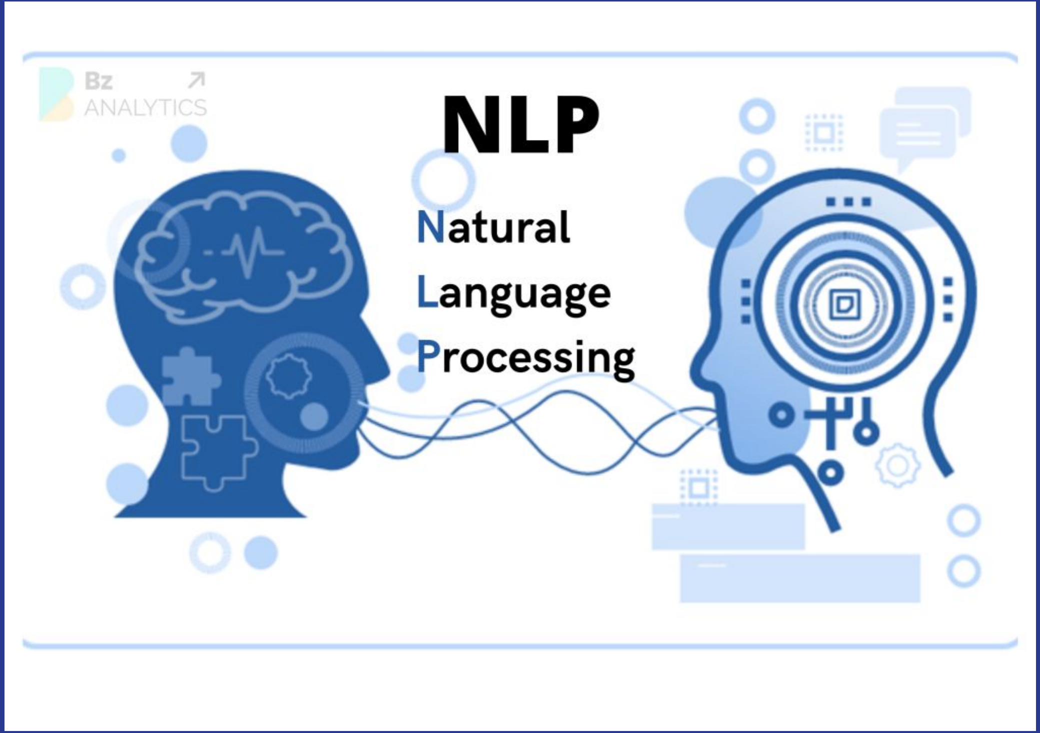
Final Portfolio- Natural Language Processing

Oluchi Obinna



Helpful Articles & Links-

1. **Understanding SHRDLU: A Pioneering AI in Language and Reasoning**
   Retrieved from: CryptLabs
2. **SHRDLU: An Early Natural Language Understanding Computer Program**
   Retrieved from: IndiaAI
3. **Introduction to Pandas: Data Science Tutorial**
   Retrieved from: DataCamp
4. **Introduction to Pandas in Python**
   Retrieved from: GeeksforGeeks
5. **Text Cleaning Methods in NLP**
   Analytics Vidhya. (2024, November 13). Retrieved from: Analytics Vidhya
6. **Introduction to NLTK: Tokenization, Stemming, Lemmatization, POS Tagging**
   GeeksforGeeks. (2023, April 15). Retrieved from: GeeksforGeeks
7. **Stemming vs Lemmatization in NLP: Must-Know Differences**
   Analytics Vidhya. (2025, January 10). Retrieved from: Analytics Vidhya

1. **Quick Introduction to Bag of Words (BoW), TF-IDF**
   Retrieved from: Analytics Vidhya
2. **Guide on Word Embeddings in NLP**
   Retrieved from: Turing
3. **What Are Word Embedding? Word2Vec in NLP**
   Retrieved from: Digitate
4. **Word Classes and Part of Speech Tagging in NLP**
   Retrieved from: Scaler
5. **What Is Parts of Speech (POS) Tagging?**
   Retrieved from: Medium
6. **Named Entity Recognition (NER) Overview**
   Retrieved from: IBM
7. **CRF-NER Stanford Software**
   Retrieved from: Stanford
8. **A Beginner's Introduction to Named Entity Recognition (NER)**
   Retrieved from: Analytics Vidhya

