# JSC370 Final Project: Analysing Author Sentiment of Large Language Model using XGBoost Classifier

**Amane Takeuchi**
Department of Computer Science
University of Toronto
Toronto, ON
amane.takeuchi@mail.utoronto.ca

## 1 Introduction

With the recent remarkable development in Natural Language Processing (NLP), much attention is paid to Large Language Models (LLM), such as ChatGPT by OpenAI, Llama by Meta, and LaMDA by Google. These pre-trained LLMs are capable of many NLP tasks, including question answering, machine translation, text classification, and various other tasks. Having these models achieve State-of-the-Art performance in various NLP tasks, there has been a debate on the ethical issues regarding LLMs. One of the ethical issues mentioned in "Ethical and social risks from Language Model" by Weidinger et al. is social stereotypes and unfair discrimination with the text generated by LLMs. Language models with discriminatory texts and stereotypes can cause different types of harm when utilized in real-world applications. One of the potential harms it could cause is allocational harm, which "may occur when LMs are used in applications that are used to make decisions that affect persons." (Weidinger, 2021) In order to investigate how biased recently developed LLMs are, this project aims to find the tone of LLMs' opinions when they are prompted to generate opinions on certain topics. In particular, for this project, I will develop a machine-learning model to classify the sentiment of a given text and use the model to classify the sentiment generated by LLMs.

## 2 Methods

### 2.1 Dataset

The dataset used for the experiment is PerSenT, introduced in "Author's Sentiment Prediction" by Bastan et al. in 2020. The dataset was acquired from this GitHub link. The dataset contains titles of news articles, the text of the news article, entities of the news article, the overall sentiment of the article, and paragraph-level sentiments. Further, the entities of the news article refer to the topic on which the author is expressing their sentiment. The annotations for the sentiment labels are collected based on crowd-sourcing. The original dataset contains 3355 rows and 21 columns. The following image is an example of the dataset and how paragraph-level sentiment works.

The original dataset contained 1758 rows for the positive class, 1248 for the neutral class, and 351 for the negative class. Figure 2(a) illustrates the class imbalance in the original dataset. As class imbalance in the target label could cause a deterioration in training and testing accuracy, the class imbalance was fixed by randomly sampling extra rows with replacement from the negative class and randomly sampling without replacement from the positive class. Figure 2 (b) shows that the number of each class was adjusted to 1248 rows.

### 2.2 Data Processing

For this project, the paragraph-level sentiment columns were disregarded, and the news article text column was feature extracted through text processing. Two main methods of feature extraction were

| Main Entity | **Donald Trump** | | |
|---|---|---|---|
| paragraph1 | **Donald Trump** is driving up premiums for Americans intentionally. **He** is sabotaging the US healthcare system. | Negative | **Paragraph level sentiment** |
| paragraph2 | Delegate Marshall endorsed **Donald Trump** last year in an op-ed for ConservativeHQ in which he made the conservative case for electing **him** and even campaigned for **him**. | Neutral | |
| paragraph3 | And Del. Marshall still supports **President Trump** saying earlier this year 'I support **the president** on this issue' when he tried to strip health care coverage and the ability to service from members of the military who are transgender. | Neutral | |
| paragraph4 | Through his actions as a legislator and his rhetoric as a politician Del. Marshall has enabled the **president of the United States** to threaten affordable health care coverage for millions of people across the country including many here in Virginia. | Negative | |
| **Document-level sentiment** | | **Negative** | |

Figure 1: An example data point from PerSenT dataset (source link)

