Social-Media-Text Emotion Analysis using Machine Learning

Karan Reddy Kandala
karanreddyk@vt.edu
Virginia Tech

Abstract

This project aims to develop a system for emotion classification in Social-Media-Text using Natural Language Processing (NLP) and machine learning. The system leverages sentiment analysis to assign emotions such as positive, negative, or neutral to input text, with features extracted through techniques like TF-IDF. By addressing challenges in current practices, such as contextual misunderstanding and dataset biases, this project aspires to improve accuracy and scalability in emotion detection.

1. Introduction

The project [2] analyzes and classifies emotions from social-media text. It attempts to understand how people feel and react through text. The specific problem is to make a reliable, fast and automatic process for texts. Currently, emotion analysis is a matter of polling and seeking social media trends, such as hashtags and likes. However, these approaches are limited and face challenges, they have bias in training datasets, misinformed (polls misinterpreting and improper scaled, to observe trends), filtering bias, to name a few. I've tried to make this process easier through Natural language processing[6] and machine learning, for large datasets, which could be used in real time.

Individuals, organizations, and governments, everyone would be impacted. People control the world, and everyone wants to know the public opinion. The implications of success could mean predicting mass emotions, trends, cycles, economic and political, and consumer preferences. The list could go on. Businesses could gain insights into consumer trends, could gather public perspective on world events. To make better recommendations, and improve consumer lives.

Paper [1] uses multiple labels to identify emotions and classify, however this is a tedious process regardless of the types of optimizations. You cannot have two conflicting emotions in a text. To differentiate from this approach this project will aim to classify into a specific label, then determines an utmost emotion.

2. Methodology

I developed a text emotional analysis pipeline, using various packages (TextBlob/Sklearn) and machine learning and NLTK (Natural Language processing Tool Kit). To be able to process data and reach a valid conclusion, a series of processes were performed.

2.1. Data Pre-processing

The raw data was downloaded from Kaggle, a reliable source for large scale data. The data is classified into Text and a Rating (sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5)). The text was extracted from various social media posts.

Case Conversion: All text was converted to lowercase to ensure uniformity. Filtering and Lemmatization: Non-alphabetic symbols, stop words, and redundant words were removed using NLTK. Lemmatization was performed to reduce words to their base forms Feature Extraction and Classification

2.2. Feature Extraction and Classification

TF-IDF Vectorization transformed text into a numerical format by assigning importance scores to terms, ignoring irrelevant filler words. Vectoring and training the data were done using Sklearn[11], a package in python. Training was done on a Logistic Regression Model [4]. This technique is known for its simplicity and effectiveness in text classification. This was the first analysis.

A second type of classification was using TextBlob[10], an open-source library for textual processing. This analysis was to understand Sentiment Polarity, and Subjectivity. Polarity measures sentiment from -1 (negative) to +1 (positive), while subjectivity measures subjectivity from 0 (objective) to +1 (subjective).

2.3. Validation and Evaluation Metrics

To evaluate the performance of emotion classification model, the dataset was split into 80% training and 20% testing. To ensure randomness, the split was done using a random state (shuffler). Four key evaluation metrics were established: accuracy, precision, recall and finally F1-score.

Accuracy measures the proportion of correctly predicted labels, while precision is used to calculate how well positive instances have been validated. Recall quantifies the model's effectiveness in identifying specific emotions. The F1 score provides a harmonic mean of precision and recall. To calculate these metrics Sklearn imports were used.

Matplotlib graphs will also be used to understand the relationship between sentiment polarity and subjective. To evaluate the data set and its contents, a distribution against sentiment polarity will be used. A class distribution graph will also be used to explore the training sets to understand implications on the ML algorithm.

The process and the algorithm could be successful if it reached a classification rate of 80% accuracy. The two processes, one using TextBlob and the other using Sklearn, could provide interesting results due to their varied levels of text processing and sentiment analysis, which can be crucial.

3. Running and Results

The program was run on Google Colab, initially the data is preprocessed. This phase is also known as lemmatizing, where unnecessary words and filler phrases are removed, and the text is converted to its layman terms.

```
0                                           text
0          i just feel really helpless and heavy hearted
1  ive enjoyed being able to slouch about relax a...
2  i gave up my internship with the dmrg and am f...
3                             i dont know i feel so lost
4  i am a kindergarten teacher and i am thoroughl...
```

Figure 1: This is a sample text without preprocessing.

```
                                      cleaned_text
0                  feel really helpless heavy hearted
1  ive enjoyed able slouch relax unwind frankly n...
2             gave internship dmrg feeling distraught
3                                 dont know feel lost
4  kindergarten teacher thoroughly weary job take...
```

Figure 2: Sample text from figure 1 after preprocessing.

