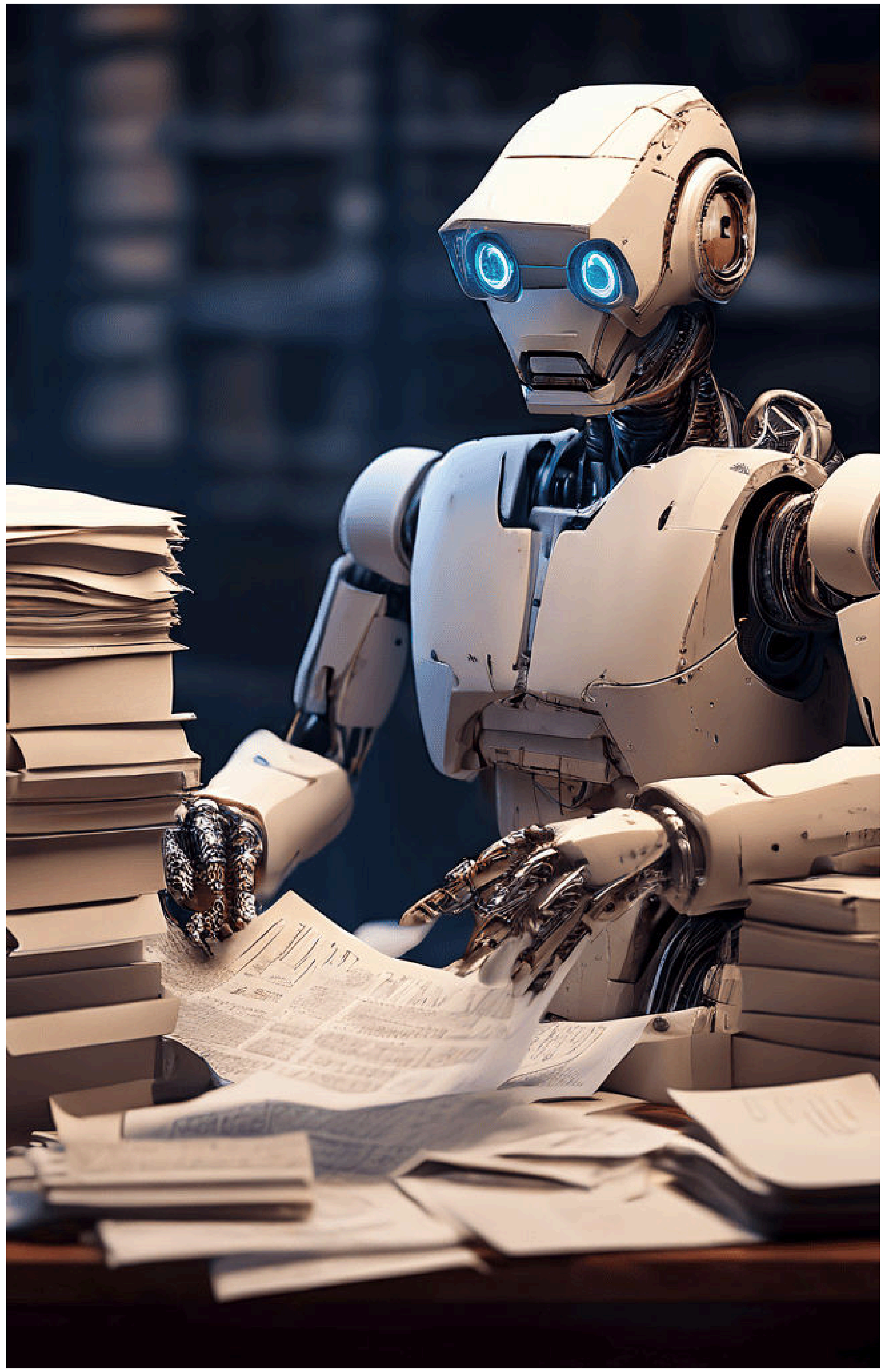
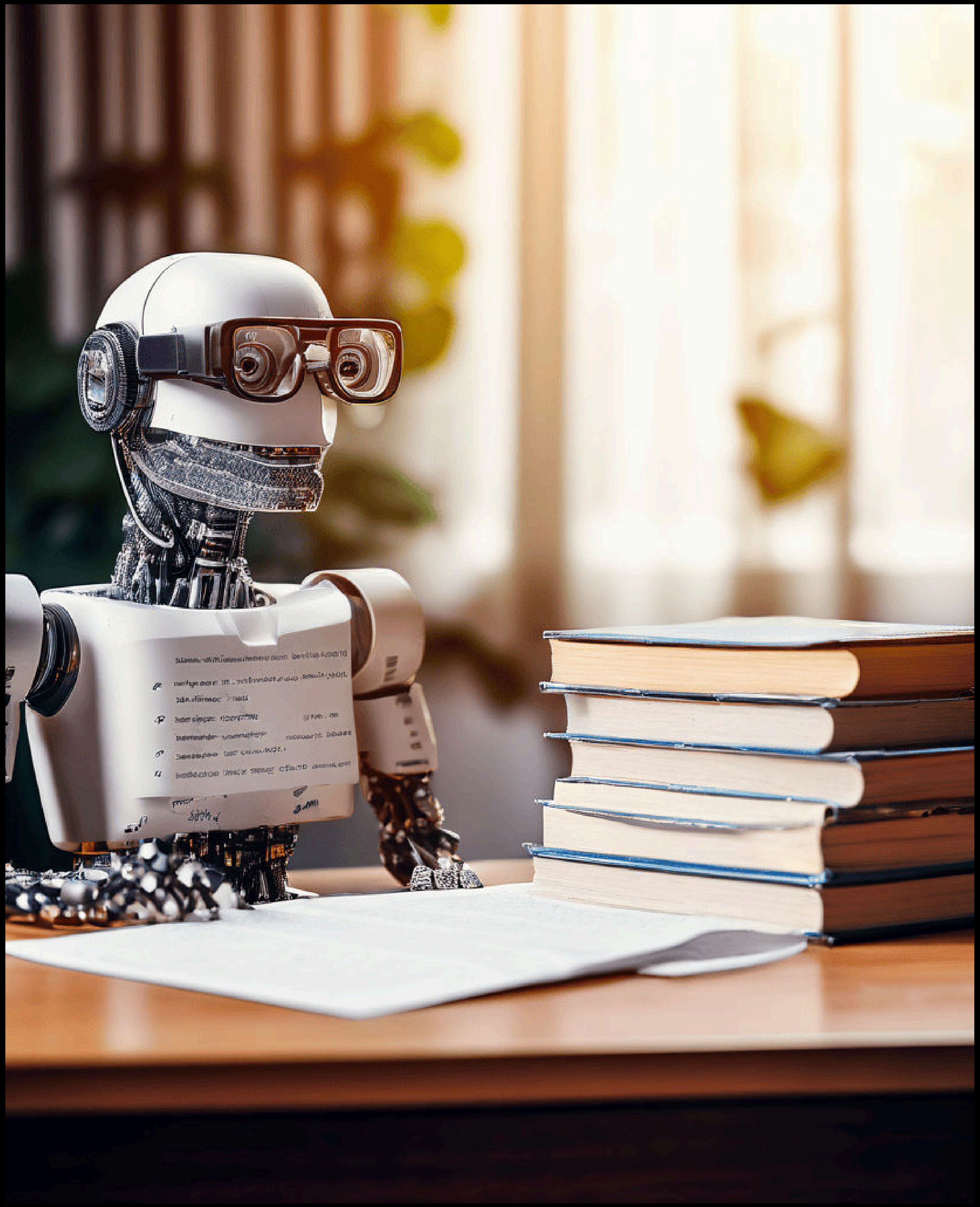