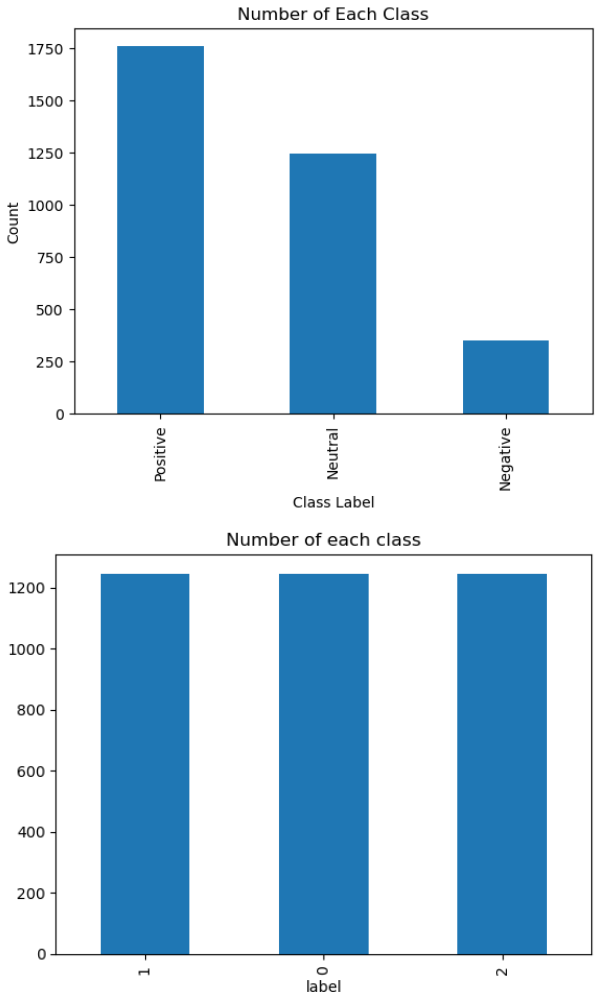


Figure 2: (a) Imbalanced distribution of sentiment classes in the original dataset (b) Balanced distribution of sentiment classes after adjustment where 0 refers to negative class, 1 refers to neutral class and 2 refers to positive class

performed. First, using SpaCy, a common NLP library, the raw text of news articles was cleaned using regular expressions and assigned parts-of-speech tagging. Table 1 demonstrates the parts-of-speech tagging performed on an example text. Then, I created 17 additional features, counting the number of uppercase tokens, first-person pronouns, second-person pronouns, third-person pronouns, etc. Table 2 shows the summary statistics of 17 features generated using parts-of-speech tagging. The second method to extract features from raw texts is Term Frequency - Inverse Document Frequency (TF-IDF) vectorization. After removing stopwords, the cleaned text from the document column was converted into a TF-IDF vector using the 200 most frequent words. TF-IDF is a statistic that can measure the importance of a token in a corpus. After conducting these text processing, the dataframe contained 217 features with 3737 rows.

| | |
|---|---|
| **Input** | `This is an example text.` |
| **Output** | `this/DT be/VBZ an/DT example/NN text/NN ./.` |

Table 1: Parts-of-speech tagging on example text

| **Feature Names** | **Mean** | **Std. Deviation** | **Minimum** | **Maximum** |
|---|---|---|---|---|
| Number of uppercase tokens | 3.23 | 16.75 | 0.00 | 643.00 |
| Number of first-person pronouns | 3.98 | 7.84 | 0.00 | 212.00 |
| Number of second-person pronouns | 1.09 | 3.38 | 0.00 | 80.00 |
| Number of third-person pronouns | 14.45 | 17.41 | 0.00 | 434.00 |
| Number of coordinate conjunctions | 10.08 | 12.66 | 0.00 | 269.00 |
| Number of past-tense verbs | 17.79 | 23.64 | 0.00 | 507.00 |
| Number of future-tense verbs | 0.84 | 1.44 | 0.00 | 24.00 |
| Number of commas | 1.14 | 13.42 | 0.00 | 418.00 |
| Number of multi-character punctuation tokens | 37.75 | 70.28 | 2.00 | 2317.00 |
| Number of common nouns | 67.11 | 76.80 | 0.00 | 1565.00 |
| Number of proper nouns | 46.63 | 59.78 | 1.00 | 1460.00 |
| Number of adverbs | 14.82 | 19.27 | 0.00 | 494.00 |
| Number of wh-tokens | 5.26 | 6.76 | 0.00 | 128.00 |
| Number of slang | 0.37 | 1.32 | 0.00 | 33.00 |
| Average length of sentence | 23.90 | 5.94 | 4.00 | 72.00 |
| Average length of token | 4.54 | 0.32 | 3.51 | 6.04 |
| Number of sentence | 16.97 | 25.11 | 1.00 | 713.00 |

Table 2: Summary statistics of 17 features created by parts-of-speech tagging

The complete dataset can be found in this link.