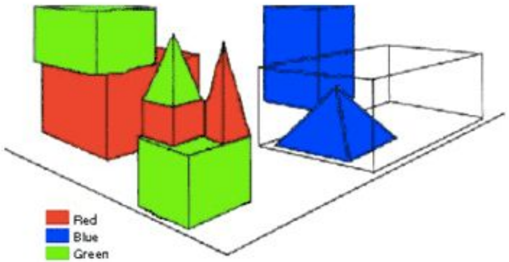
Module 1- Introduction to NLP – Understanding SHRDLU

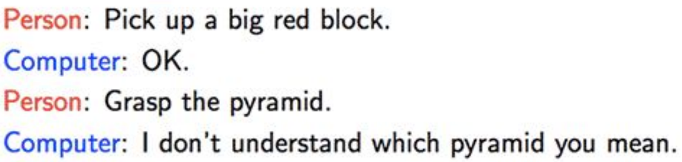
Activities:
In this module, I studied SHRDLU, one of the earliest natural language processing (NLP) programs developed at MIT by Terry Winograd (1968–1970), to understand how computers began processing human language.

Results:
From this module, I learned how SHRDLU demonstrated fundamental concepts of natural language understanding:

– **Program overview**: SHRDLU allowed users to give text-based commands in everyday English to manipulate objects in a virtual "blocks world" of simple shapes (blocks, cones, balls).
– **Vocabulary & syntax**: Used a small vocabulary (~50 words) consisting of nouns (block, cone), verbs (move, place on), and adjectives (big, blue) to interpret user inputs and generate responses.
– **Language understanding**: Processed complex commands like “Put a big red block on top of the green cube” and answered questions about the block world’s state, showcasing syntax, semantics, and pragmatics handling.
– **Reasoning ability**: Simulated human-like reasoning by understanding object relationships, following instructions, and remembering prior interactions.
– **Legacy and influence**: SHRDLU laid the groundwork for modern NLP applications such as chatbots, virtual assistants (like ChatGPT), Google Translate, and AI-powered customer service systems.

What I Learned:
I gained foundational knowledge of how early AI systems began to understand and process natural language, and how these concepts have evolved into today’s advanced NLP applications.





Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I don't understand which pyramid you mean.

Module 2- IBM Debater Technical Report

Activities:
I analyzed the IBM Debater Presentation, which describes how IBM Project Debater uses AI to process data, build arguments, and rebut opposing views. I also completed a lab on Python data structures like lists, tuples, sets, and dictionaries.

Results:
From the IBM Debater, I learned:
– **Data Ingestion**: The AI analyzes large text and uses NLP to extract key info.
– **Argument Construction**: It organizes facts to build strong arguments.
– **Rebuttal Generation**: It responds to opponents' arguments in real-time.
– **Speech Delivery**: It presents arguments with a confident, synthetic voice.
– **Human-AI Collaboration**: IBM Debater aids human decision-making, not replacing it.

What I Learned:
I gained insights into how AI can build arguments and how Python data structures are used to manage information in coding.

From the Python Lab, I learned about:
– **Lists**: Ordered, mutable, can have duplicates.
– **Tuples**: Ordered, immutable, can have duplicates.
– **Sets**: Unordered, no duplicates, mutable.

– **Dictionaries**: Unordered, no duplicates, mutable.

Module 3- The Pandas Library and Its Uses in Data Science

Activities:
I learned about the Pandas library, a Python tool that helps manipulate, clean, and analyze data. I also completed a lab on the steps in a data analysis project.

Results:
From studying Pandas, I learned:
– **DataFrames & Series**: A Series is a one-dimensional array, and a DataFrame is a two-dimensional table, both used for storing and manipulating data.
– **Data Cleaning**: Pandas can handle missing data, remove duplicates, and filter information.
– **Data Transformation**: It allows easy modification and reorganization of data.
– **Data Analysis**: Pandas offers functions for sorting, grouping, and summarizing data.
– **Data Visualization**: Although Pandas doesn't make graphs, it works well with libraries like Matplotlib and Seaborn for visualization.

What I Learned:
I gained hands-on experience with how Pandas helps clean and organize data, making data analysis much easier. I also learned the steps involved in conducting a thorough data analysis project.

In the Data Analysis Lab, I learned about:
– **Data Background & Objectives**: Understand the problem and context of the data.
– **Data Description**: Get familiar with the dataset's columns.
– **Exploratory Data Analysis (EDA)**: Includes data exploration, cleaning, and visualization.
– **Conclusion**: Summarizing findings for presentation.