3.1. Sklearn Processing

The first run involved processing using Sklearn[11]. The data was then split and trained using a simple logistic regression model, by transforming the cleaned text into numerical features using TF-IDF vectorization with a maximum of 1000 features.:

- Accuracy: 0.8674695904608815
- Precision: 0.8675571796300415
- Recall: 0.8674695904608815
- F1: 0.8663919081909444

These results demonstrate the effectiveness of the logistic regression model in classifying emotions from social media text. The high accuracy indicates that the model is quite reliable in identifying emotions such as joy, sadness, and surprise from the text provided ~86.7% of the time. The precision of ~86.8% indicates that when the model predicts a specific emotion, it is accurate most of the time. Recall of ~86.7% means the model was effective at detecting the emotions across the dataset, although it could still miss some instances. These results show that the logistic regression model is effective at classifying emotions, but there may be room for improvement, especially with complex or subtle emotions that might not be well-represented in the training data.

3.2. TextBlob Sentiment Analysis

To improve classification[10], Polarity and Subjectivity were used in conjunction with the logistic regression model. Sentiment polarity and Sentiment subjectivity are combined with the TF-IDF features extract from text data, this is done using a hstack (horizontal stack), merging the sparse TF-IDF matrix with the dense sentiment matrix into a singular, containing all the features. The benefits of this could be better regression predictions, when running with additional contextual data. The result after the model is run are:

- Accuracy: 0.8994865766176435
- Precision: 0.8984933704723532
- Recall: 0.8994865766176435
- F1: 0.8988573042133501

The model is performing quite well with an accuracy of around ~90%, and the precision and recall are very close to each other, indicating that the model is effectively distinguishing between the different sentiments or emotion classes. The added layer of sentiment matrix may have resulted in ~3% better performance in the logistical regression model compared to the Sklearn processing.

3.3. Sentiment Distribution Plots

The distribution plots are created to understand the skew of data.

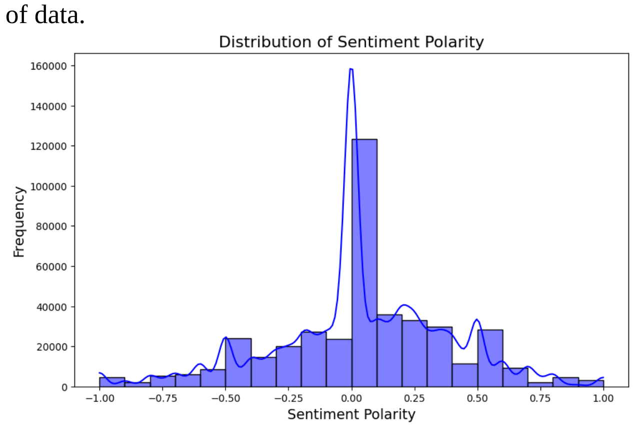

Figure 3: Distribution of Sentiment Polarity in trained data.

This kernel density plot in figure 3, This visualization helps in understanding the overall sentiment of the dataset. By plotting the distribution, you can see whether most of the data is positive, negative, or neutral, which gives an early insight into the general mood conveyed by the text. The histogram groups sentiment polarity values into discrete bins and plots their frequency, helping observe: The central tendency (where most of the data points fall), The spread (how widely the sentiment polarity values are distributed). The shape of the distribution (whether it's skewed to one side, indicating more positive or negative sentiments).

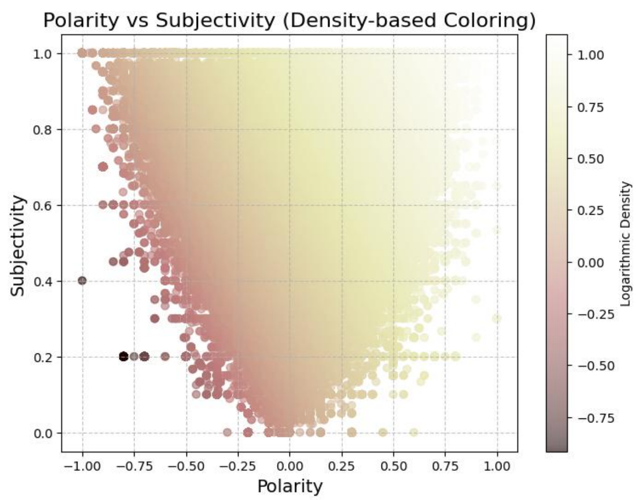

Figure 4: Distribution of Sentiment Polarity in trained data. Polarity from -1 (negative) to +1 (positive), subjectivity from 0 (objective) to +1 (subjective).