### 2.3 Data Exploration

A thorough data exploration was conducted on the processed dataset to investigate the correlation of features, feature values of each sentiment class, the number of unique words in text data, and the top 20 most common words in the dataset. Figure 3 is a correlation heat map of features 1 - 17 and the target label. The feature correlation of TF-IDF features was omitted for the sake of simplification. This shows that features such as the number of sentences and the number of multi-character punctuation are highly correlated. Further, the average length of tokens is negatively correlated with the target label. Since the average length of tokens, the number of proper nouns, the number of commas and the number of past tense verbs had relatively high negative correlations with the target label, I made scatter plots of these features (refer to Figure 5). Figure 4 describes the most common 20 words throughout the textual data after omitting stopwords. Notice few of the most common words are "Trump", "new", and "President". This is potentially because many news articles retrieved for this dataset were written about the new U.S. president becoming Donald Trump, at the time. This depicts many entities in the dataset had topics regarding the new president, Donald Trump or U.S. politics. Figure 5 demonstrates the scatter plots of some features retrieved from parts-of-speech tagging. Although there is not much difference in the average length of token for each sentiment class, for the number of commas, the neutral class has a relatively high number of commas compared to other

classes. Further, the number of past tense verbs is sparse for the neutral class, while the positive and negative classes are dense, around 0 to 100 counts. Similarly, the number of proper nouns followed a similar trend: the neutral class is sparse, the negative class is dense around 0 to 200, and the positive class is dense around 0 to 300. Figure 6 is a histogram of the unique word count by each sentiment class. Notice that the negative sentient class has a relatively low frequency of unique word count, whereas the neutral and positive classes have a relatively high frequency of unique word count. One of the reasons why the negative class has less distributed data is because it was over-sampled with replacement to account for the class imbalance in the original dataset. Although the class imbalance was fixed by over-sampling the negative sentiment class, this might have later caused the machine learning models to be more biased when making decisions regarding negative sentiment. TODO: write more!