Module 4- Data Visualization with Matplotlib and Seaborn

Activities:

I explored how data visualization tools like Matplotlib and Seaborn are used to create visual representations of data. I also worked on a lab where I applied these tools to visualize a dataset.

Results:

From studying Matplotlib and Seaborn, I learned:

– **Matplotlib**: A library used for creating static, animated, and interactive visualizations in Python.

– **Seaborn**: Built on top of Matplotlib, Seaborn is used for creating more attractive and informative statistical graphics.

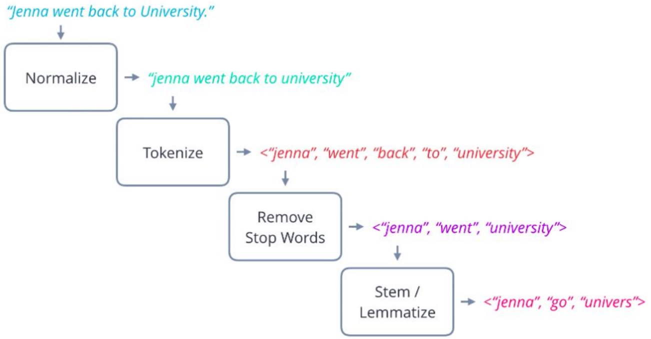
– **Types of Plots**: Both libraries allow you to create different types of plots, such as line charts, bar charts, scatter plots, and histograms.

– **Customizations**: I learned how to customize plots by changing labels, titles, and colors to enhance the visual appeal and clarity of the data.

What I Learned:

I gained experience in using data visualization tools to make data easier to understand and interpret. Creating clear and meaningful visuals helps to convey the story behind the data.

In the lab, I worked on creating plots like bar charts and histograms to represent a dataset. I also learned how to analyze trends and patterns through visualization.



"Jenna went back to University."

Normalize → "jenna went back to university"

Tokenize → <"jenna", "went", "back", "to", "university">

Remove Stop Words → <"jenna", "went", "university">

Stem / Lemmatize → <"jenna", "go", "univers">

Module 5- Lab 05: Tokenization

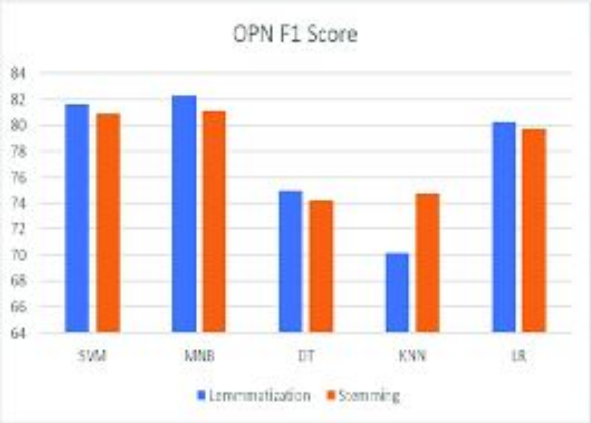
Activities:
In this module, I built a sentiment analysis model using Python, focusing on classifying text data as positive or negative.

Results:
From the module activities, I learned the complete process of developing a basic sentiment classifier:

– **Data cleaning**: Removed special characters, converted text to lowercase, and eliminated stopwords to prepare the dataset.
– **Tokenization and stemming**: Broke sentences into words and applied stemming to simplify word forms.
– **Feature extraction**: Used `CountVectorizer` to convert text into numerical feature vectors.
– **Training a model**: Applied the Naive Bayes classifier to train the sentiment analysis model.
– **Model evaluation**: Tested the model on sample inputs to measure accuracy and detect bias.
– **Prediction**: Classified new text samples to determine their sentiment (positive or negative).

What I Learned:

I gained hands-on experience in preparing text data, extracting features, training a machine learning model, and evaluating its performance, deepening my understanding of how AI systems can interpret human opinions.



Module 6- Bag of Words and TF-IDF Models

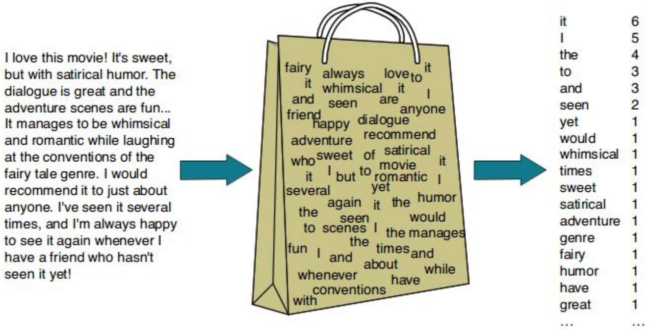
Activities:
In this module, I created a basic product recommendation system using user preferences and item similarity.