Figure 4 plots the relationship between two continuous variables: sentiment polarity and sentiment subjectivity. Each point represents a pair of values from these two features, with: polarity on x axis and subjectivity on y. Scatter plots are useful for detecting correlations or patterns between two variables, helping you understand how they vary together. Here, you can look for any linear or non-linear relationships between the two sentiment measures. The third scale is the density as a function of color, the redder the color the more concentration of points and vice versa.

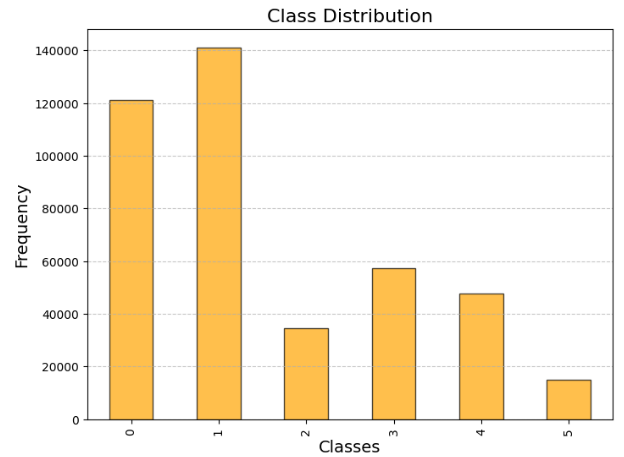

Figure 5: Distribution of classes in raw data. (sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5)).

Figure 5 shows the distribution of data in the raw format. This is the raw classified data that has not been altered and shows the distribution of different emotions in text that was manually classified. This bar graph can provide insight as to why certain classes may be underrepresented or why the algorithm may have performed better in certain scenarios.

4. Evaluation and Analysis

The methodology using traditional logistical regression algorithm results in ~86% accuracy, this value fits the 80% range that was established as part of the success criteria. Although this method is great and provides good accuracy it was a challenge to determine the amount of text filtering. By lemmatizing the text, it may sometimes result in too much or too little text left to analyze. I anticipate that filtering the text further would not result in better accuracy, rather it would take longer to filter them than to train the model itself. This is already an issue with this implementation, filtering the words makes the program run much longer.

The second method involves the conjunction of TF-IDF along with TextBlob sentiment polarity and subjectivity, using a hstack. This procedure seeks to utilize sentimental polarity and subjectivity to improve the accuracy and other stats of the prediction model [2]. This resulted in an ~89% accuracy, this suggests that sentiment polarity and subjectivity act as good descriptors in classifying textual emotions. Thereby this is a successful conclusion. However, there could be a case where the difference between TextBlob and Sklearn could have been more influential rather than just the addition of sentiment layers. This failed to account for before the project completion, Thie also clarifies that the model did not overfit the data as accuracy is in a valid range between 80-90%.

4.1. Graphical Analysis

Figure 3 provides a graphical view of the datapoints with respect to sentiment polarity. To explore this further it is important to understand what this really means. Polarity measures the good and bad in text, the feel of it. -1 to 1, with negative being worse and positive being great/happy. This shows the skew in data and how it corresponds to neutral emotions often. In reality it might not be wrong, people often share updates or just posts that are much more neutral or positive in nature, this has to be a result of why social media even exists, to portray a version of ourselves, although debatable. This analysis shows that social texts are often neutral or positive in nature. The spike at 0.00-0.25 is critical.

In figure 4, another interesting perspective of data and the correlation can be observed, texts that are negative are often highly objective, or loose or not context dependent. This can be seen on the Polarity x-scale, where values below 0 are often dense and subjective in the y-scale. Although the same pattern can be observed in extremes of either side, this begs the question of whether the data is truly proportional and reliable.

This leads to the final graph, the classification based on the raw data classifications into bins of emotions. The bar graph can be observed in Figure 5 to be extremely skewed towards x - 0 and 1, sadness and joy. This forms the first basis of argument against the reliability of data. The other emotions are minute compared. This could have been solved with a dataset that has better, more balanced data.