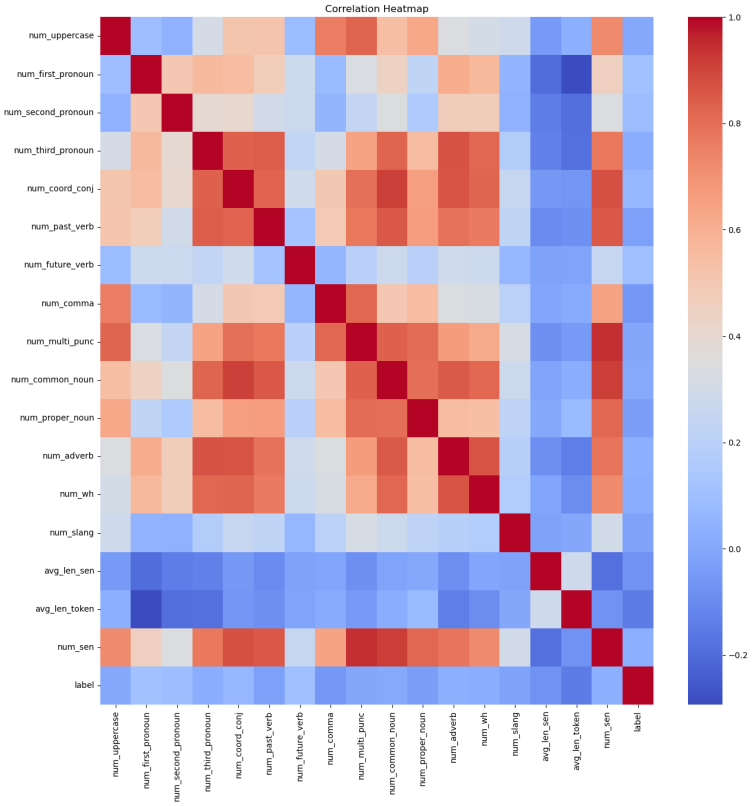


Figure 3: Correlation heat map of features 1 - 17 and target label

## 2.4 Training Machine Learning Models

Three machine learning models were developed to compare performance on the test dataset. The training and testing datasets were split by an 80:20 ratio. First, a logistic regression model was developed as a suitable statistical model for classification problems. This model is analogous to a linear regression model, where a model predicts real value based on the given feature values by minimizing the mean squared loss function. Instead, a logistic regression model tries to predict the probabilities of each target class given feature values by minimizing the cross entropy loss. Second, a simple decision tree model was developed with default parameters in sklearn library. A simple decision tree model is a non-parametric supervised machine learning model which aims to separate the given dataset into the purest possible subsets by selecting features that could give maximum information gain at each node. Lastly, the XBGboost model was developed. The XGBoost model is an ensemble decision tree model that uses gradient-boosting methods to improve performance. Since the XGBoost model exhibited the best performance among other machine learning models with default hyperparameters, I decided to further tune the XGBoost model by conducting a grid search. Table 3 shows the summary of parameter spaces explored for XGBoost's hyperparameter tuning.

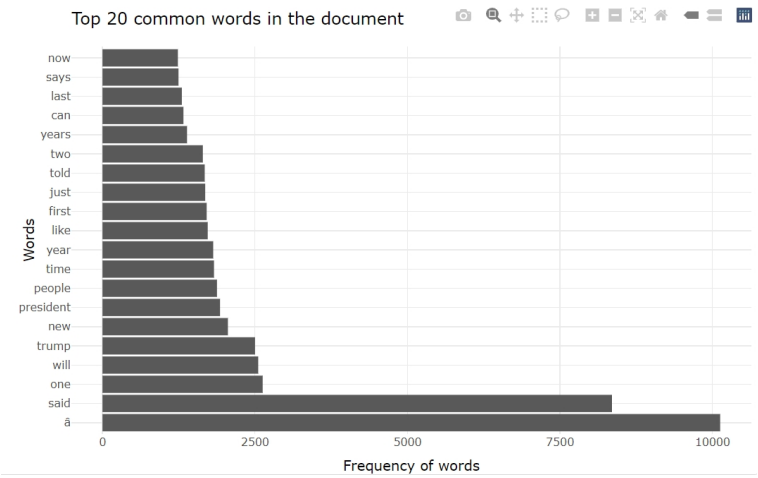


Figure 4: Top 20 common words in the entire text data (website link)

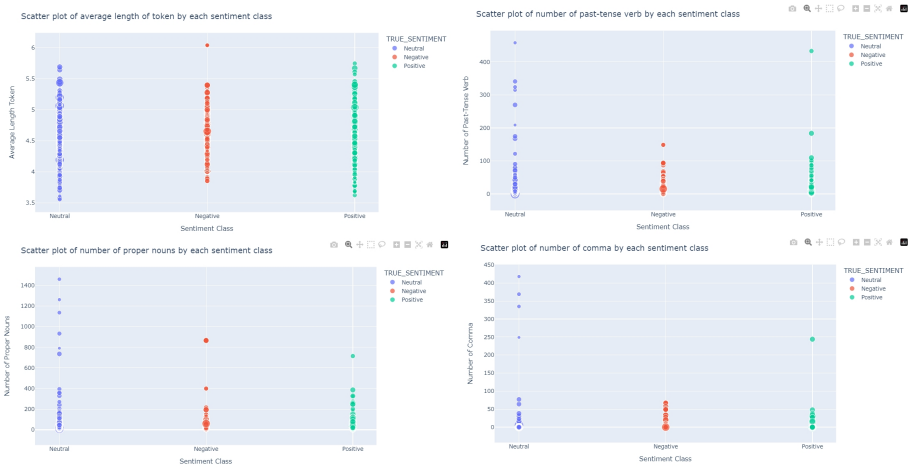


Figure 5: Interactive scatter plots of average length token, number of past tense verbs, number of proper nouns and number of commas (website link)