Results:
From the module activities, I learned how to design and implement a recommendation engine:

– **Data collection**: Gathered a sample dataset of user preferences for products.
– **Similarity calculation**: Used cosine similarity to compare products based on user ratings.
– **Recommendation generation**: Built a function to suggest similar products to users based on their previous interactions.
– **Matrix manipulation**: Applied NumPy and pandas to manage and compute similarity matrices.
– **Evaluation**: Tested recommendations to verify relevance and accuracy for various user profiles.

What I Learned:

I developed a foundational understanding of how recommendation systems personalize product suggestions using similarity measures and how AI enhances customer experience in e-commerce.



I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Module 7-Sentiment Analysis

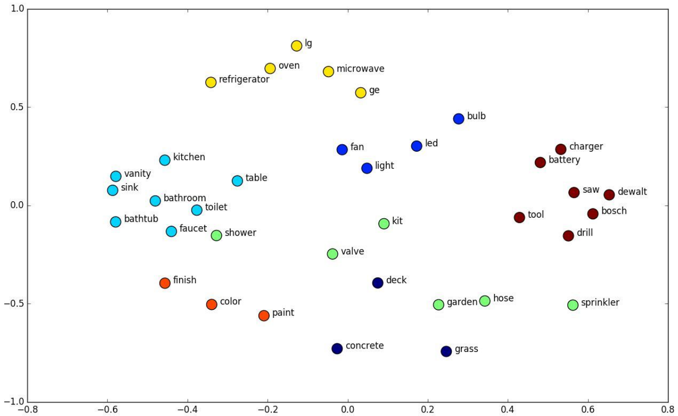
Activities:
In this module, I built a sentiment analysis system to classify text as positive, negative, or neutral.

Results:
From the module activities, I learned how to process and analyze text data for sentiment classification:

– **Data preprocessing**: Cleaned text by removing punctuation, converting to lowercase, and tokenizing words.
– **Stopword removal**: Eliminated common words (e.g., “the,” “is”) to reduce noise in the data.
– **Feature extraction**: Converted text into numerical features using Bag-of-Words and TF-IDF methods.
– **Model training**: Trained a classifier on labeled sentiment data.
– **Prediction and evaluation**: Used accuracy, precision, and recall to evaluate model performance on unseen data.
– **Practical application**: Tested the model on real-world samples, such as product reviews and tweets.

What I Learned:

I gained practical experience in text mining, natural language processing, and how AI can interpret human emotions in text to support decision-making and market analysis.



Module 8- Text Summarization

Activities:
In this module, I developed an automatic text summarizer to condense long documents into shorter summaries.

Results:
From the module activities, I learned how to extract key information and generate concise summaries:

– **Data preprocessing**: Cleaned and tokenized text into sentences and words.
– **Stopword removal and stemming**: Removed common words and reduced words to their base form to simplify analysis.
– **Sentence scoring**: Calculated word frequencies and assigned scores to sentences based on keyword density.
– **Summary generation**: Selected the top-scoring sentences to create an extractive summary.
– **Evaluation**: Compared generated summaries with original text to assess relevance and coherence.
– **Practical application**: Tested summarizer on news articles and reports to extract main points efficiently.

What I Learned:

I learned how AI can process large volumes of text to highlight important content, which is useful for research, journalism, and information management.

Module 9- Text Classification

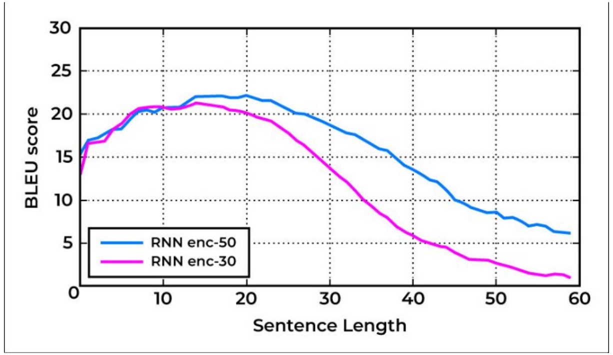
Activities:
In this module, I built a text classifier to categorize documents into predefined labels using machine learning techniques.

