## **ChatGPT Twitter Dataset: Final Report**

Thomas Antony

## **Abstract**

Sentiment analysis uses natural language processing to find the emotions in the text and gain knowledge into public opinion. This project focuses on tweets about ChatGPT, using a Kaggle dataset to understand how users feel. By cleaning and analyzing the data with methods like TF-IDF vectorization and machine learning classification, we sorted tweets into Positive, Neutral, and Negative categories. After trying several models like Logistic Regression, Decision Tree, Random Forest, XGBoost, and Adaboost. Logistic Regression model performed the best, achieving the best balance between simplicity, accuracy, and interpretability.

**Introduction**

ChatGPT has become a big topic on social media, with people having both good and bad opinions on it. While many praise its potential, others show concerns about its use and functionality. The aim of this project is to categorize user opinions into Positive, Neutral, or Negative. After thoroughly preparing the data and testing various models, we uncover how people view this revolutionary AI.

**Data Description**

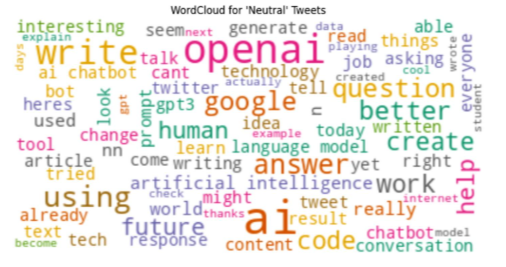
The dataset was sourced from Kaggle and contains tweets from Twitter. The dataset has over 219,000 rows, each featuring user opinions about ChatGPT. The tweets represent a range of sentiments, making the data ideal for this analysis. Alongside the tweets, the dataset provides sentiment labels like Positive, Neutral, or Negative which allowed us to train and evaluate machine learning models for accurate classification.

**Wordcloud**

To understand recurring themes, we created word clouds for each sentiment category. For Positive tweets, words like "innovative" and "useful" stood out, while Negative tweets featured terms such as "bug" and "error." These visualizations quickly reveal the core ideas expressed by users and provide a snapshot of public sentiment.