Project: NLP Document Classification

Final Project Report

A project by Konstantinos Soufleros



- Group Name: NLP_Task_Force-Document Classification
- Name: Konstantinos Soufleros
- Email: soufleros.kostas@gmail.com
- Country: Serbia
- Company: Data Glacier
- Specialization: NLP
- Github: https://github.com/kostas696/DG_Intern/tree/main/week13
- Internship Batch: LISUM30
- Date: 29/04/2024

Introduction

Document classification involves assigning categories to text documents, such as news articles, emails, or forum posts. Our aim is to demonstrate how machine learning techniques can be applied to classify documents accurately and efficiently.

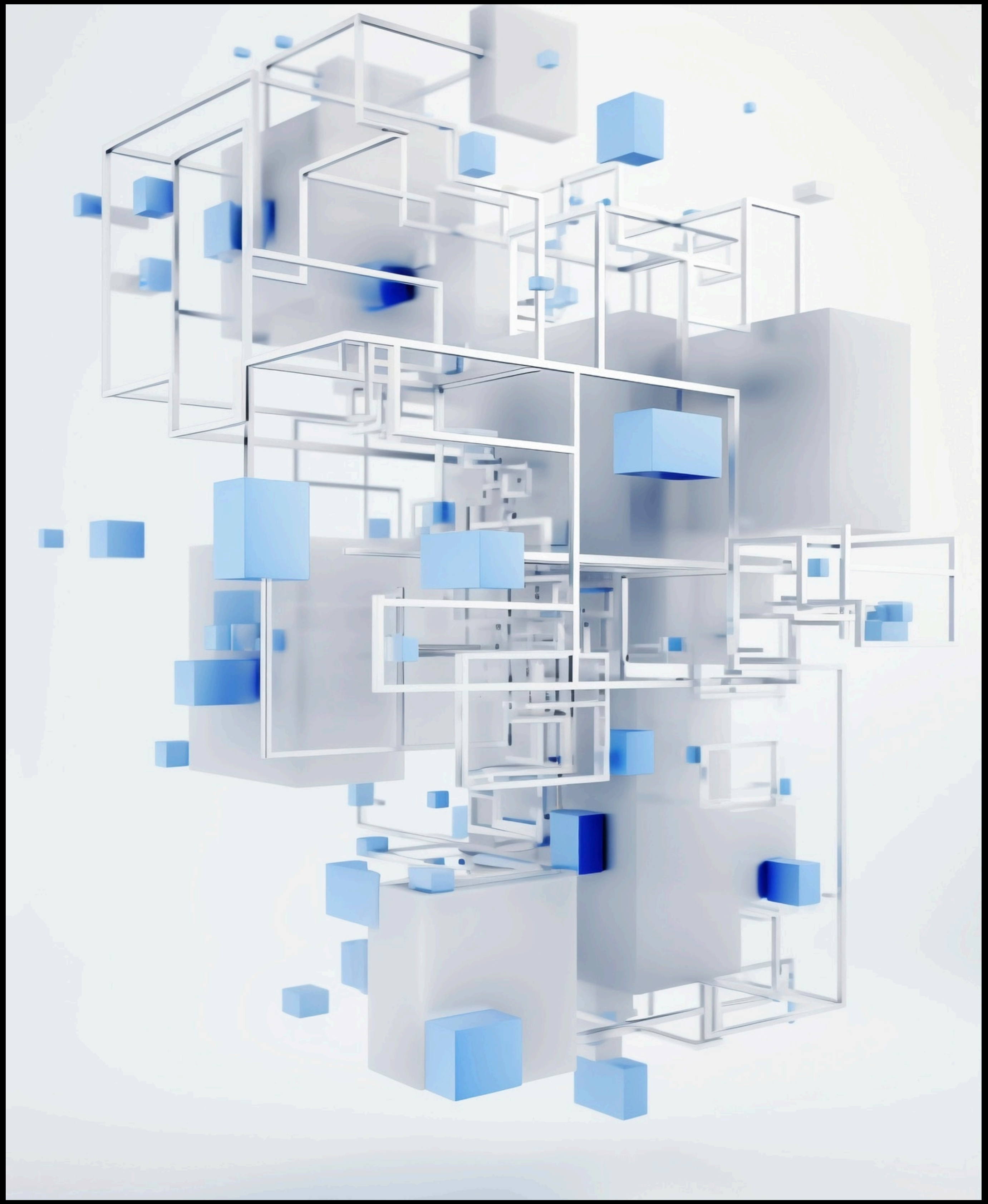
Objective: The primary objective of this project is to develop a robust document classification system using the 20 Newsgroups dataset. This dataset consists of approximately 20,000 documents, divided into 20 different newsgroups. Our goal is to train a machine learning model that can accurately classify documents into their respective newsgroups based on their content. This system has applications in various domains, including spam filtering, email routing, and sentiment analysis.

Agenda

- Problem Description
- Data Information
- Data Understanding
- Exploratory Data Analysis (EDA)
- Modeling
- Model Evaluation
- Model Selection
- Fine-tuning and Optimization
- Final Model Deployment
- Conclusion

Problem Description

The problem revolves around analyzing a dataset containing newsgroup documents categorized into various topics. The **primary objective** is to gain insights into the content of these documents, understand the prevalent themes within each newsgroup, and identify any patterns or trends present in the data.



Data Information

Total number of observations: 19997

Total number of newsgroups: 20

Number of observations per newsgroup:
rec.sport.hockey: 1000
sci.med: 1000misc.forsale: 1000
alt.atheism: 1000
rec.sport.baseball: 1000
rec.autos: 1000
comp.sys.ibm.pc.hardware: 1000
rec.motorcycles: 1000
talk.politics.mideast: 1000
talk.politics.misc: 1000
talk.politics.guns: 1000
talk.religion.misc: 1000
comp.windows.x: 1000
sci.space: 1000
comp.graphics: 1000
comp.sys.mac.hardware: 1000
comp.os.ms-windows.misc: 1000
sci.crypt: 1000
soc.religion.christian: 997
sci.electronics: 1000

Size of the data (in bytes): 46132928

This balanced distribution ensures that there is no bias towards any particular category during the analysis.

Data Understanding

The dataset consists of documents from 20 different newsgroups, covering a wide range of topics such as sports, religion, politics, technology, and more. Each document is labeled with its corresponding newsgroup category, providing a structured format for analysis.

The data is structured and tabular, organized into a pandas DataFrame with two columns: 'Newsgroup' and 'Original_Content'.