The main relationship between the data used for training and the rate of classification could have been impacted due to this inherent and unexpected bias in dataset. Therefore, questioning the validity of the results obtained, although an accuracy of 85% has been reached, it begs the question if the classifications were right indeed. The algorithm could have been trained or figured out that repeatedly responding with joy and sadness provided better results on a larger scale, when a valid prediction could not be reached.

5. Reproducibility

The data and training packages used throughout this program is all open sources. The model is trained and accessible through my model. The model parameters are also fully reproducible, although certain adjustments may be required based on the dataset[3] and the way that data is classified. The code for this program is available on GitHub.[2]

The flow of the program is provided as follows:

1. Data Collection
    a. Load the dataset containing text and labels.
2. Data Preprocessing
    a. Remove stop words
    b. Tokenize and lemmatize text.
    c. Extract sentiment features (polarity and subjectivity) using TextBlob.
3. Feature Engineering
    a. Convert text data into numerical representations using TF-IDF.
    b. Combine TF-IDF features with sentimental features.
4. Dataset Splitting
    a. Split the data into training and testing sets.
    b. Model Development
5. Train a logistic regression model on the training set.
    a. Model Evaluation
    b. Predict on the testing set.
6. Compute performance metrics: accuracy, precision, recall, F1-score.
7. Visualization
    a. Plot graphs for sentiment polarity distribution.
    b. Scatter plot of polarity vs. subjectivity with density-based coloring.
8. Model and Artifacts Saving
    a. Save the trained model and the TF-IDF vectorizer for future use

6. Conclusion

The project was mildly successful in developing a text emotion classifier[2], using Natural Language Processing and ML techniques. By leveraging TF-IDF for feature extraction and logistical regression for classification, in conjunction with TextBlob for sentiment analysis, this project achieved notable results. The integration of sentiment polarity and sentiment objectivity features improved the accuracy levels to ~90% from ~87%.

The graphical analysis provided a deeper understanding of the data’s sentiment polarity and its relation to objectivity. Graphs also helped us understand the data and the inherent bias in the data that could have resulted in enhanced results.

Despite achieving the project’s objectives, the results reveal areas for improvement. The model’s effectiveness is affected by the bias in the dataset, and more balanced data could yield better generalization. It serves as a foundation for further research and development in real-time emotion analysis systems, with implications for businesses, governments, and individuals seeking to understand public sentiment. The reproducibility of the project ensures it can be expanded and adapted for more complex applications, paving the way for future advancements in sentiment analysis and emotion classification.

7. References

[1] Liu, X., Shi, T., Zhou, G. *et al.* Emotion classification for short texts: an improved multi-label method. *Humanit Soc*

Sci Commun **10**, 306 (2023). https://doi.org/10.1057/s41599-023-01816-6
[2] My Code: https://github.com/karanreddyk/ML
[3] ⚡, N. E. (2024, February 5). *Emotions*. Kaggle. https://www.kaggle.com/datasets/nelgiriyewithana/emotions
[4] CS 4824/ECE 4424 Project. (n.d.). https://people.cs.vt.edu/dbhattacharya/courses/cs4824/project.html#project-proposal-4-sentiment-analysis-using-machine-learning
[5] Bharti SK, Varadhaganapathy S, Gupta RK, Shukla PK, Bouye M, Hingaa SK, Mahmoud A. Text-Based Emotion Recognition Using Deep Learning Approach. Comput Intell Neurosci. 2022 Aug 23;2022:2645381. doi: 10.1155/2022/2645381. PMID: 36052029; PMCID: PMC9427219.
[6] *View of Natural Language Processing (NLP) for sentiment analysis in social media*. (n.d.). https://carijournals.org/journals/index.php/IJCE/article/view/2135/2523
[7] *Suvrat. (2024, October 7). Sentiment analysis using Python. Analytics Vidhya.* https://www.analyticsvidhya.com/blog/2022/07/sentiment-analysis-using-python/
[8] Machová, K., Szabóova, M., Paralič, J., & Mičko, J. (2023, September 20). Detection of emotion by Text Analysis Using Machine Learning. Frontiers in psychology. https://pmc.ncbi.nlm.nih.gov/articles/PMC10548207/
[9] 1. Language Processing and python. (n.d.). https://www.nltk.org/book/ch01.html
[10] *Tutorial: Quickstart*¶. Tutorial: Quickstart - TextBlob 0.18.0.post0 documentation. (n.d.). https://textblob.readthedocs.io/en/dev/quickstart.html
[11] *User guide*. scikit. (n.d.). https://scikit-learn.org/stable/user_guide.html