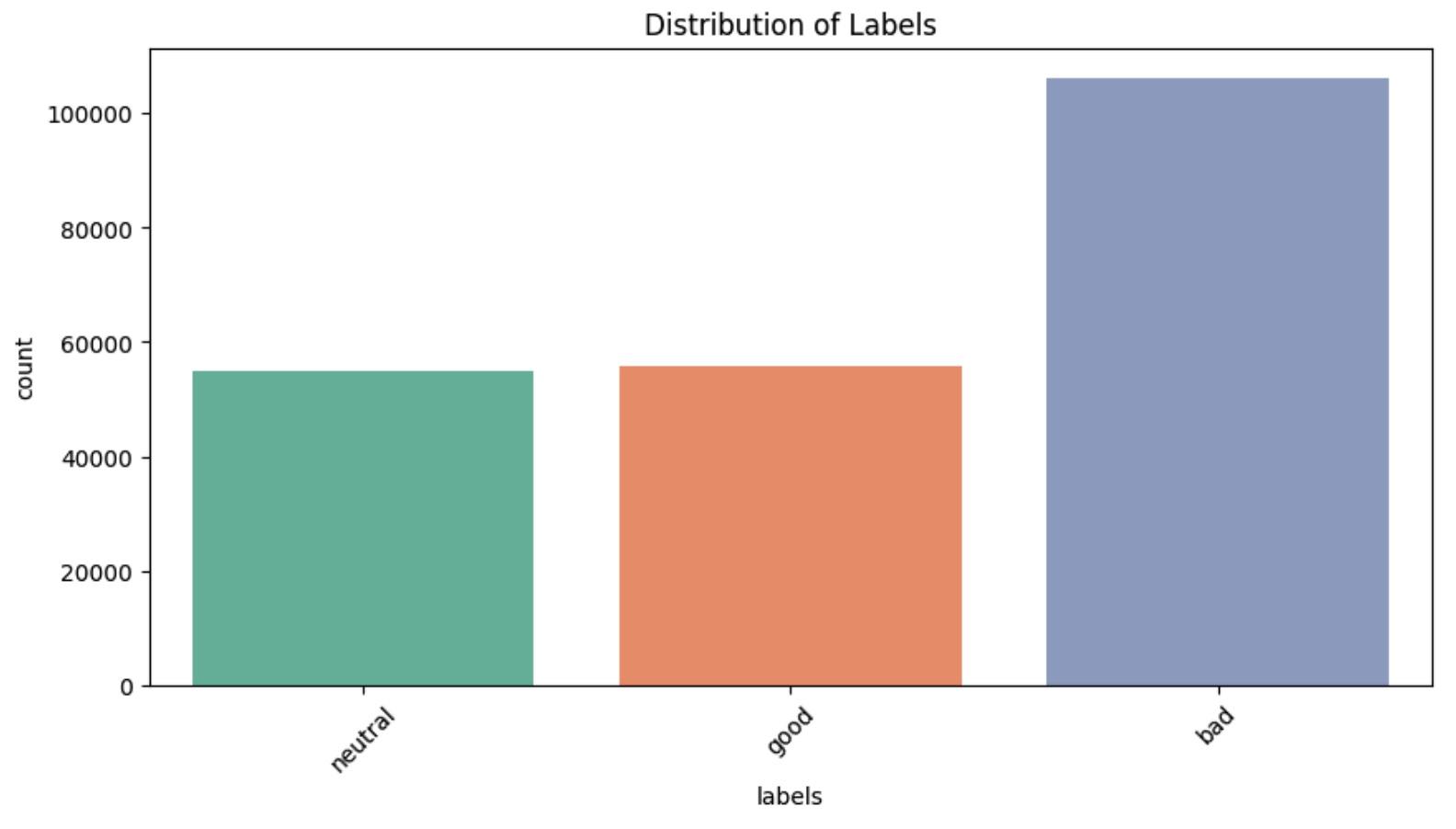


## **Data Preprocessing**

Effective preprocessing was crucial to make sure we got high-quality input to our machine learning models. The preprocessing pipeline is a couple of steps. First, cleaning was performed to remove links, emojis, hashtags, and special characters from the tweets. Next, duplicate tweets were eliminated to avoid redundancy in the dataset. To focus on relevant data, only English-language tweets were retained using a language detection tool. The textual data was then transformed into numerical representations using TF-IDF vectorization, which captured the importance of terms in the context of the dataset. Finally, sentiment labels were encoded into numerical values Negative = 0, Neutral = 1, Positive = 2 to prepare the data for machine learning. These steps ensured the dataset was clean, uniform, and ready for analysis.

## **Exploratory Data Analysis**

Exploratory data analysis showed several important insights about the dataset. The distribution of sentiment labels like Good, Neutral, and Bad was visualized, showcasing a significantly higher number of Bad tweets compared to Good and Neutral ones. This imbalance informed our decisions for downstream processing and modeling strategies. Further exploration using visualizations like word clouds revealed key patterns in the data. Words such as "problem" and "scary" dominated the Negative category, showing that user are frustrated and potentially worried about chatgpt, while Positive tweets often featured terms like "impressive" and "better," reflecting optimism and gratitude towards chatgpt. These insights guided both data preprocessing and the selection of appropriate machine learning models for sentiment classification.



