Alice Tang and Yinda Chen

Code and Documentation Review for Yuang Guo and Heng Bai

I. Overall GitHub Structure

The overall structure of the GitHub is clear and allows those interested in the project to efficiently locate each week's assignments and relevant documents. For this peer review, it was easy to find the materials to evaluate. However, we have a few suggestions for improvement.

While this current structure works well for our review, it may not be as intuitive for external users, such as potential employers or resumes, if you plan on showcasing this project on your resume. To improve clarity and organization, it may be helpful to use the cookiecutter template that was suggested in the first week's assignment. The template recommends organizing files into separate folders, such as data, docs, models, notebooks, references, reports, and src. For example, a few of the data-related files are currently located in the main branch. Moving these into designated folders would make the repository more structured and easier to navigate.

Implementing this approach may significantly help with the overall professionalism and usability of your GitHub repository.

II. Code Output and Jupyter Notebook Review

Below is a review we've conducted of each week's strengths, weaknesses, and potential suggestions for improvement in the code.

Week 2 - Week 3

- **Strengths**
    - The initial stages of the project focused on collecting and cleaning the dataset. The data preprocessing pipeline was systematically set up, including tasks such as text normalization (lowercasing, removing special characters, etc.). This ensured that the data was ready for further processing and model training.
- **Weaknesses**
    - While the data cleaning process was well-executed, the early stages lacked advanced text preprocessing techniques like tokenization, lemmatization, or stemming, which could have further improved text quality and consistency.
- **Suggestions**

- Future work should consider incorporating more sophisticated text preprocessing steps like lemmatization or stemming to better handle variations in word forms. Also, data augmentation methods could have been explored to address any issues with class imbalance early on.

Week 4 - Week 5

- **Strengths**
  - These weeks focused on implementing various text embedding techniques, such as CountVectorizer and TF-IDF, which are essential for converting text into machine-readable formats. The models were trained on these embeddings, and the early results were promising.
- **Weaknesses**
  - Although TF-IDF showed promising results, the CountVectorizer model could have been further optimized. Additionally, the model evaluation process could have included more comprehensive metrics, such as precision and recall, instead of only focusing on accuracy.
- **Suggestions**
  - Future work should explore additional embedding techniques, such as Word2Vec or fastText, which could capture semantic relationships between words more effectively. Also, model evaluation should include more diverse metrics to gain a better understanding of performance, especially in cases of class imbalance.

Week 6 - Week 7

- **Strengths**
  - These weeks were dedicated to implementing and testing machine learning models, including KNN, XGBoost, and MLP. The performance of each model was evaluated, and a good amount of experimentation with hyperparameter tuning was conducted.
- **Weaknesses**
  - The hyperparameter tuning process, although valuable, was not deeply documented. It would have been beneficial to record how different hyperparameter configurations impacted model performance to ensure future

reproducibility. Additionally, the training process for XGBoost and MLP could have been more optimized.

- **Suggestions**
    - It would be beneficial to incorporate a more structured hyperparameter optimization process, possibly using techniques such as grid search or random search. Detailed logging of hyperparameter configurations and model performance would allow for better insights into the tuning process and lead to more informed decisions.

Week 8 - Week 9

- **Strengths**
    - Focus shifted to handling class imbalance and improving model performance on the minority class using SMOTE. This helped in achieving a more balanced model and provided better performance on underrepresented classes.
- **Weaknesses**
    - The model evaluation during these weeks still relied heavily on accuracy, which can be misleading when dealing with imbalanced datasets. The evaluation did not include advanced metrics such as AUC or the confusion matrix, which would have been valuable for deeper insights.
- **Suggestions**
    - Incorporating additional evaluation metrics such as precision, recall, F1 score, and AUC would provide a more comprehensive assessment of the model's performance, particularly for imbalanced datasets. Also, exploring alternative class balancing techniques, such as undersampling or using class weights, could yield further improvements.

Week 10 - Week 13

- **Strengths**
    - In the final weeks of the project, focus was placed on fine-tuning the best model combination (TF-IDF + MLP) and preparing it for deployment. The model's interpretability was also considered through the use of SHAP values, ensuring that the model's decision-making process was transparent and understandable.
- **Weaknesses**

- Despite the good final model performance, there was minimal attention paid to inference speed or the scalability of the solution. The project did not test the model under real-world constraints, such as processing large amounts of data in a reasonable time.

- **Suggestions**
  - Future work should consider evaluating the model's inference time and scalability, especially if the model will be deployed in real-time systems. It would also be beneficial to explore model compression techniques or server-side optimizations to ensure the model can scale efficiently in production.