- No missing values
- Document length variation
- Presence of special characters, numbers, and stopwords.
- Need for preprocessing

Exploratory Data Analysis (EDA)

- Dataset Extraction
- Dataset Exploration
- Data Preprocessing: Removed metadata headers, emails, numbers, and 'GMT' indicators. Tokenized the text and converted words to lowercase. Removed stopwords, punctuation, and single characters. Lemmatized the tokens to their base forms. Use of two methods for preprocessing.
- Visualizations: Generated WordClouds for each newsgroup to visualize common words. Plotted the distribution of document lengths after the preprocessing. Displayed average document lengths per newsgroup.
- Insights: Derived insights from the data exploration process.

Preprocessing with two methods

Original Text
print(df["Original_Content"][1500])

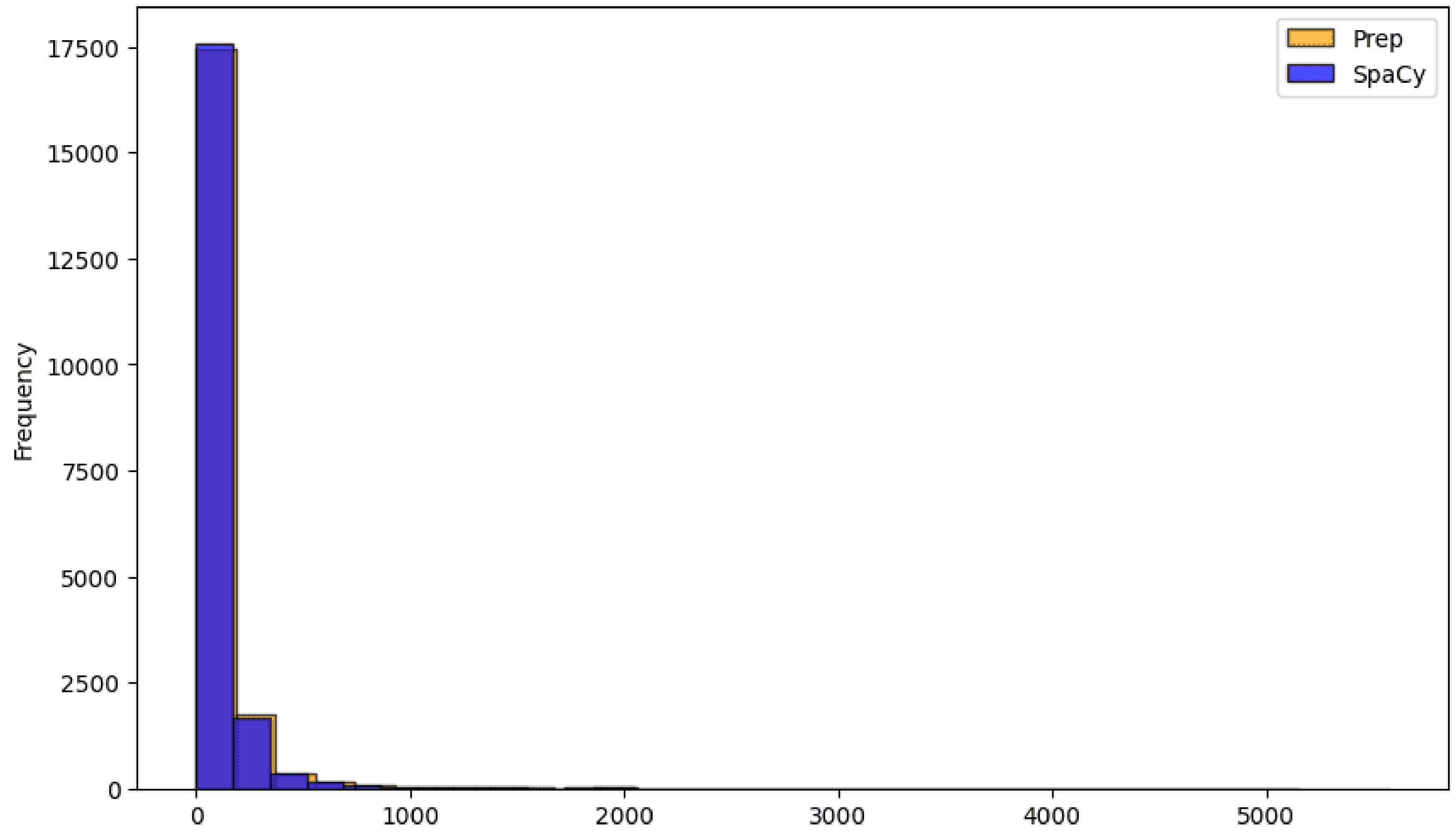
Xref: cantaloupe.srv.cs.cmu.edu sci.research:4033 sci.med:58154 alt.psychoactives:2253 sci.psychology:11783Newsgroups: sci.research,sci.med,alt.psychoactives,sci.psychologyPath: cantaloupe.srv.cs.cmu.edu!crabapple.srv.cs.cmu.edu!fs7.ece.cmu.edu!europa.eng.gtefsd.com!howland.reston.ans.net!zaphod.mps.ohio-state.edu!uwm.edu!ux1.cso.uiuc.edu!usenet.ucs.indiana.edu!jh224-718622.ucs.indiana.edu!userFrom: bshelley@ucs.indiana.edu ()Subject: Xanax...please provide infoMessage-ID: <bshelley-060493181020@jh224-718622.ucs.indiana.edu>Followup-To: sci.research,sci.med,alt.psychoactives,sci.psychologySender: news@usenet.ucs.indiana.edu (USENET News System)Nntp-Posting-Host: jh224-718622.ucs.indiana.eduOrganization: Indiana UniversityDate: Tue, 6 Apr 1993 23:15:26 GMTLines: 9I am currently doing a group research project on the drug Xanax. I wouldbe exponentially gracious to receive any and all information you couldprovideme regarding its usage, history, mechanism of reaction, side effects, andother pertinent information. I don't care how long or how short yourresponse is.Thanks in advance!Brent E. Shelley

