Moby Dick Word Frequency Analysis

Executive Summary

This project demonstrates a complete end-to-end text analytics workflow in Python by extracting, cleaning, and analyzing the text of Herman Melville's "Moby Dick" from Project Gutenberg. The analysis finds the most frequent non-common words in the novel and visualizes word frequency distributions, showcasing the power of Python's web scraping and NLP tools.

Key outcomes:

- Web scraping and parsing of Moby Dick from a public HTML source
- Data cleaning: tokenization, lowercasing, and stop word removal using NLTK
- Discovery and plotting of the most common content words in the novel

Recommendation:
Similar methods can be applied to any classic novel or large text to understand thematic word use and stylistic choices.

Technical Report

1. Introduction

The goal is to analyze the text of "Moby Dick" programmatically—extracting the novel from the web, cleaning it, and determining word usage patterns. Tools used: `requests`, `BeautifulSoup`, and `NLTK`.

2. Data and Sources

- **Primary Text Source:**
  Project Gutenberg Moby Dick HTML
- **Packages Used:**
  - `requests` (web scraping)
  - `BeautifulSoup` (HTML parsing)
  - `nltk` (text processing: tokenization, stopword removal, frequency analysis)
  - `Counter` (from `collections` for word counting)
  - `matplotlib` (for plotting, via `nltk.FreqDist.plot()`)

3. Methodology / Code

A. Download & Parse HTML

```
import requests
from bs4 import BeautifulSoup

# Download the HTML web page
r = requests.get('https://s3.amazonaws.com/assets.datacamp.com/production/project_147/dat
r.encoding = 'utf-8'
moby_text = r.text
```

B. Extract Raw Text

```
html_soup = BeautifulSoup(moby_text, 'html.parser')
text = html_soup.get_text()
```

C. Tokenize and Normalize

```
import nltk
tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')
tokens = tokenizer.tokenize(text)
words = [w.lower() for w in tokens]
```

D. Remove Stopwords

```
nltk.download('stopwords')
stop_words = nltk.corpus.stopwords.words('english')
words_no_stop = [w for w in words if w not in stop_words]
```

E. Compute and Visualize Frequency

```
freqdist = nltk.FreqDist(words_no_stop)
freqdist.plot(25)
top_ten = freqdist.most_common(10)
print(top_ten)
```

4. Analysis & Key Results

- **Most Frequent Words:**
  The word frequency distribution (excluding common English stopwords) reveals the dominant terms in the novel—likely including "whale", "ahab", and "sea".

- **Visualization:**
  A plot of the 25 most frequent content words provides an immediate sense of Moby Dick's main themes and recurring motifs.

5. Conclusions

- **Pipeline Success:**
  This notebook demonstrates a functional and repeatable pipeline for downloading, cleaning, and analyzing public domain text data.
- **NLP Showcase:**
  The workflow is a strong example of applied data science and NLP skills, ideal for portfolios.