Results:
From the module activities, I learned how to create and evaluate a classifier:

– **Data preprocessing**: Cleaned text, removed stopwords, and applied stemming for uniformity.
– **Feature extraction**: Converted text into numerical form using TF-IDF (Term Frequency-Inverse Document Frequency) to reflect word importance.
– **Model training**: Used a Naive Bayes classifier to learn from labeled training data.
– **Prediction**: Classified unseen text based on learned patterns.
– **Evaluation**: Assessed model accuracy using a confusion matrix and calculated performance metrics (precision, recall, F1-score).
– **Practical application**: Applied classifier to categorize movie reviews into positive or negative sentiments.

What I Learned:

I gained practical skills in applying machine learning algorithms to classify text data, an important task for applications like spam detection, sentiment analysis, and document organization.



Module 10- Machine Translation and Seq2Seq Models

Activities:
In this module, I studied how AI translates text between languages and implemented a Seq2Seq model for machine translation.

Results:
From the module activities, I learned about different machine translation methods and built a translation model:

– **RBMT (Rule-Based MT)**: Uses linguistic rules and dictionaries, but struggles with context.
– **SMT (Statistical MT)**: Uses statistical patterns from bilingual text data to improve translation.
– **NMT (Neural MT)**: Uses deep learning models (e.g., LSTMs, Transformers) for fluent, context-aware translations.
– **Data preprocessing**: Tokenized and cleaned German-English sentence pairs.
– **Seq2Seq model**: Built an encoder-decoder network using LSTM layers to translate German sentences into English.
– **Model training**: Trained the model on aligned sentence pairs and optimized translation accuracy.
– **Evaluation**: Tested translations to assess quality and fluency of model output.

What I Learned:

I gained an understanding of how different machine translation methods work and how Seq2Seq models can be used to build an AI system capable of translating languages effectively.

Module 11- Midterm – Building a Chatbot

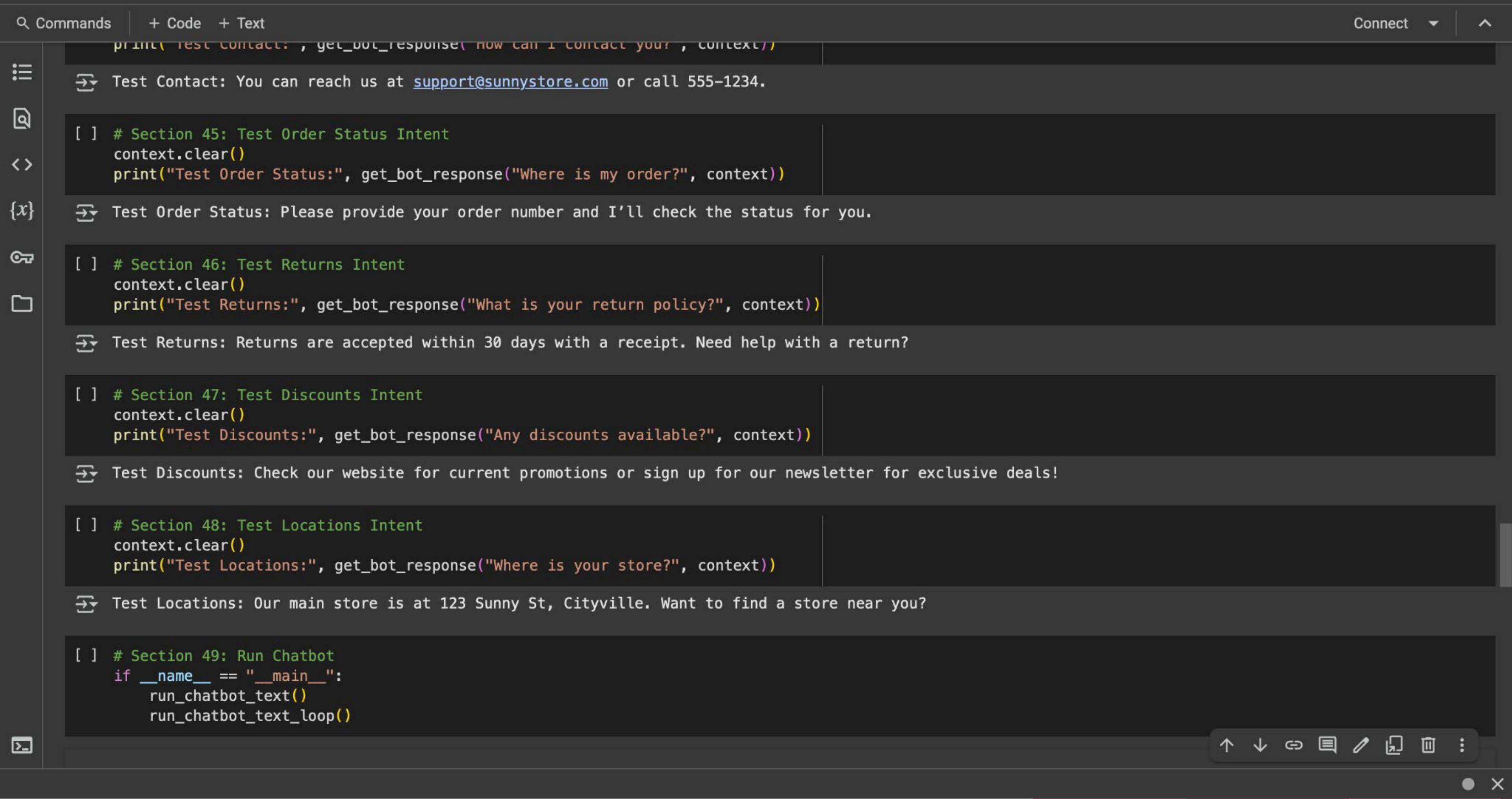
Activities:
In this module, I developed a text-based chatbot using Python and applied natural language processing techniques to handle user inputs.

