Week 13: Deliverables

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

PROBLEM DESCRIPTION

The problem revolves around analyzing a collection of newsgroup posts to gain insights into the topics, themes, and sentiments expressed within them. Newsgroups are online discussion forums where users share information, ask questions, and engage in conversations related to specific topics.

The key challenges in this project include:

- **Data Understanding and Cleaning**: The dataset comprises a vast amount of unstructured text data. Cleaning and preprocessing this data to remove noise, irrelevant information, and standardize its format are essential steps.
- **Topic Modeling**: Identifying the main topics or themes discussed across different newsgroups is crucial for understanding the content.
- **Sentiment Analysis**: Understanding the sentiment expressed in the posts can provide valuable insights into the users' opinions and attitudes towards various topics.
- **Visualization**: Visualizing the data through word clouds, topic distributions, and sentiment heatmaps can facilitate better interpretation and communication of the findings.

BUSINESS UNDERSTANDING

Leveraging the rich source of information available in newsgroup posts can provide businesses with valuable insights that can drive strategic decision-making, enhance customer satisfaction, and foster business growth.

- **Market Research**: Understanding the topics and sentiments prevalent in newsgroups related to specific industries or products can provide valuable market insights.
- **Customer Feedback Analysis**: Newsgroup posts often contain valuable feedback and opinions from customers about products or services.
- **Competitive Analysis**: Monitoring discussions in newsgroups related to competitors' products or industry trends can provide valuable competitive intelligence.
- **Community Engagement**: Engaging with users in newsgroup communities can help businesses build brand awareness and foster relationships with potential customers.

PROJECT LIFECYCLE

Weeks	Date	Deliverables
Week 7	19 March 2024	Problem Understanding
Week 8	26 March 2024	EDA of Dataset
Week 9	2 April 2024	Data Preprocessing
Week 10	9 April 2024	Model Building & Training
Week 11	16 April 2024	Performance Evaluation & Reporting
Week 12	23 April 2024	Model Deployment
Week 13	29 April 2024	Model Inference

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.

TYPE OF DATA

The data is structured and tabular, organized into a pandas DataFrame with two columns: 'Newsgroup' and 'Content'. The 'Newsgroup' column contains categorical labels indicating the category of each document, while the 'Content' column contains the textual content of the documents.

DATA PROBLEMS

- No missing values: The dataset does not contain any missing values in either the 'Newsgroup' or 'Content' columns, as confirmed by checking for null values.
- Document length variation: The length of documents varies across different newsgroups, with some containing longer texts compared to others. This variation in document length might affect certain analyses or models.
- Presence of special characters, numbers, and stopwords: The textual content contains special characters, numbers, and stopwords that may not contribute meaningfully to the analysis. These elements need to be removed or filtered out to focus on relevant textual features.
- Need for preprocessing: The textual content requires preprocessing to standardize the format, remove unnecessary elements, and prepare it for further analysis or modeling tasks.

APPROACHES TO ADDRESS DATA PROBLEMS

Text Preprocessing: A preprocessing function is applied to clean and standardize the textual content. This function involves:

- Removal of metadata headers, emails, numbers, and 'GMT' mentions.
- Tokenization to split the text into individual words.
- Lowercasing to ensure consistency in word case.
- Removal of punctuation, non-alphabetic characters, single characters, and stopwords.
- Lemmatization to reduce words to their base or dictionary form.

Word clouds and histograms are generated to visualize the distribution of words and document lengths across different newsgroups. These visualizations help in identifying common themes, prevalent words, and document length patterns within each category.

Average document lengths per newsgroup are calculated to understand the variation in document sizes across different categories.

Using the SpaCy Method we:

- Utilized SpaCy model for preprocessing, including tokenization and lemmatization.
- Applied SpaCy preprocessing to 'Original_Content' column.
- Visualized document length distribution and common words in SpaCy preprocessed content.
- Recalculated average document length per newsgroup.
- Compared document length distribution and common words between original and SpaCy preprocessed content.
- Assessed differences in preprocessing impact on document lengths and common words.

Results:

Both methods effectively cleaned text data, but SpaCy offered streamlined preprocessing. Document length and common word distributions varied slightly between methods. SpaCy preprocessing showed potential improvements in tokenization and lemmatization.

SpaCy-based text preprocessing offers a more efficient and comprehensive approach compared to the original method. By leveraging advanced NLP capabilities, SpaCy improves the accuracy and consistency of preprocessing tasks, leading to better-quality textual data for subsequent analysis and modeling. The comparison highlights the benefits of adopting SpaCy for text preprocessing in data science workflows.