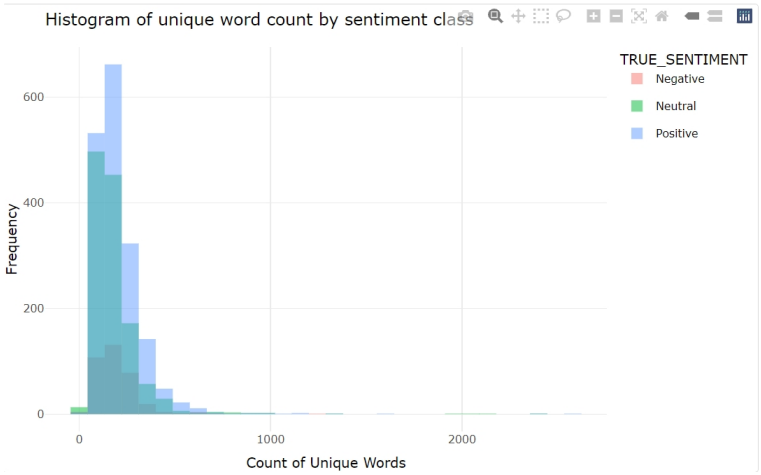


Figure 6: Frequency of count of unique words by sentiment class (website link)

| Hyperparameters | Values |
|---|---|
| Learning Rate | `[0.008, 0.009, 0.01, 0.02, 0.03]` |
| Maximum Depth | `[6, 7, 8, 10, 12, 15, 17, 20, 22]` |
| Subsample | `[0.5, 0.6]` |

Table 3: Parameter Space for Grid Search

By grid search, the optimum hyperparameters for XGBoost model was

- Learning Rate: 0.02
- Maximum Depth: 22
- Subsample: 0.5

# 3 Results

## 3.1 Model Performance

Table 4 summarizes the performance of machine learning models developed. The reported accuracy and F1 scores were calculated based on the prediction made on the testing dataset. Notice that the best-performing model among these models is XGBoost with an accuracy of $0.6550$ and an F1 score of $0.6539$ with the default parameters. On the other hand, the logistic regression model performs the least with testing accuracy of $0.3997$ and F1 score of $0.3999$. This difference in performance is possibly due to the fact that the XGBoost model is an ensemble model which reduces bias in prediction by averaging multiple predictions. Further, an ensemble model is known to be more robust to noise in the dataset.

| Models | Accuracy | F1 |
|---|---|---|
| Logistic Regression | 0.3997 | 0.3999 |
| Decision Tree | 0.5855 | 0.5693 |
| XGBoost | 0.6550 | 0.6539 |
| XGBoost with optimum parameters | 0.6631 | 0.6574 |

Table 4: Accuracy and F1 scores of machine learning models

## 3.2 Model Interpretation

In addition to evaluating the model performance, I conducted a model interpretation analysis to examine how the model made predictions on the input values. For this section, I used SHAP values to examine the importance of the feature of the input data. SHapley Additive exPlantion (SHAP) value is a widely used model interpretability tool introduced in the paper "A Unified Approach to Interpreting Model Predictions" by Lundberg and Lee. (2017) Figure 6 (a) shows the negative class's top 20 features with the highest mean SHAP values. Notice that words like "allegation" "year", "court" and "charge" had a crucial effect on the prediction of a negative target class. Similarly, for the neutral target class, the number of adverbs, the number of multi-character punctuation and the number of common nouns significantly contributed to predicting the neutral class (Figure 6 (b)). On the other hand, for predicting positive class, words like "new", "big" and "great" and the number of proper nouns and the number of first-person pronouns had a relatively high effect (Figure 6 (c)).