Results:
From the module activities, I learned key steps in building and refining a chatbot system:

– **Environment setup**: Used `subprocess` to automatically install NLTK and handled errors with `try-except`.
– **Text preprocessing**: Tokenized sentences using `word_tokenize`, removed stopwords, applied stemming (PorterStemmer), and cleaned text (removed non-alphanumeric characters and lowercased).
– **Intent detection**: Defined keyword lists for intents (greetings, products, hours) and matched user input to generate responses.
– **Response system**: Provided specific replies for recognized intents and a fallback response for unrecognized inputs.
– **Conversation management**: Logged user input, detected repeated intents, and implemented a function to clear conversation history for new sessions.
– **Chat loop**: Built a continuous interaction loop where the chatbot responded until the user typed “exit”.
– **Testing**: Verified chatbot responses with sample inputs to ensure accurate intent detection and response generation.

What I Learned:

I learned how to apply NLP methods to build a functional chatbot, manage user interactions, and design a basic conversational AI system.



```
print("Test Contact:", get_bot_response("How can I contact you?", context))
```

```
Test Contact: You can reach us at support@sunnystore.com or call 555-1234.
```

```
[ ] # Section 45: Test Order Status Intent
    context.clear()
    print("Test Order Status:", get_bot_response("Where is my order?", context))
```

```
Test Order Status: Please provide your order number and I’ll check the status for you.
```

```
[ ] # Section 46: Test Returns Intent
    context.clear()
    print("Test Returns:", get_bot_response("What is your return policy?", context))
```

```
Test Returns: Returns are accepted within 30 days with a receipt. Need help with a return?
```

```
[ ] # Section 47: Test Discounts Intent
    context.clear()
    print("Test Discounts:", get_bot_response("Any discounts available?", context))
```

```
Test Discounts: Check our website for current promotions or sign up for our newsletter for exclusive deals!
```

```
[ ] # Section 48: Test Locations Intent
    context.clear()
    print("Test Locations:", get_bot_response("Where is your store?", context))
```

```
Test Locations: Our main store is at 123 Sunny St, Cityville. Want to find a store near you?
```

```
[ ] # Section 49: Run Chatbot
    if __name__ == "__main__":
        run_chatbot_text()
        run_chatbot_text_loop()
```

Final Thoughts

Throughout this course, I gained a solid understanding of Artificial Intelligence and data science concepts. I learned how AI systems like SHRDLU and IBM Debater work, using Natural Language Processing and Machine Learning to engage in tasks like human debate and data analysis. I explored how tools like Pandas simplify data manipulation, making it easier to clean and analyze data. Additionally, I learned about machine learning algorithms and how to evaluate models for tasks like classification and regression.

The course provided hands-on experience with Python libraries and data science techniques, allowing me to apply what I learned through labs and projects. Overall, this course has enhanced my ability to analyze data, understand AI systems, and apply machine learning in real-world situations.