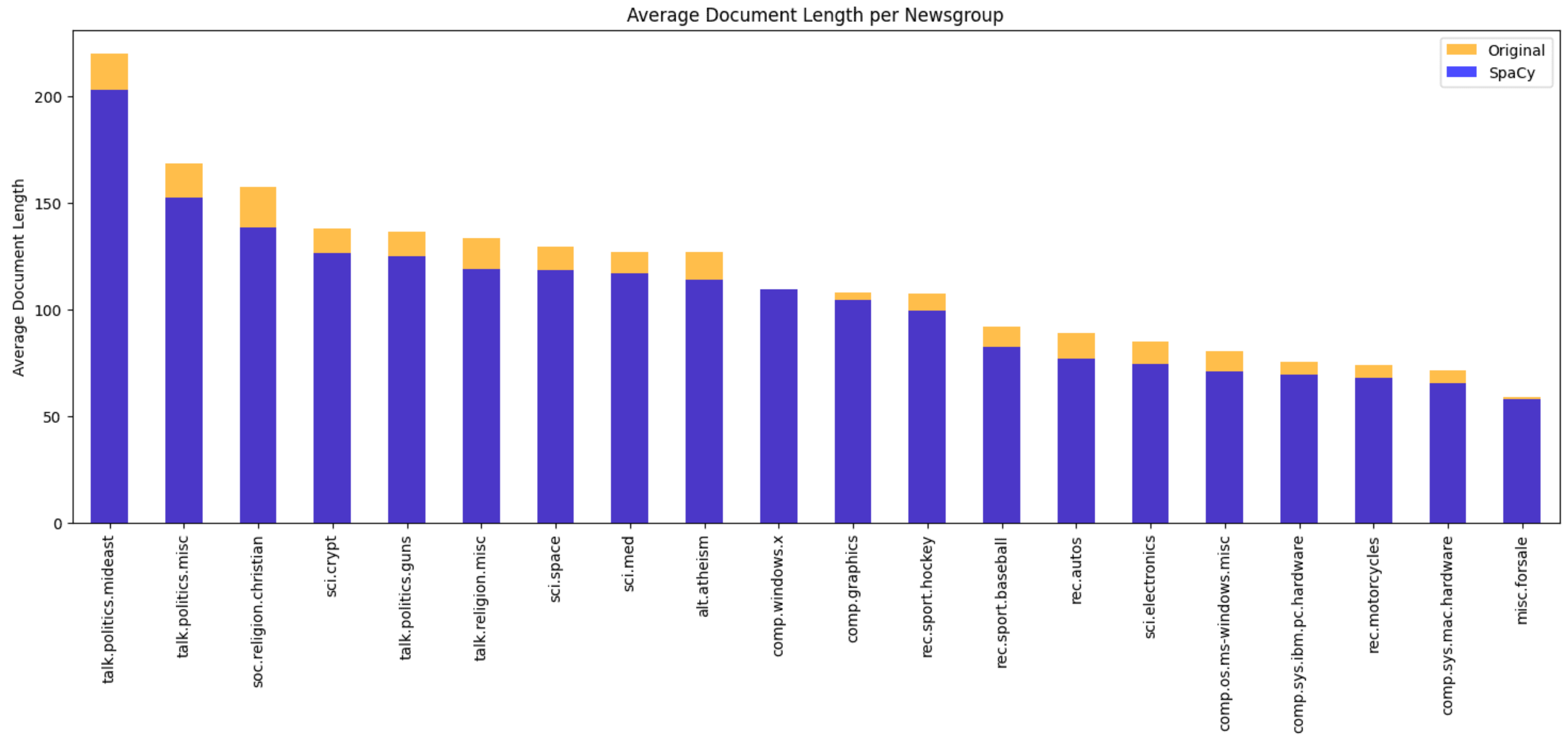
Preprocessing with first method
df["Content_prep"][1500]

'currently group research project drug xanax would exponentially gracious receive information could provide regarding usage history mechanism reaction side effect pertinent information care long short response thanks advance brent shelley'

Preprocessing with SpaCy
df['Content_spacy'][1500]

'currently group research project drug xanax exponentially gracious receive information provide usage history mechanism reaction effect pertinent information care long short response thank advance brent shelley'





Most Common Words in Preprocessed Content:

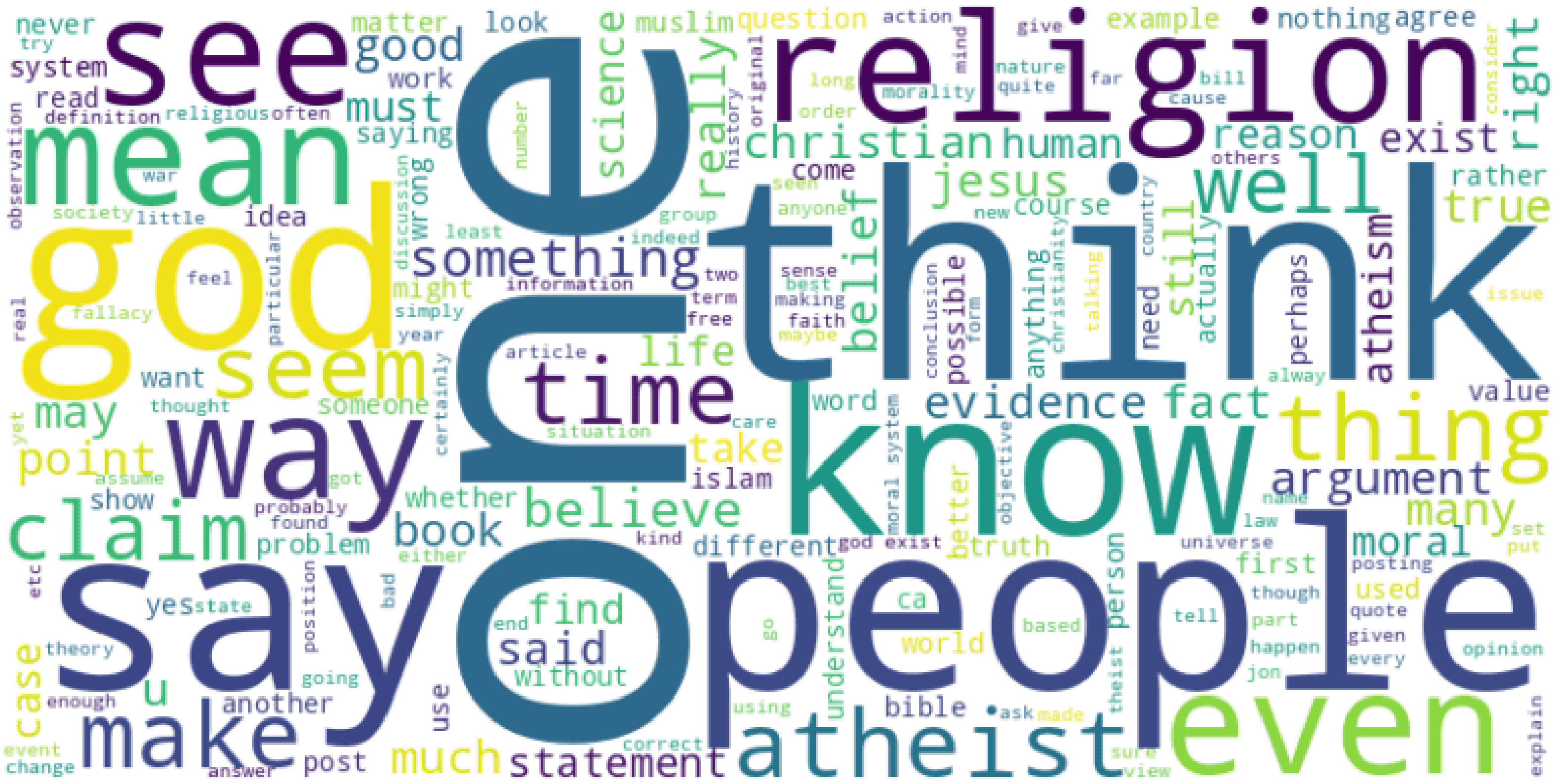
[('would', 16216), ('one', 15072), ('people', 9881), ('like', 9479), ('know', 9084), ('get', 8754), ('time', 7730), ('think', 7646), ('also', 7117), ('could', 6510), ('use', 6416), ('make', 6262), ('say', 5999), ('right', 5878), ('good', 5600), ('year', 5568), ('way', 5507), ('even', 5503), ('system', 5451), ('new', 5370)]

Most Common Words in SpaCy Content:

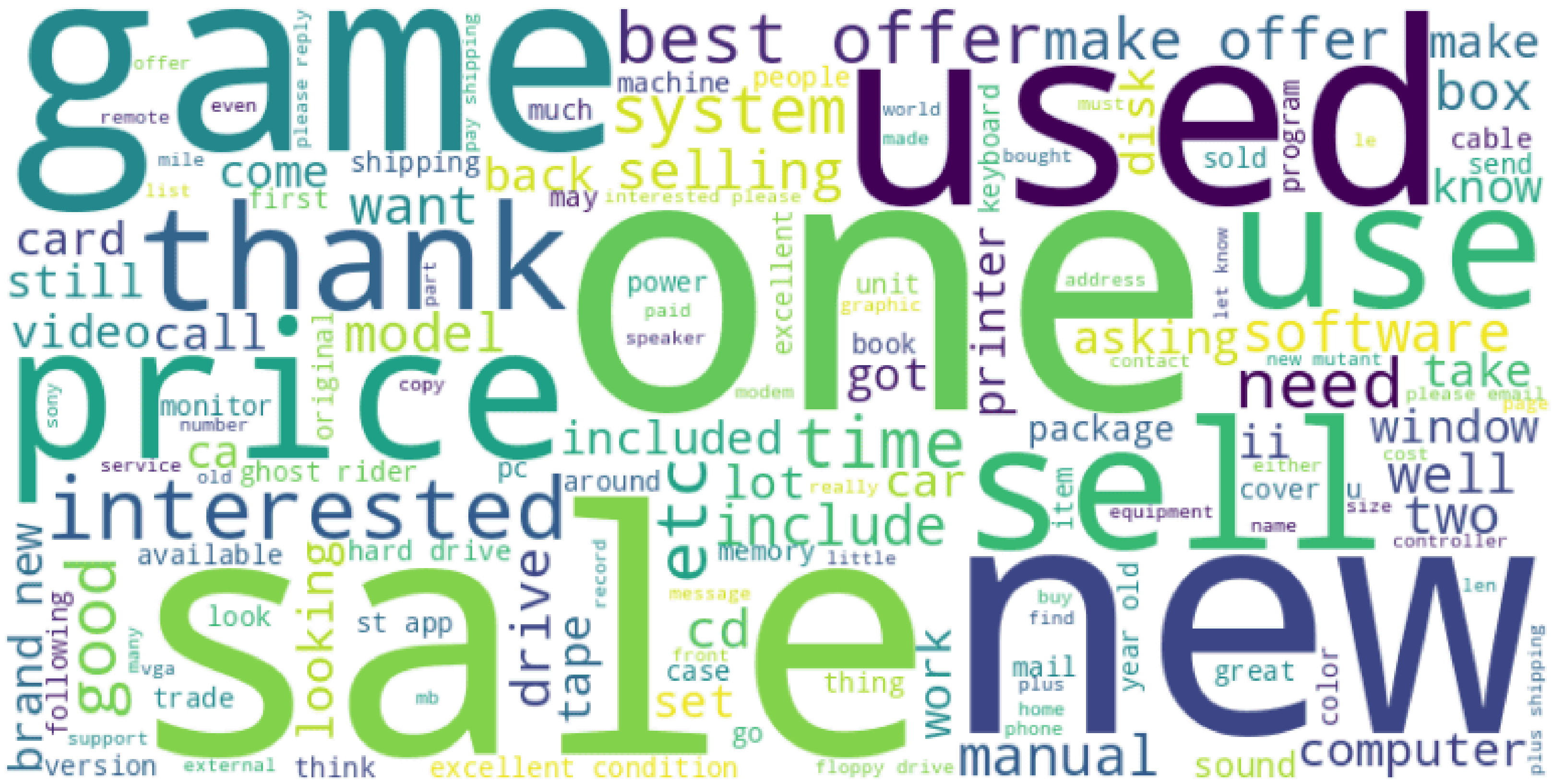
[('know', 10850), ('people', 9919), ('like', 9741), ('think', 9405), ('x', 8369), ('time', 7830), ('good', 7492), ('use', 7267), ('say', 6743), ('work', 6190), ('right', 5832), ('year', 5681), ('want', 5675), ('go', 5671), ('new', 5616), ('way', 5580), ('come', 5368), ('thing', 5339), ('look', 5261), ('find', 5128)]

