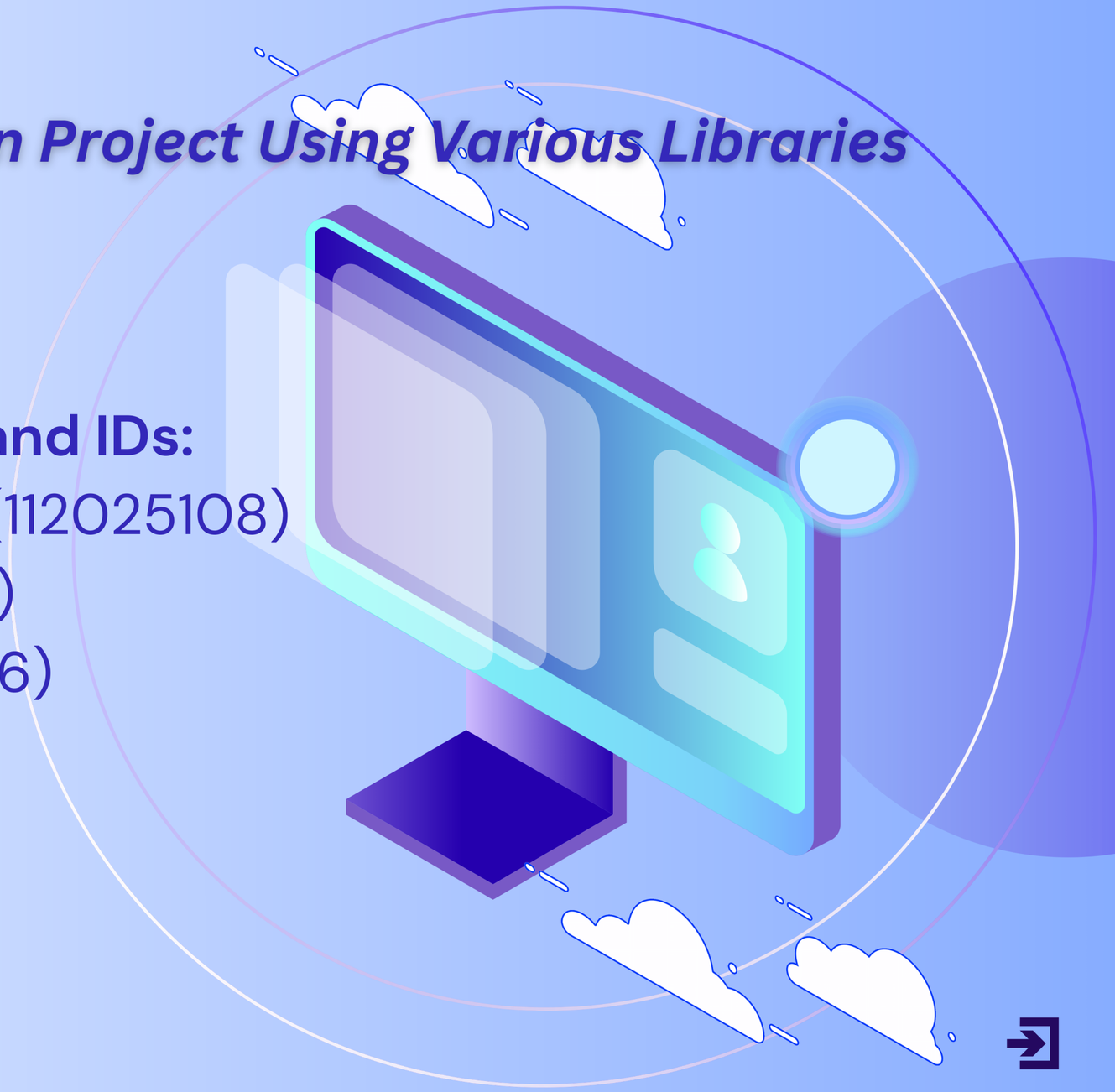
Multi-Functional Python Project Using Various Libraries