MODELING

Best Model Selection

The dataset was split into training, validation, and test sets. Two feature representations were utilized: TF-IDF from the original content and TF-IDF from SpaCy preprocessed text. Two models were evaluated: Naive Bayes and SVM.

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

Hyperparameter Tuning

Hyperparameter optimization using RandomizedSearchCV was employed for hyperparameter tuning due to its efficiency compared to grid search. The search space included parameters for SVM models. The best parameters and score were identified through Randomized optimization.

Further evaluation of the best fine-tuned model was conducted on the test set to assess its performance.

The results and best model were saved using pickle for future reference.

MODEL EVALUATION

Evaluation results for Naive Bayes with TF-IDF features from original content:

F1-score: 0.7870150082967651

ROC-AUC score: 0.9824499575838908

Evaluation results for SVM with TF-IDF features from original content:

F1-score: 0.8084097615468069

ROC-AUC score: 0.9841962751354506

Evaluation results for Naive Bayes with TF-IDF features from SpaCy preprocessed text:

F1-score: 0.7912974031564216

ROC-AUC score: 0.9829453549077195

Evaluation results for SVM with TF-IDF features from SpaCy preprocessed text:

F1-score: 0.8119686592997641

ROC-AUC score: 0.984382666205357

Model Selection

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

Fine-tuning and Optimization

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}

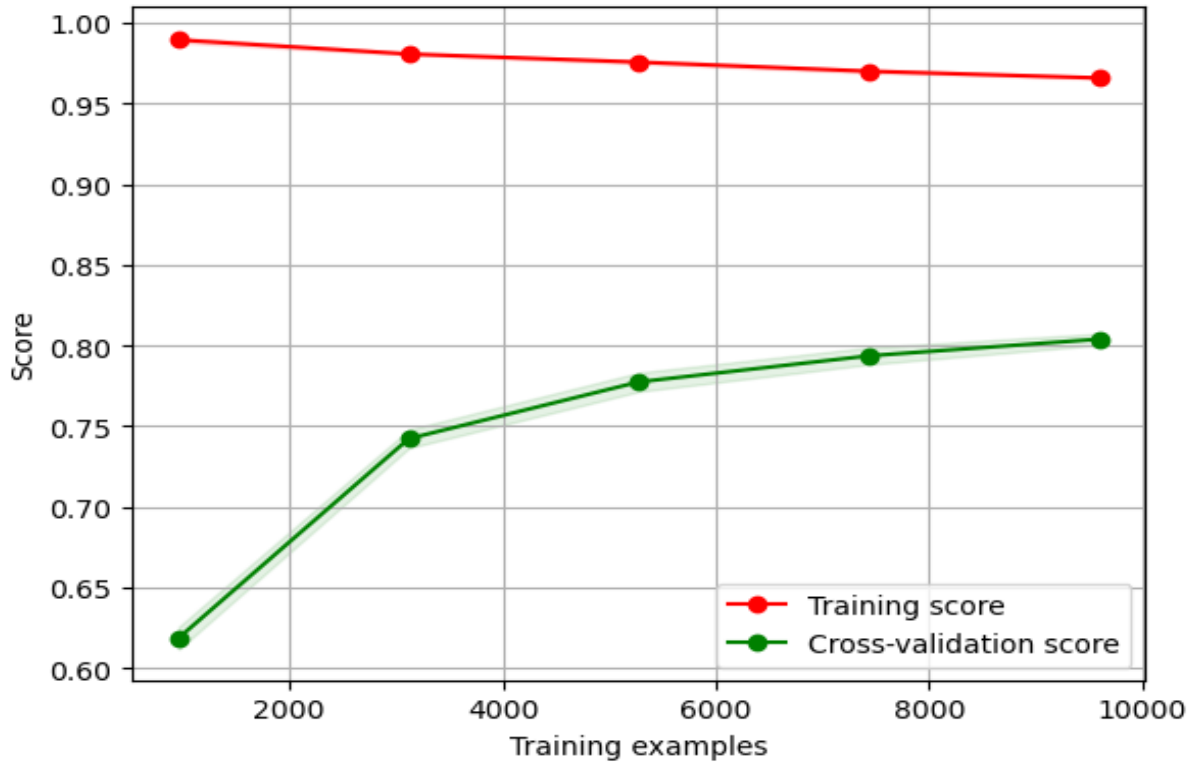
Best score: 0.8171082037725308

Evaluation results for the fine-tuned model on the test set:

- F1-score: 0.8294339676375511
- ROC-AUC score on the test set: 0.987086775596335

Overall, the fine-tuned model demonstrates strong performance in classifying text documents into their respective categories, with high precision, recall, and F1-scores across most categories and a high ROC-AUC score, indicating robustness and effectiveness in classification. More specific, our metrics of choice:

The weighted average F1-score of 0.829 indicates overall good performance of the model across all categories.
The ROC-AUC score evaluates the model's ability to distinguish between the positive and negative classes. A score of 0.987 indicates high performance in terms of class separation, with values closer to 1 indicating better performance.