WordCloud Examples

alt.atheism

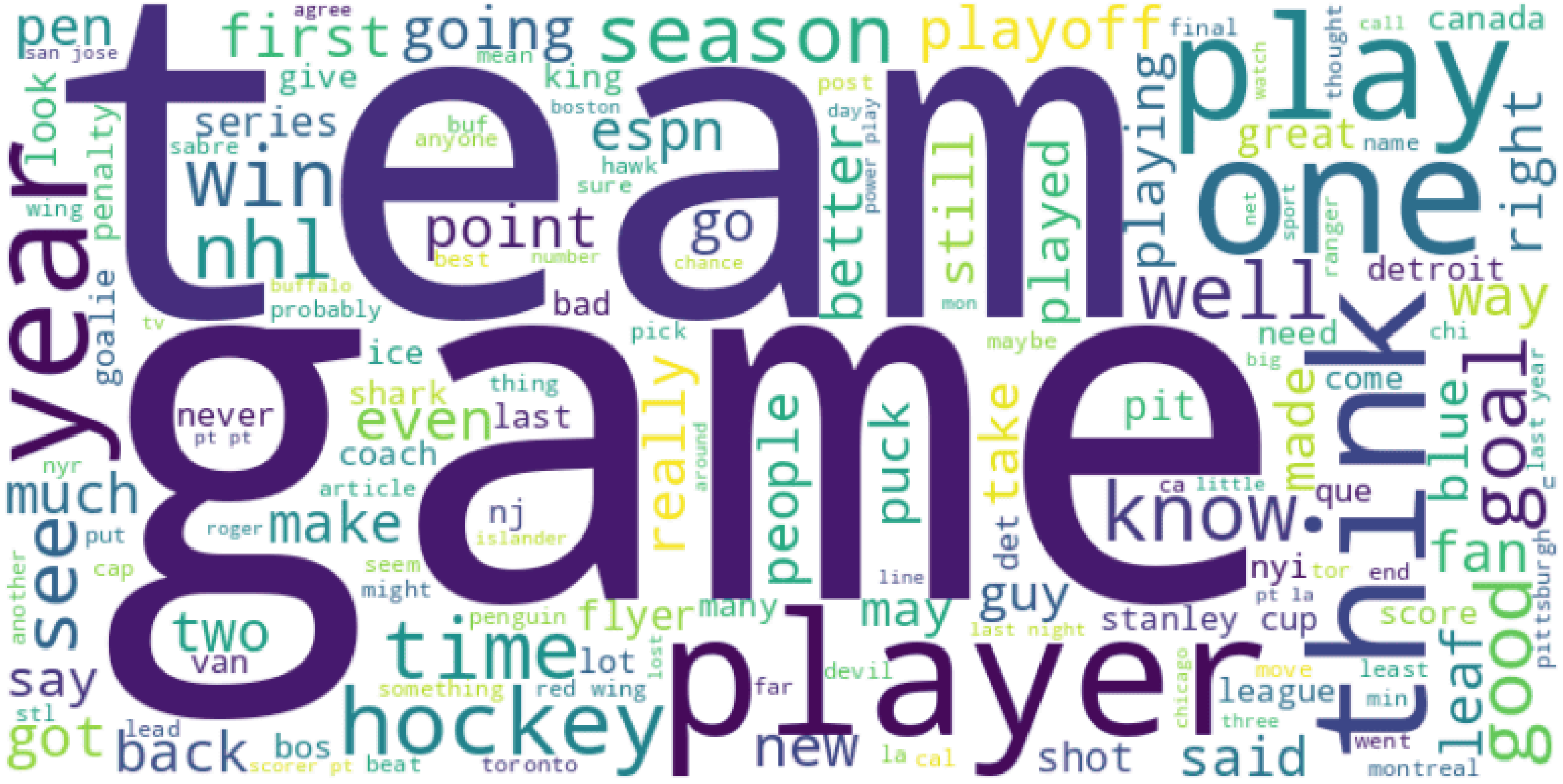


misc.forsale



sci.med

rec.sport.hockey



Modeling

Modeling Approach

- **Preprocessed Dataset:** Utilized preprocessing techniques including metadata removal, tokenization, lemmatization, and TF-IDF feature representation.
- **Feature Representation**: We will utilize TF-IDF (Term Frequency-Inverse Document Frequency) for feature representation.
- **Model Training and Evaluation**: Train models including Naive Bayes and SVM (Support Vector Machine) with linear kernel. Evaluate models using F1-score and ROC-AUC score.
- **Evaluation Metrics**: F1-score: Measures the balance between precision and recall. ROC-AUC score: Indicates the model's ability to distinguish between classes.

Model Evaluation

Training and Evaluating

Evaluation results for Naive Bayes with TF-IDF features from original content:
1. F1-score: 0.7870150082967651
2. ROC-AUC score: 0.9824499575838908

Evaluation results for SVM with TF-IDF features from original content:
1. F1-score: 0.8084097615468069
2. ROC-AUC score: 0.9841962751354506

Evaluation results for Naive Bayes with TF-IDF features from SpaCy preprocessed text:
1. F1-score: 0.7912974031564216
2. ROC-AUC score: 0.9829453549077195

Evaluation results for SVM with TF-IDF features from SpaCy preprocessed text:
1. F1-score: 0.8119686592997641
2. ROC-AUC score: 0.984382666205357

Model Selection

Best Model

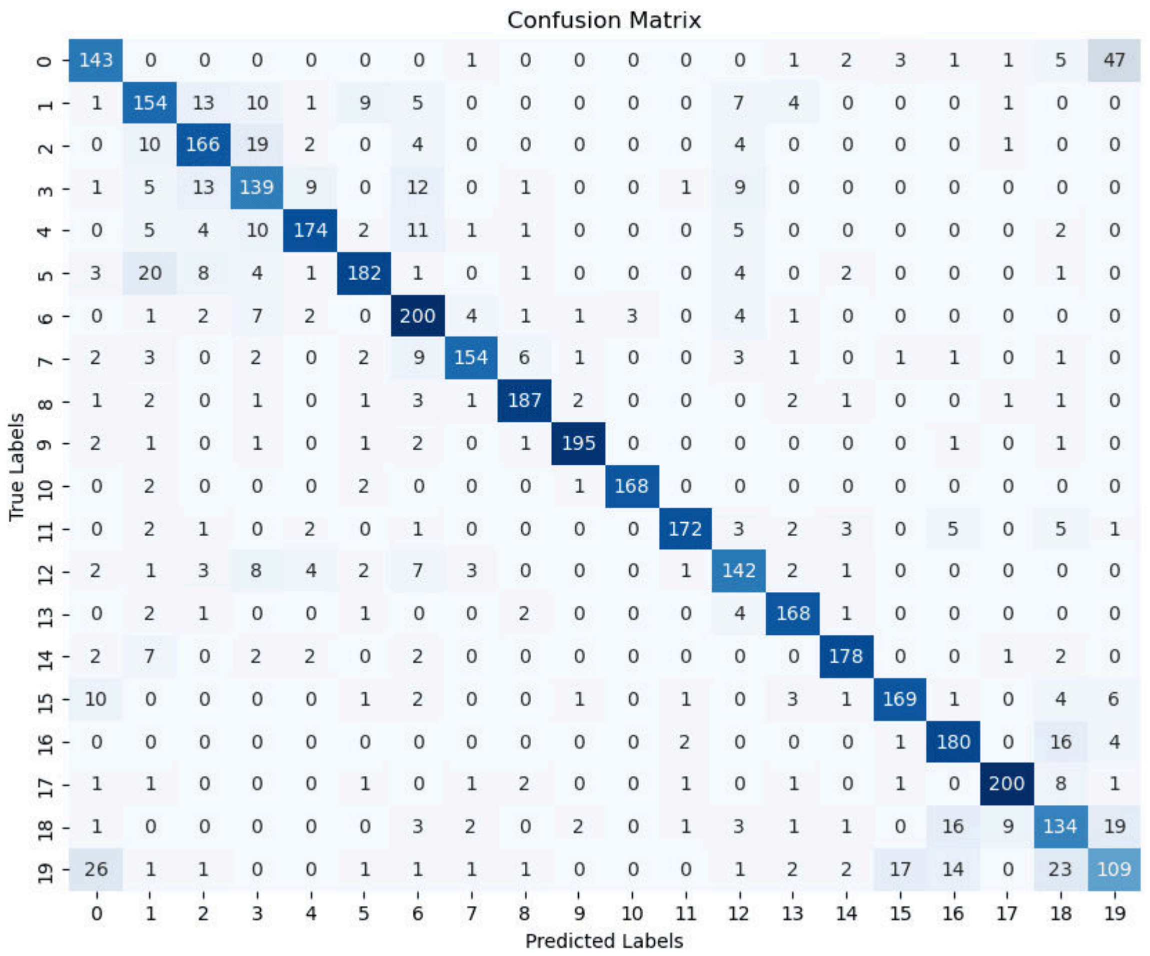
Best model based on F1-score and ROC-AUC: ***SVM with TF-IDF from SpaCy preprocessed text***

Best Mode	Feature Representation	F1-scor	ROC-AUC Scor
SVM	TF-IDF from SpaCy preprocessed text	0.811969	0.984383