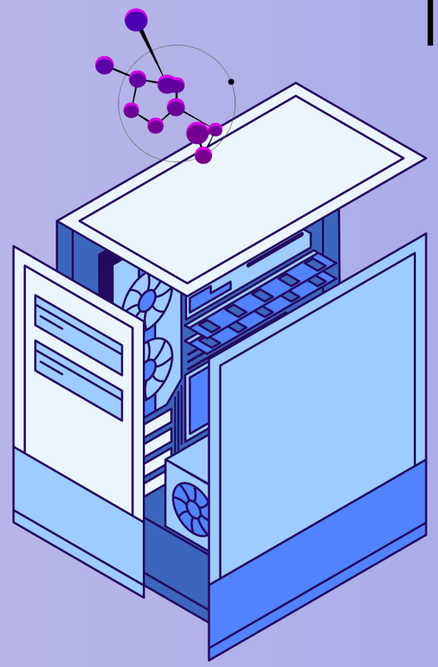


Group 5: Student Names and IDs:

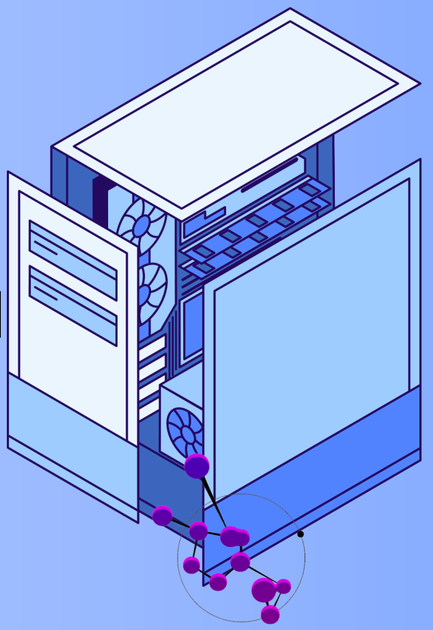
- Sinethemba Ndlangamandla (112025108)
- Mpendulo Maseko (112025110)
- Chandru Ravikumar (112710376)

April 2024

Introduction



This project aims to demonstrate the integration of various Python libraries to automate the process of text data analysis and summarization. By using libraries such as requests, NLTK, spaCy, Pandas, and Matplotlib, the project showcases how different tools can work together to achieve a comprehensive data analysis pipeline.



Libraries Used

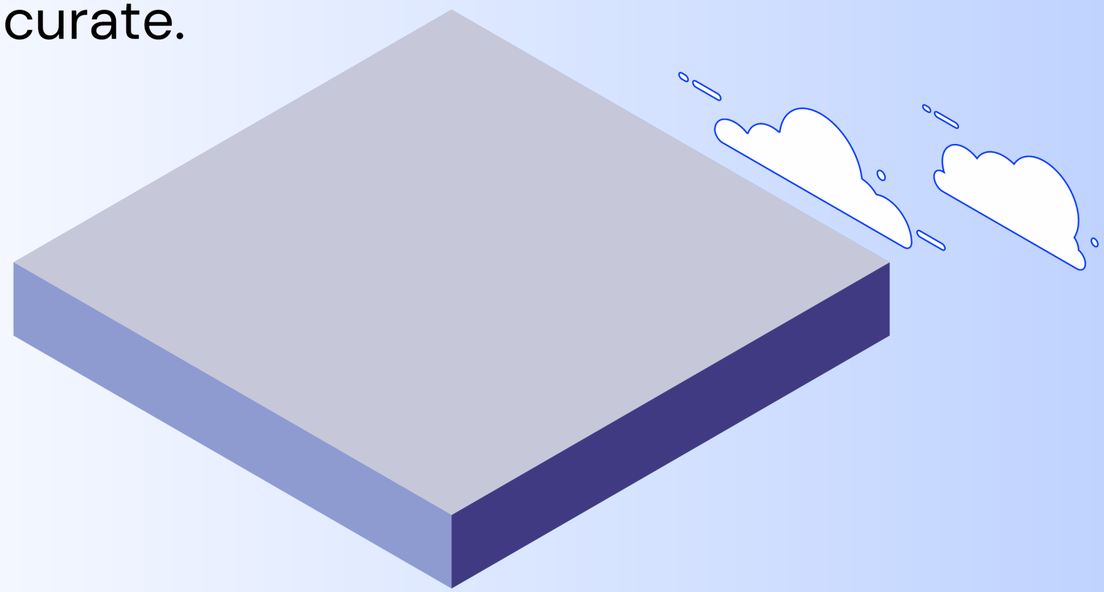
- requests: For fetching text data to send HTTP requests to a specified URL and retrieve the text data.
- NLTK (Natural Language Toolkit): For text preprocessing such as tokenization and removing stopwords.
- spaCy: For natural language processing and summarization for more advanced NLP tasks like tokenization, named entity recognition (NER), and text summarization.
- Pandas: For data manipulation to organize and manipulate data in DataFrame format, making it easier to handle and analyze.
- Matplotlib: For data visualization to create visual representations of the data, such as bar charts
- Scikit-learn: For machine learning for building and evaluating machine learning models, specifically for predicting word frequencies in this project.

