and content of the text data, which is crucial for downstream NLP tasks.

Although I mostly followed pre-written code, this lab gave me a practical view of how preprocessing is applied in real-world scenarios and how to handle the workflow from code execution to GitHub submission.

c. Questions or Comments

- Future instructions could benefit from clearer, more concise wording.
- A diagram illustrating the connection between Colab, local files, and GitHub would make the workflow easier to understand.
- I would like to explore writing preprocessing code from scratch in future labs to better internalize the concepts.