III. Final Report Feedback

For this section, we mainly focused on the final document that represents an end-to-end project, since we thought it would provide the best summary of your project.

Overall, the document gives a clear and concise overview of the project, including the workflow, model architecture, and highlights. However, as a document that encompasses the entire process, we did feel as if it was a bit too brief and lacked explanation in specific key areas. Adding some more detail and expanding on specific sections would help better communicate the depth of your work and the insights you've gained throughout the term.

A. Suggestions for Improvement

1. **Introduction**
   a. This was a great section that did well in summarizing the project's objects and approach. However, there was some context lacking. We would have liked to see a bit more as to why Twitter sentiment analysis is important and relevant to real-world applications, as well as a brief mention of specific challenges of sentiment analysis on social media.
2. **Workflow Overview**
   a. The way this section is structured makes it very concise and easy to follow, however it is a bit surface-level. This section could be made stronger by expanding on the data preprocessing steps (explain why you chose lemmatization over stemming), the model training process (discuss more dataset characteristics

or hyperparameters optimized), as well as the evaluation metrics (accuracy, in your case, and explain why).

3. **Results and Visualizations**
    a. Perhaps adding some visualizations of the confusion matrix you talked about, along with a brief analysis, would significantly improve the section.
    b. Additionally, it would have been interesting to see all the different models you tried and show a comparison table with the baseline model.
    c. Discussing patterns with misclassifications of the model struggling would have been interesting to read as well.
4. **Highlights and Strengths**
    a. While the strengths are well-noted, the section could be made more impactful by connecting these points to specific results.
5. **Suggestions for Improvement**
    a. This is a really strong section! A suggestion could be to give a more detailed reflection on what didn't work or the areas where the model underperformed.
6. **Conclusion**
    a. The conclusion does well with wrapping up the project, but it feels slightly rushed, a suggestion could be to summarize the key results and findings of the project.

Overall, your group did a solid job with this document and provides a good foundation, but expanding on the sections above would provide a more holistic picture of your project and the work involved.

IV. Overall Project Strengths and Weaknesses

A. Strengths

Systematic Approach

The project covers all key stages of a machine learning pipeline, including data preprocessing, feature extraction, model building, evaluation, interpretability analysis, and deployment. Each phase was executed well, ensuring smooth progress and solid results.

Comprehensive Model Comparison and Optimization

The project thoroughly explored multiple text embedding methods (CountVectorizer, TF-IDF, Word2Vec) and different machine learning models (e.g., KNN, XGBoost, MLP). Through extensive experimentation and tuning, the best model combination (TF-IDF + MLP) was selected, achieving an impressive accuracy of 94.5%.

Focus on Model Interpretability and Ethics

The project emphasized model interpretability by using SHAP to analyze feature importance and validate the model's performance across different sentiment categories. This focus on transparency and fairness is crucial for building user trust in real-world applications.

Effective Data Augmentation and Handling

The project addressed class imbalance issues using SMOTE, which helped improve performance on minority classes. Data cleaning, regularization, and feature selection were also well-executed, ensuring the model's robustness.

Well-Prepared for Deployment

The project saved the trained model and provided clear instructions on how to deploy it for real-time predictions, demonstrating good foresight for practical application.

B. Weaknesses

Limited Model Optimization Analysis

While hyperparameter tuning was performed, the analysis of model optimization was somewhat shallow. There was limited discussion on how specific hyperparameters affected the model's performance, and the rationale behind parameter choices could have been explored in greater detail.

Lack of Advanced Feature Engineering

Although the data was cleaned and preprocessed well, the project could have benefited from more advanced feature engineering techniques, such as using more complex embeddings like BERT, or incorporating additional features such as sentiment lexicons or sentiment polarity.

Narrow Performance Evaluation Metrics

The project primarily used accuracy as the performance metric. Other evaluation metrics, such as recall, F1 score, and ROC curve, were underused. These metrics would provide a more comprehensive understanding of the model's performance, especially when dealing with imbalanced datasets.

Lack of Inference Speed and Scalability Testing

While the project achieved good accuracy, there was no testing of the model's inference speed or scalability with large datasets. This is critical for production environments, where real-time prediction speed and scalability matter.

Overall Evaluation

The project successfully completed all key stages of a sentiment analysis pipeline, with notable strengths in model selection, data preprocessing, and deployment readiness. While there are areas for improvement, particularly in hyperparameter optimization, feature engineering, and evaluation metrics, the project is solid and provides a strong foundation for further enhancement.