This plot illustrates the learning curves of the machine learning model. The x-axis represents the number of training examples, while the y-axis represents the model's performance score. The red line shows the training score, indicating how well the model fits the training data as more examples are used. The green line shows the cross-validation score, which measures the model's performance on unseen data during training.

This indicates that the model's performance on unseen data (cross-validation) improves significantly with more training examples, suggesting that the model generalizes better to new data as the training set size increases.

Overall, the gap between the training and cross-validation scores decreases as more training examples are used, indicating that the model's variance decreases with additional data. This suggests that the model benefits from more training data and may further improve with additional training examples.

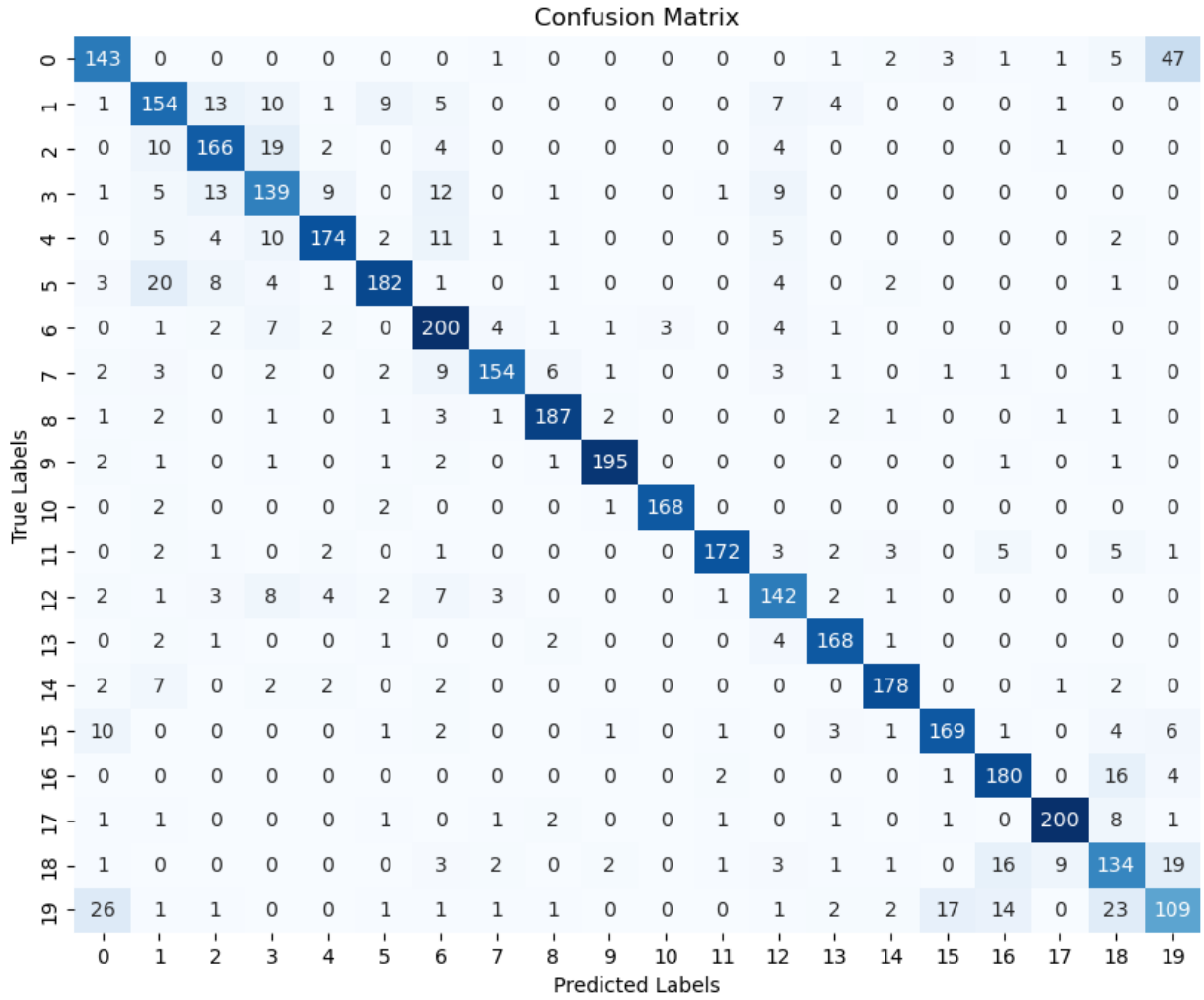
Our confusion matrix visualizes the performance of the model by comparing the true labels (actual categories) against the predicted labels. Each cell in the matrix represents a combination of predicted and true labels. The values in the cells indicate the number of instances that fall into each category.