## **Models**

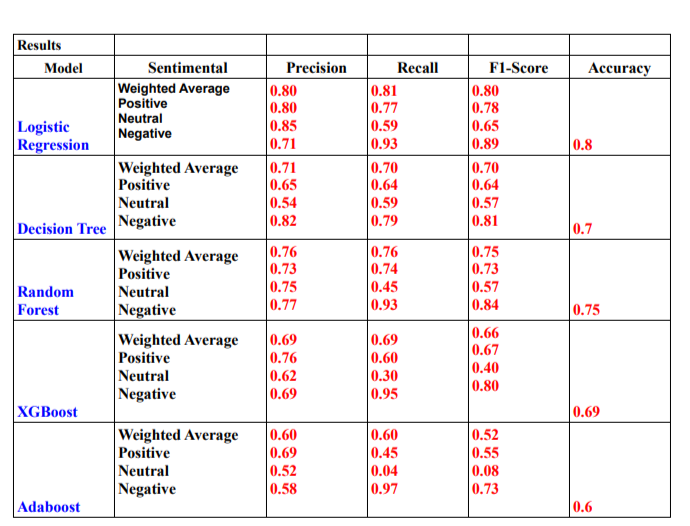
This project implemented and evaluated five machine learning models to classify tweet sentiment. Logistic Regression served as a simple, interpretable baseline model and consistently delivered strong performance. Decision Tree models were explored for their ability to capture non-linear patterns, but they were prone to overfitting without regularization. Random Forest improved upon Decision Tree by combining multiple trees to reduce overfitting and improve accuracy. XGBoost, a powerful gradient boosting algorithm, performed well with complex data but required significant computational resources. Due to its limitations with larger datasets, we randomly selected 100,000 samples from the dataset to train the XGBoost model effectively. Finally, Adaboost focused on improving weak classifiers by emphasizing misclassified samples to improve performance.

## **Evaluation Metrics**

## To evaluate model performance, several metrics were used, including Precision, Recall, F1-Score, and Accuracy. Precision measured how many predicted positive observations were truly positive, while Recall assessed the model’s ability to identify all actual positive observations. The F1-Score, a harmonic mean of Precision and Recall, provided a balanced metric. Finally, Accuracy measured the overall correctness of each model. Logistic Regression excelled across all metrics, making it the best-performing model for this analysis.

## **Results**

The Logistic Regression model outperformed the others, achieving the best balance between simplicity, accuracy, and interpretability. While it serves as a baseline, its competitive performance demonstrates its reliability for sentiment analysis.



## **Discussion**

The analysis highlighted the strengths and weaknesses of each model. Logistic Regression proved to be simple, interpretable, and highly effective, achieving the highest scores across multiple metrics. Decision Tree captured non-linear patterns but struggled with overfitting, which limited its generalizability. Random Forest offered improved robustness compared to Decision Tree, but it wasn’t as consistent as Logistic Regression. XGBoost worked well with complex data but didn’t outperform simpler models and required more computational resources. Adaboost, while an interesting approach, struggled to keep up with the performance of the other models. Ultimately, Logistic Regression demonstrated that straightforward approaches can often outperform more complex methods when applied thoughtfully.

## **Conclusion**

This project demonstrates how machine learning can automate sentiment analysis effectively. By combining robust preprocessing, feature extraction, and model evaluation, we gained valuable insights into how users perceive ChatGPT. Logistic Regression is clearly the best-performing model overall, with the highest accuracy and strong metrics across all categories. Random Forest comes in as a solid second choice, especially excelling in detecting Negative sentiment. The Decision Tree is decent but less consistent, especially for Neutral sentiment. XGBoost and Adaboost, on the other hand, struggle significantly, particularly with Neutral sentiment, making them less reliable for this task. Overall, Logistic Regression stands out as the most dependable model.

## **Future Work**

To extend this study, we suggest:

1. Exploring advanced NLP models like BERT for better contextual understanding.
2. Expanding the dataset to include tweets in multiple languages.
3. Implementing deep learning models such as LSTMs for more nuanced text classification.

### **References**

Kaggle Dataset: https://www.kaggle.com/datasets/charunisa/chatgpt-sentiment-analysis?select=file.csv

Google Colab:https://colab.research.google.com/drive/1FrpgFinyh8cd2kxaBpXpOf2bCHsDhg3n?usp=sharing