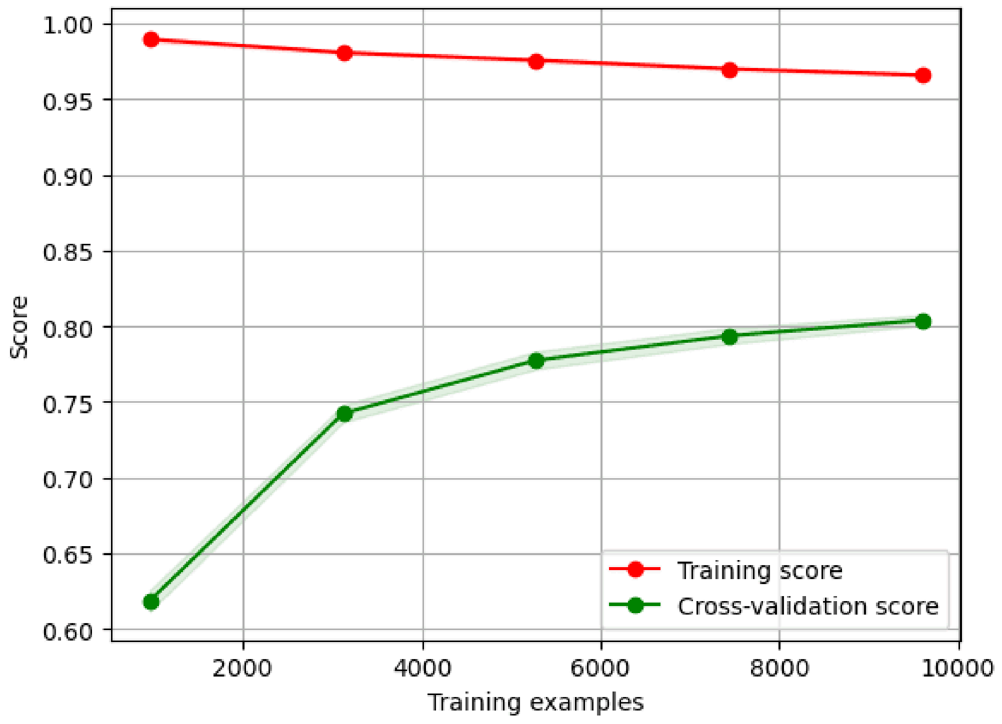
Fine-tuning and Optimization

Hyperparameter Tuning with RandomizedSearchCV

RandomizedSearchCV(cv=5, estimator=SVC(kernel='linear', probability=True), n_jobs=-1,
 param_distributions={'C': [0.1, 1, 10],
 'kernel': ['linear', 'poly', 'rbf', 'sigmoid']},
scoring='f1_macro')
estimator: SVC(kernel='linear', probability=True)

