The objective of this project was to build a **reusable text preprocessing pipeline** that can standardize and clean raw text data for **Natural Language Processing (NLP) tasks**.
The pipeline ensures the text is consistent, noise-free, and ready for downstream tasks such as feature extraction and model training.

Methodology

1. Load Dataset

- Used a sample dataset of text sentences (can be replaced with CSV/JSON).
- Dataset loaded into a Pandas DataFrame for easier manipulation.

2. Convert Text to Lowercase

- All text converted to lowercase for uniformity.
  Example: "Hello WORLD" → "hello world"

3. Remove Punctuation & Special Characters

- Regex used to strip unwanted characters like .,!?@#.

4. Remove Numbers (if irrelevant)

- Numeric values removed (e.g., 123), unless important for analysis.

5. Tokenization

- Split text into words (tokens) using **NLTK word_tokenize**.
  Example: "hello world" → ["hello", "world"]

6. Remove Stopwords

- Common words (e.g., *the, is, and*) removed using **NLTK stopword list**.
  Example: "the cat is on the mat" → ["cat", "mat"]

7. Stemming / Lemmatization

- **Stemming:** reduces words to their base root (e.g., "studying" → "studi").
- **Lemmatization:** converts words to dictionary form (e.g., "studying" → "study").
- Both were applied in this pipeline.

8. Save Cleaned Text

- Tokens rejoined into cleaned sentences.

- Final dataset saved as:
  - **CSV file** → cleaned_dataset.csv
  - **JSON file** → cleaned_dataset.json

Implementation

- Implemented in **Python** using:
  - **NLTK** (tokenization, stopwords, stemming, lemmatization)
  - **spaCy** (advanced lemmatization)
  - **pandas** (data handling)
  - **regex** (cleaning text)
- All preprocessing steps encapsulated in a reusable **TextPreprocessor class**.

Text Preprocessing Pipeline Deliverables

1. **Cleaned Dataset** (CSV & JSON format).
2. **Reusable Preprocessing Class** (TextPreprocessor).
3. **Documentation** (inline code comments + this summary report).
4. **Sample Notebook/Demo** (before & after examples).
5. *(Optional)* **Unit tests** (not included).

Original Sentence	**Cleaned Output**
Hello WORLD! This is a sample sentence, with numbers like 123.	hello world sampl sentenc number like
NLTK & SpaCy are amazing tools for NLP preprocessing!!	nltk spaci amaz tool nlp preprocess
The cats are running, studied hard, and will be studies again...	cat run studi hard studi

Conclusion

The preprocessing pipeline successfully:

- Normalized text by removing noise (punctuation, numbers, stopwords).
- Reduced word forms via stemming/lemmatization.
- Produced a clean dataset ready for feature extraction and model training.

This pipeline is **modular, reusable, and extendable** for any future NLP tasks.