The rows represent the true labels (actual categories) of the data.

The columns represent the predicted labels (categories predicted by the model).

The diagonal elements from top-left to bottom-right represent the instances where the true label matches the predicted label, indicating correct predictions.

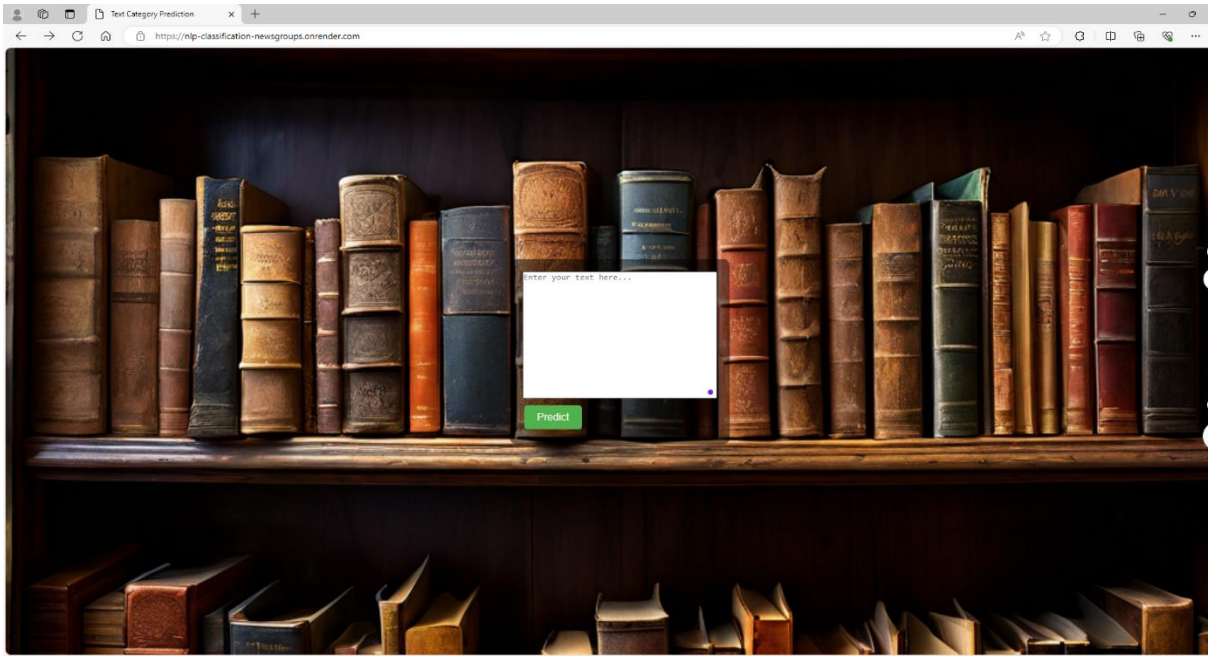
Off-diagonal elements represent instances where the predicted label does not match the true label, indicating misclassifications.

