User Sentiment Analysis in ChatGPT App Reviews Case Study Rubric

DS-4002 - Fall 2024 - Ayush Acharya
Due: Subject to the instructor
Submission Format: Upload link to Github repository on UVA Canvas

Individual Assignment

Why am I doing this?

The goal of this task is to put your data science skills to the test with a real-world dataset. By analyzing reviews from the ChatGPT app, you'll dive into sentiment analysis and topic modeling—two powerful NLP techniques—and see firsthand how they can shed light on user feedback. Imagine yourself as a product analyst tasked with helping OpenAI understand their users better. This project is a chance to work on something meaningful.

What am I going to do?

In this case study, you'll dive into the world of user sentiment and feedback for the ChatGPT app by analyzing reviews across different app versions and countries. Your goal is to identify key themes and trends in user satisfaction that could help OpenAI refine the app based on version-specific insights.

1. **Access the GitHub Repository**: Start by accessing the GitHub repository, where all the necessary datasets, code, and documentation are stored. This repository will guide you through each stage, from data preparation to final analysis.
2. **Gather and Prepare Data**: The dataset of user reviews includes fields such as ratings, comments, thumbs-up counts, app versions, and country information. Using Python, clean and preprocess this data to make it analysis-ready.
3. **Perform Sentiment Analysis**: Use natural language processing (NLP) techniques to classify each review as positive, negative, or neutral. Apply a pre-trained sentiment analysis model and validate its accuracy by comparing it with a manually labeled subset of reviews. The source file (jupyter notebook) already provides an implementation of this using the SentimentIntensityAnalyzer() package from nltk.sentiment.vader. Now utilize the DistilBERT model from the transformers library to carry out the same task and compare the findings from the two models.
4. **Conduct Topic Modeling**: Using Latent Dirichlet Allocation (LDA), identify common themes and topics in user feedback. Convert the text data into numerical format, run the LDA model, and adjust parameters to ensure coherent topic clusters. The source file (jupyter notebook) already provides an implementation of this using the LatentDirichletAllocation() package from sklearn.decomposition. Now utilize the Non-negative Matrix Factorization (NMF) model from sklearn.decomposition to conduct the topic modeling and compare the findings from the two models.
5. **Analyze and Visualize Results**: With your sentiment and topic data prepared, create visualizations (such as bar charts and word clouds) to highlight trends across app versions, revealing insights into user satisfaction and recurring concerns.
6. **Deliver Insights**: Summarize your findings in a clear, concise report with supporting visualizations, and findings across the two models. Offer recommendations for OpenAI on how they might improve user experiences, based on sentiment and topic trends across different app versions.

The entire project can be accessed and executed from the GitHub repository at https://github.com/htb4hv/Case-Study-ChatGPT-Reviews.

Your final deliverables should include:

- GitHub repository with all materials used
- README.md: quick summary of the project conducted along with final deliverables
- Source Code File(s): all the scripts used with well commented code
- REFERENCES.md: A document listing any sources, articles, datasets, or libraries that were essential to your project.
- Final Report: A clear, concise report that showcases your findings, complete with visualizations and actionable insights based on your analysis.

Tips for Success:

- Spend Time with the Data: Take time to explore the dataset and really understand it before diving into analysis. This will make your insights sharper.
- Try Different Approaches: NLP can be nuanced. Experiment with different methods for sentiment analysis and topic modeling to capture the most valuable insights.
- Stay Organized: Keeping your files and code neatly organized will make it easier for others (and yourself!) to follow your work.
- Seek Feedback: Don't hesitate to reach out to peers or the instructor—this is a complex project, and feedback can help improve your approach and outcomes.

How will I know I have succeeded?

If all of the criteria requirements in the rubric below are met, the study can be deemed successful.

Spec Category	Spec Details

Formatting	• Single Github repository (submitted via link on Canvas)
	  ○ Create a new Github repository for this project titled "CaseStudy_ChatGPT" that contains
	  ■ README.md
	  ■ LICENSE.md
	  ■ Source Code File
	  ■ REFERENCES.md
	  ■ Final Report
	  • Generated Visualizations
	  • Findings
	  • Comparison between the implemented & suggested models

README.md	• Concise summary regarding what you've

	produced for this case study. Does not need to be detailed, but rather serve as an abstract to inform people regarding your project.
Source Code File	● Data Exploration & Analysis ● Sentiment Analysis using SentimentIntensityAnalyzer() & DistilBert() ○ Compound Score (Metric) ● Topic Modeling using LatentDirichletAllocation() and Non-negative Matrix Factorization (NMF) model. ● Well commented code throughout.
REFERENCES.md	A markdown file called “REFERENCES.md” that cites all resources utilized for this case study in IEEE Document format.
Final Report	A final report in a PDF format that compiles all of the generated visualizations, data exploratory analysis/finding, and similarities and comparisons on the findings using the provided models (SentimentIntensityAnalyzer() & LatentDirichletAllocation()) and the models to be implemented (Sentiment Analysis using DistilBert from transformers library & Non-negative Matrix Factorization (NMF) from sklearn).