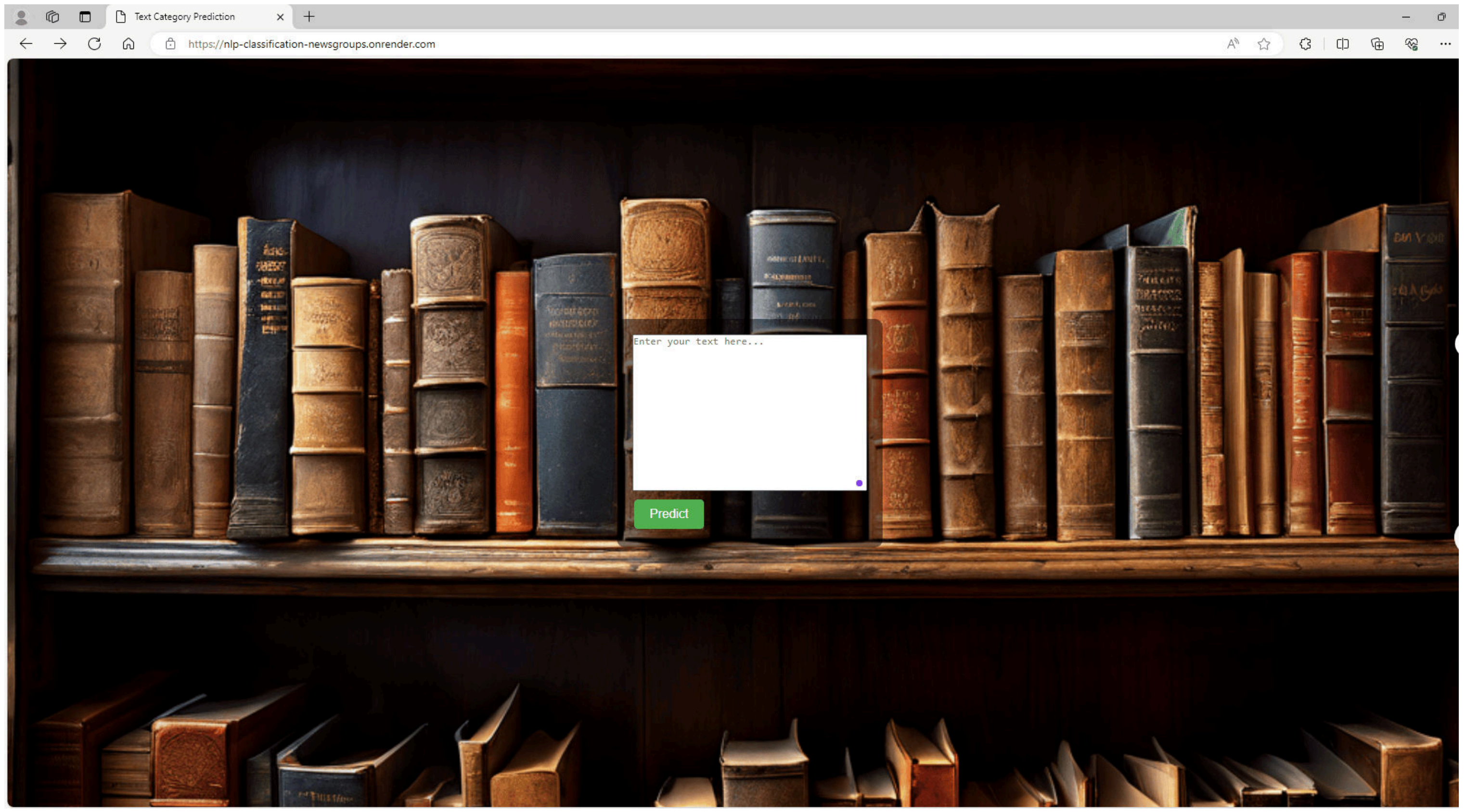


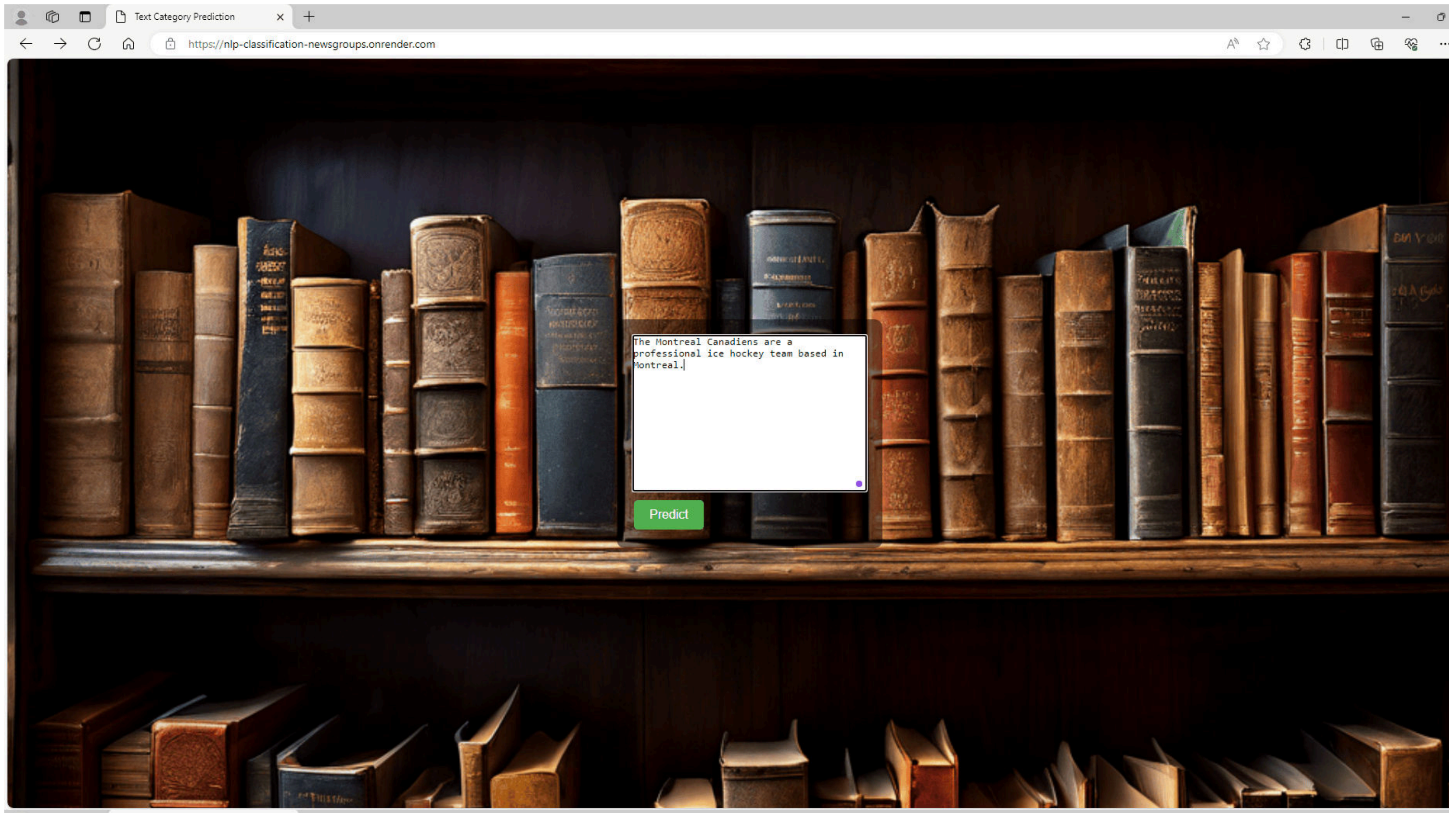
Best parameters: {'kernel': 'linear', 'C': 1}
Best score: 0.8171082037725308

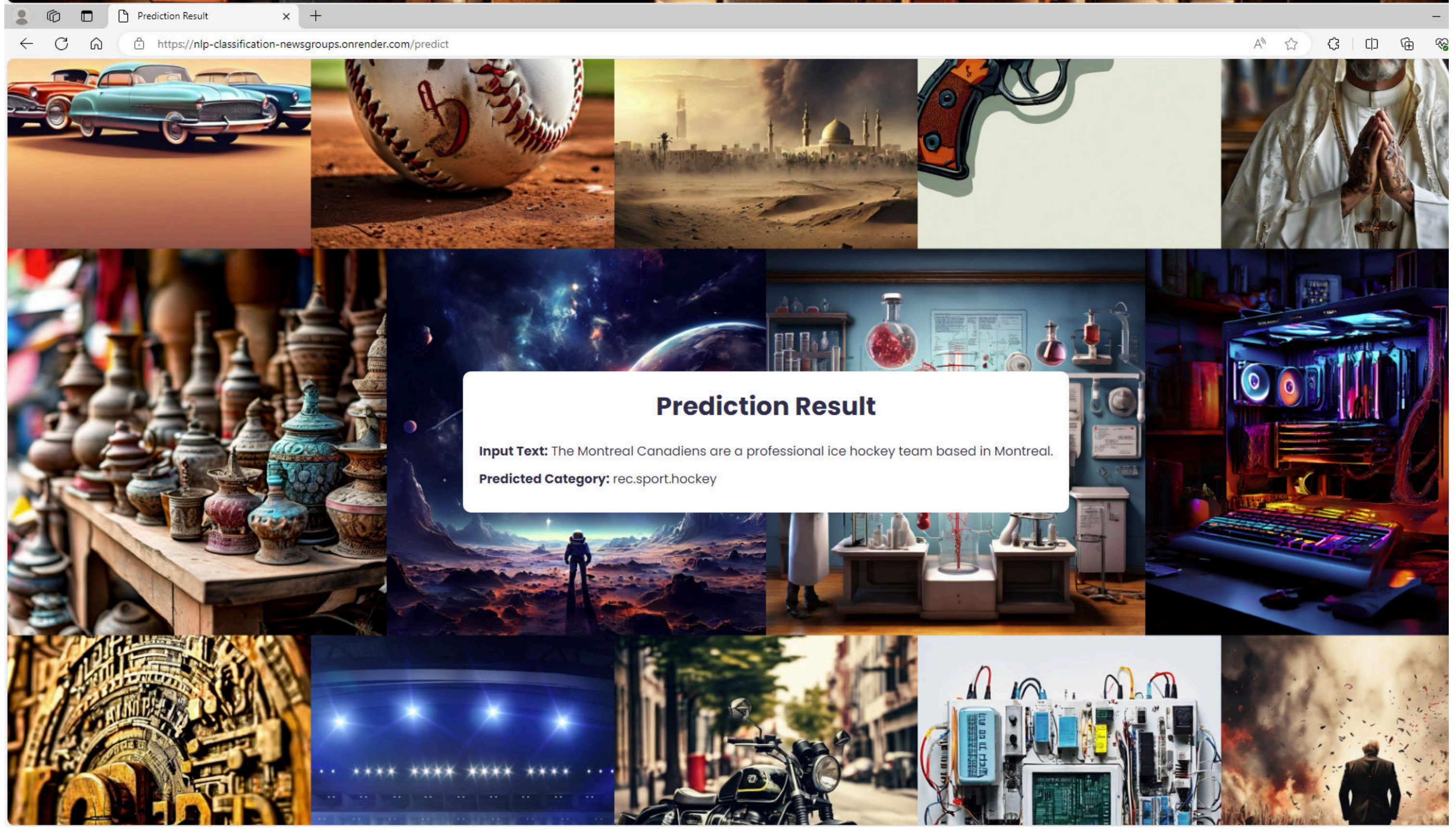


Evaluation results for the fine-tuned model on the test set:
F1-score: 0.8294339676375511
ROC-AUC score on the test set: 0.98708677596335

Final Model Deployment







- **Created a Flask Web Application**
  Developed a Flask web application to host the trained model. Utilized Python and Flask framework for building the backend.
- **Deployed on Render Platform**
  Leveraged Render platform for seamless deployment. Ensured scalability and reliability of the deployed application.
- **Deployment Link**
  https://nlp-classification-newsgroups.onrender.com
  Access the deployed application for real-time classification.
- **Screenshots**
  Attached screenshots showcasing the deployed application.

Conclusion

In conclusion, advanced text classification using NLP opens up a world of possibilities for extracting insights from unstructured text data. By leveraging cutting-edge techniques, organizations can unlock valuable information and improve decision-making processes.

Thanks!

Do you have any questions?
soufleros.kostas@gmail.com
https://www.linkedin.com/in/konstantinos-soufleros
https://github.com/kostas696