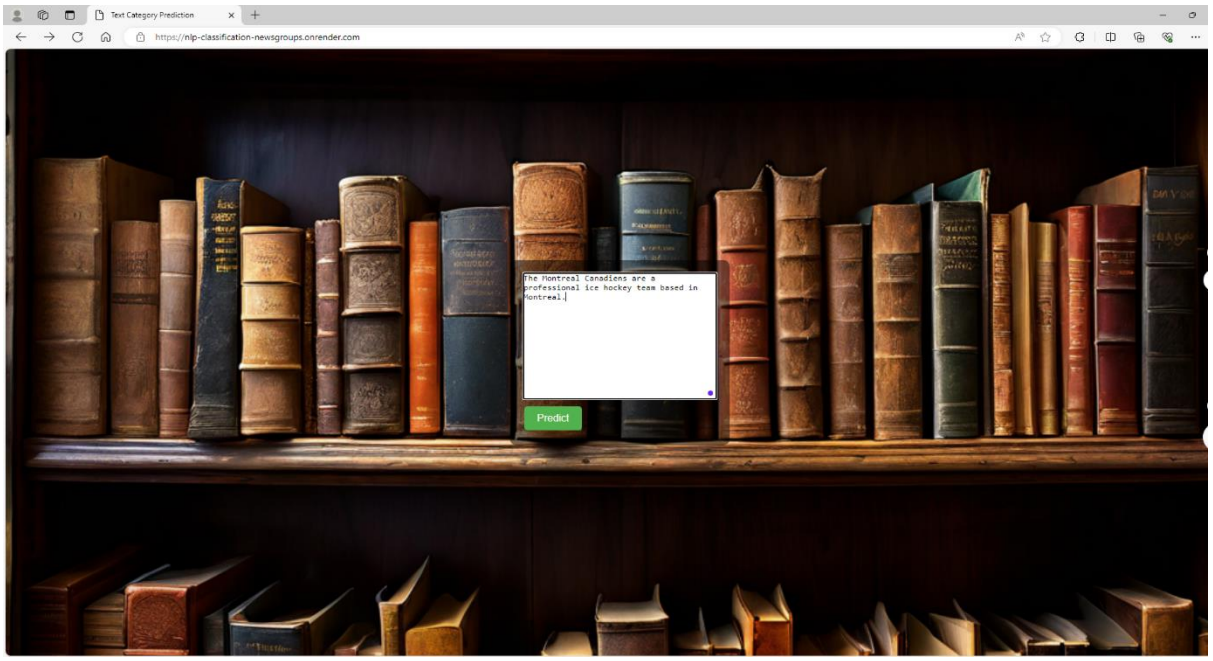


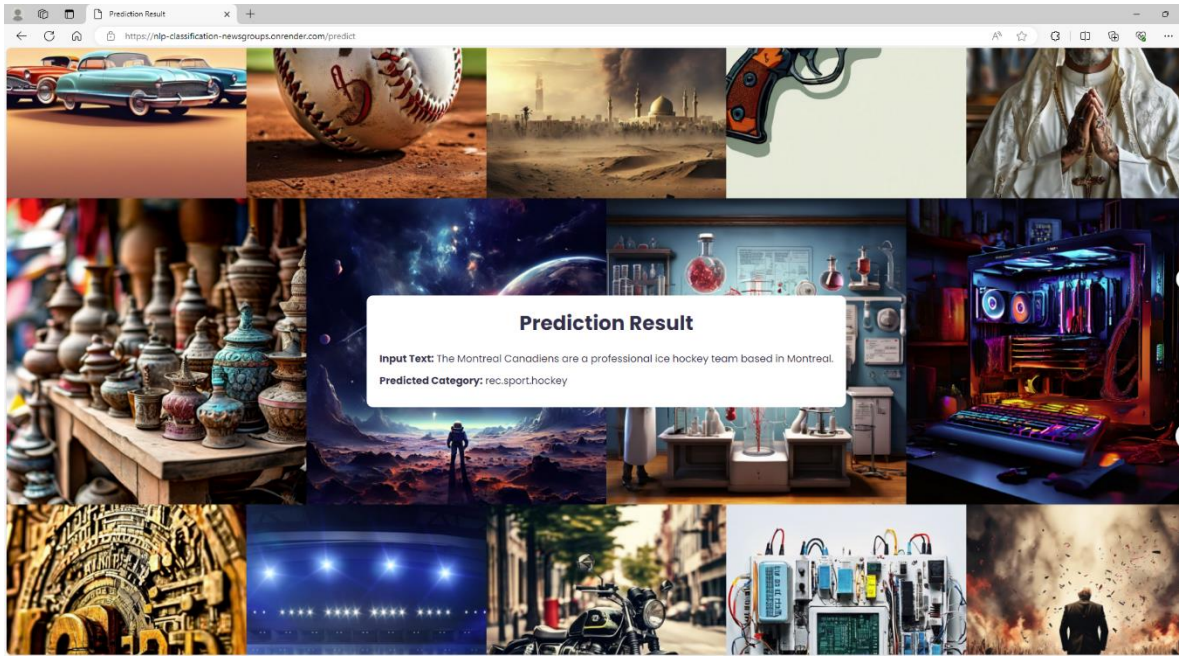
FINAL MODEL DEPLOYMENT

- 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.




Text Category Prediction
https://nlp-classification-newsgroups.onrender.com
The Montreal Canadiens are a professional ice hockey team based in Montreal.
Predict


Prediction Result
https://nlp-classification-newsgroups.onrender.com/predict
Prediction Result
Input Text: The Montreal Canadiens are a professional ice hockey team based in Montreal.
Predicted Category: rec.sport.hockey