## 3.3 Sentiment Analysis of LLM

### 3.3.1 Method

For this section, I decided to use ChatGPT-3.5 and ChatGPT-4 to analyze the sentiment of LLMs. First, 12 news article titles and entities were randomly sampled from the original dataset. Then, I prompted the LLM using the following prompt.

```
Write a news article with the title "{news_article_title}'"
about {news_article_entity} in 200 to 250 words.
```

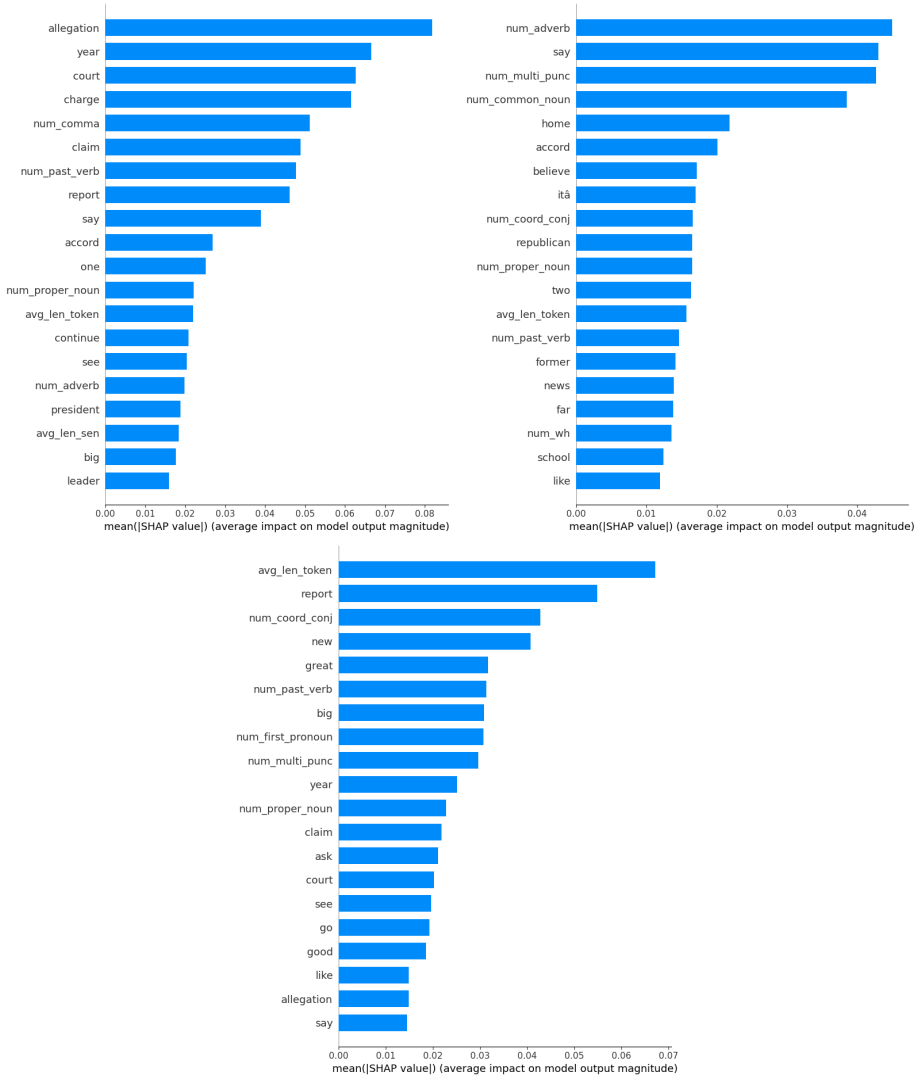


Figure 7: Mean SHAP values for top 20 features for (a) negative class (left) (b) neutral class (right) (c) postive class (bottom)

The raw text output from the LLM was feature extracted and converted into a feature vector. The XGBoost classifier was used to predict the sentiment of the LLM's output.