Fetching Text Data

URL: The Old Man and the Sea by Ernest Hemingway

- The '*requests*' library allows us to send HTTP requests and handle the responses. In this project, we used requests.get() to retrieve the text data from the provided URL.

- By automating this process, we can ensure that the data is always up-to-date and accurate.



Text Preprocessing

- Library used: NLTK
- Use NLTK to remove common stopwords (e.g., "the," "is," "and").
- Tokenization: Tokenize the text into individual words..
- Filter Out Punctuation: Remove punctuation marks to clean the text further to ensure accurate analysis.

Word Frequency Calculation

- Identified the top 10 most common words. to understand the main themes and topics in the text.

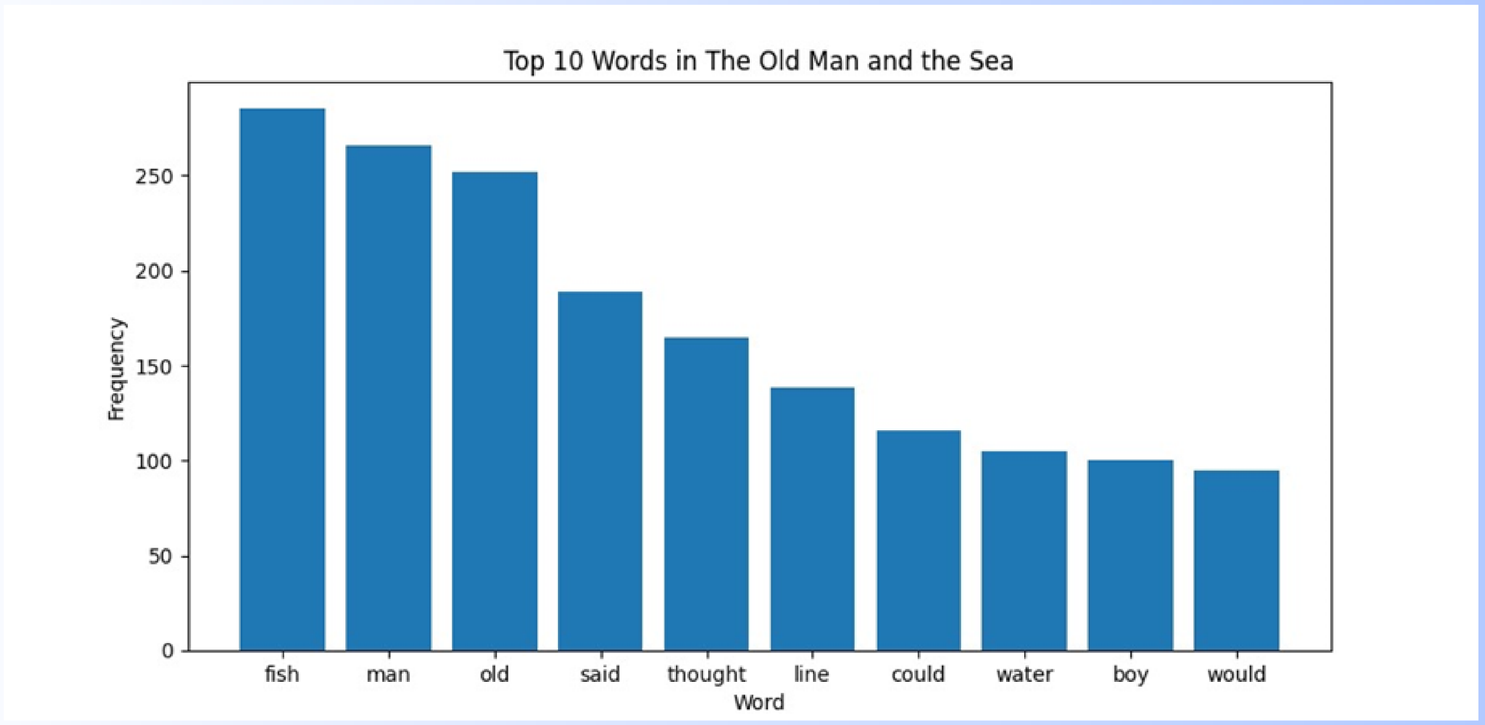
- Calculated the frequency of each word in the filtered text using collections.Counter to count the occurrences of each word in the text.