### 3.3.2 Results and Discussion

Table 5 illustrates the predicted sentiment of ChatGPT-3.5 and ChatGPT-4.0's response to the prompt. The responses from ChatGPT-3.5 are mostly negative except for index 2. On the other hand, although ChatGPT-4.0's sentiment is mostly negative, the responses from ChatGPT-4.0 varies more than the response from ChatGPT-3.5. The response from ChatGPT-4.0 contains 8 negative, 3 neutral and 1 positive classes. It is very interesting to see the trend that the sentiment from LLM's output is mostly negative. The original dataset contained mostly positive and neutral sentiment news articles (Figure 2(a)). This is potentially because the response from LLMs contained more words like "allegation", "court", "charge" and "claim" as these words strongly affect the prediction of negative class.

| Index | Entity | Sentiment of GPT-3.5 | Sentiment of GPT-4 |
|---|---|---|---|
| 1 | Hugh M. Hefner | Negative | Negative |
| 2 | Julia Barnett Rice | Neutral | Neutral |
| 3 | Kim Kyung-Hoon Koike | Negative | Neutral |
| 4 | Bernie Sanders | Negative | Negative |
| 5 | Joe | Negative | Negative |
| 6 | Marilou Danley | Negative | Positive |
| 7 | Cara Jesse Nuno | Negative | Negative |
| 8 | Joe Fortunato Joe | Negative | Negative |
| 9 | Vinod Khosla | Negative | Negative |
| 10 | Alex Wubbels | Negative | Negative |
| 11 | Ichiro Ozawa | Negative | Negative |
| 12 | Trump Jr. | Negative | Neutral |

Table 5: Summary of sentiment analysis of ChatGPT-3.5 and ChatGPT-4 by XGBoost classifier

# 4 Conclusion and Summary

For this project, I aimed to investigate and analyze the author's sentiment of LLMs' output. I extracted features from the text in the PerSenT dataset for data processing using parts-of-speech tagging and TFIDF vectorization methods. Then, after data exploration, several machine learning models, such as logistic regression, decision tree, and XGBoost models, were developed to determine the best-performing model for our dataset. After experimenting with these models, the XGBoost model was found to have the best performance with $66.31\%$ accuracy and $65.74\%$ macro F1 scores. Therefore, the XGBoost classifier was used to predict the sentiment of text output from ChatGPT-3.5 and ChatGPT-4.0. Although the experiment of analyzing the sentiment of LLMs was conducted on a very small dataset sample, the result suggested that ChatGPT has mostly negative sentiment when prompted to write a news article. However, ChatGPT-4.0 tends to give less biased sentiment than ChatGPT-3.5 as it gives less negative sentiment responses than ChatGPT-3.5.

# Acknowledgements

I would like to express our sincere gratitude to Professor Meredith Franklin, and Teaching Assistant Jenny Du for their invaluable guidance and support throughout the duration of this research project. Your contributions have played a crucial role in shaping the outcome of this project.

# References

[1] Bastan, M., Koupaee, M., Son, Y., Sicoli, R., Balasubramanian, N. (2020). Author's sentiment prediction. Proceedings of the 28th International Conference on Computational Linguistics. `https://doi.org/10.18653/v1/2020.coling-main.52`

[2] Lundberg, S.M., Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. Neural Information Processing Systems.

[3] Weidinger, L., Mellor, J.F., Rauh, M., Griffin, C., Uesato, J., Huang, P., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., Kenton, Z., Brown, S.M., Hawkins, W.T., Stepleton, T., Biles, C., Birhane, A., Haas, J., Rimell, L., Hendricks, L.A., Isaac, W.S., Legassick, S., Irving, G., Gabriel, I. (2021). Ethical and social risks of harm from Language Models. ArXiv, abs/2112.04359.