- collections.Counter: provides an easy way to count the occurrences of each word in the text. By using Counter.most_common(), we can quickly identify the top words.

Data Visualization

- Library: Matplotlib

- Matplotlib is used to create a bar chart that visualizes the frequency of the most common words in the text.



Natural Language Processing with spaCy

- Library: spaCy
- Tokenization: Tokenized text into sentences.
- Sentence Scoring: Calculated sentence scores based on word frequency
- Summary Selection: Selected top sentences for the summary using heapq.
- heapq.nlargest() is used to select the most significant sentences based on their scores.

Generated Summary

- The generated summary should provide a concise and accurate representation of the original text, highlighting the key points and important information

```
Summary:
The shark's head
was out of water and his back was coming out and the old man could hear
the noise of skin and flesh ripping on the big fish when he rammed the
harpoon down onto the shark's head at a spot where the line between his
eyes intersected with the line that ran straight back from his nose.
 No one would steal from the old man but it was better to take the sail
and the heavy lines home as the dew was bad for them and, though he was
quite sure no local people would steal from him, the old man thought
that a gaff and a harpoon were needless temptations to leave in a boat.
```

Machine Learning Task

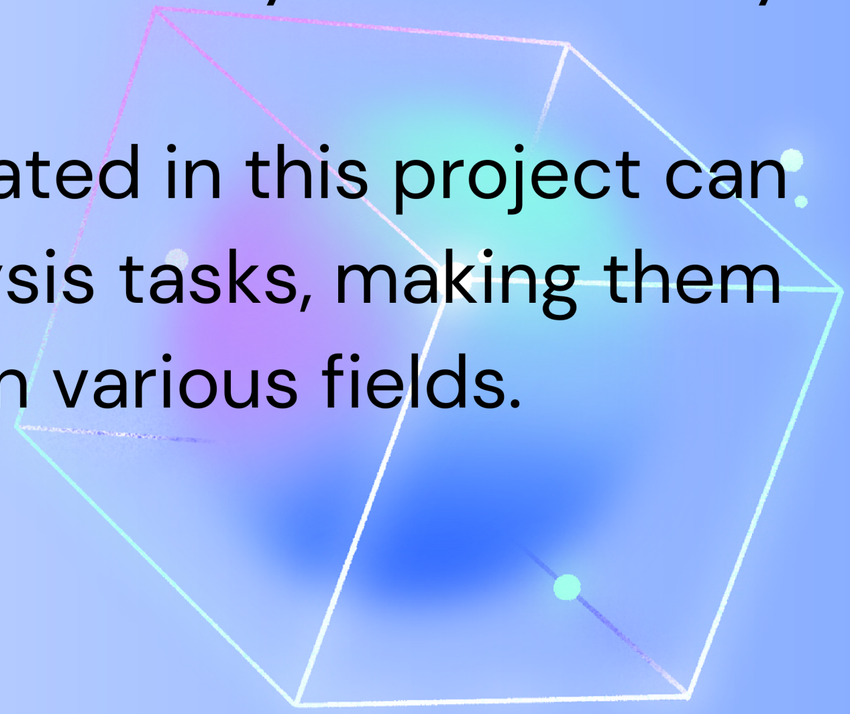
- Library: Scikit-learn

- Task: Predict word frequencies using a Random Forest Regressor.

- Using Scikit-learn, we can build a machine learning model to predict word frequencies. T
- The Mean Squared Error (MSE) is a common metric used to evaluate the performance of regression models. A lower MSE indicates better model performance and accuracy.

Conclusions

- This project highlights the power of using multiple libraries in Python to perform complex data analysis tasks. By automating the processes, we can achieve greater efficiency and accuracy in data analysis.
- The methods and techniques demonstrated in this project can be applied to a wide range of data analysis tasks, making them valuable tools for professionals in various fields.



Future Work

- Future work can focus on exploring more sophisticated machine learning models, such as neural networks or ensemble methods, to enhance prediction capabilities.

- The methodologies used in this project can be applied to various types of textual data, including news articles, social media posts, and research papers.

- Enhancing the summarization process can involve incorporating more advanced NLP techniques, such as transformer models or topic modeling, to produce more accurate and informative summaries.