Avichai mazin: 211780267
Orel mazin: 213435902

Project In GitHub: Emotion_detection

Emotion detaction classification

***Background*:** Emotion detection from text is a key task in NLP with applications in fields like mental health, customer service, and social media analysis. Recognizing emotions in short texts, such as tweets, is especially challenging due to informal language and limited context.

Our work relates to efforts like the GoEmotions dataset, which highlights the complexity of capturing subtle emotional cues in text.

Our Research Question: *Which modeling approach best captures emotion in short form social media text?*

This project focuses on **emotion classification** in short texts from Twitter. Our goal is to compare different modeling approaches and determine which one best captures emotional content in brief, informal text.

We evaluate three categories of models:

- **Classical machine learning models**: Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, Support Vector Machines (SVM), XGBoost, and AdaBoost.
- **Deep learning**: Convolutional Neural Network (CNN).
- **Pre-trained language models**: fine-tuned BERT (added in the final section).

Dataset Description

We used a publicly available dataset from kaggle of English-language tweets from twitter labeled with six basic emotions:

sadness (0), joy (1), love (2), anger (3), fear (4), surprise (5).

The dataset includes 416,809 samples, each consisting of a short tweet and its corresponding emotion label. All labels are manually annotated and verified through crowd-sourcing. Each data point includes the following fields:

- `text`: the tweet content (string)
- `label`: the emotion class (integer 0–5)

There are no missing values in the dataset.

Class Distribution:

A clear class imbalance was firstly exists: joy and sadness are overrepresented, while surprise and love are rare, so we changed it:

Number of samples per class (before and after downsampling)

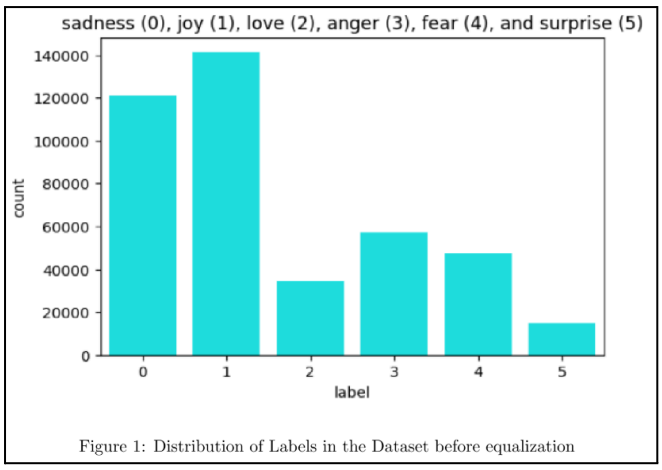
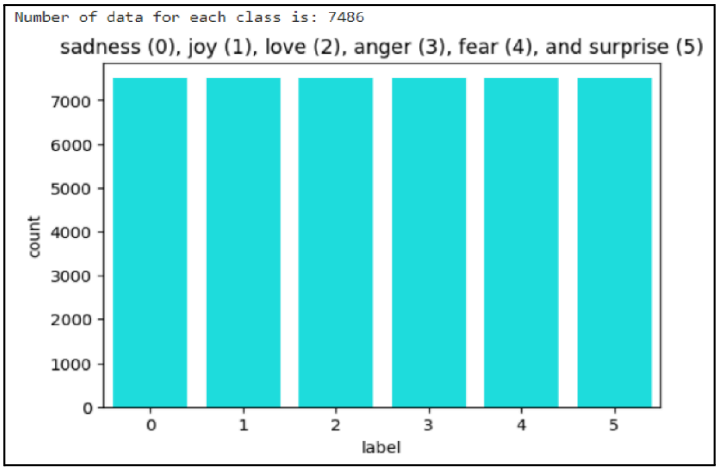


Figure 1: Distribution of Labels in the Dataset before equalization



To draw meaningful and reliable conclusions from the models performance, it was essential to deeply understand the structure and characteristics of the dataset.
Therefore, a comprehensive Exploratory Data Analysis (EDA) was conducted prior to modeling, allowing for informed choices regarding preprocessing, model selection, and evaluation.

EDA (Exploratory Data Analysis) and Preprocessing

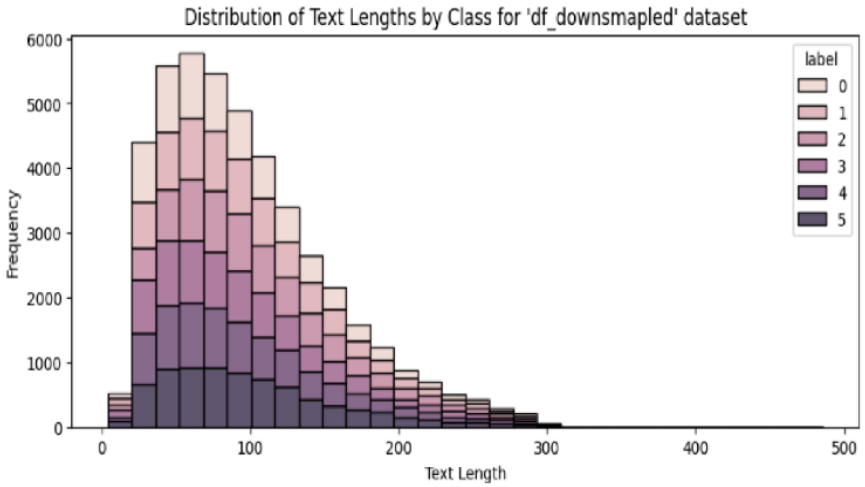
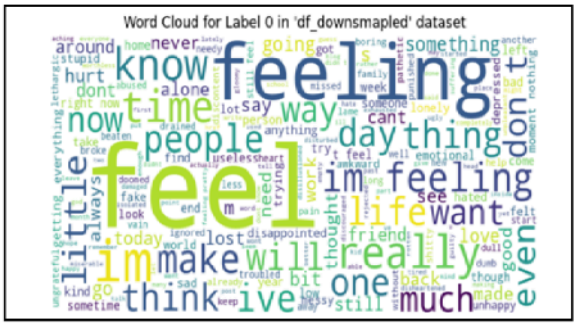


Figure 1: Tweet length distributions by emotion class. Most tweets are short (20–120 words), and all classes show a similar pattern.



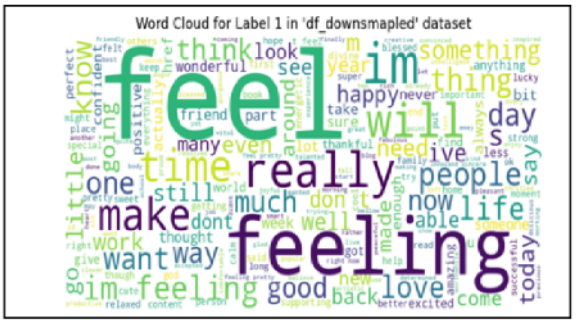


Figure 2: Most frequent words in each emotion class. Words like *feel* really *feeling* appear a lot, meaning people often say how they feel in tweets.

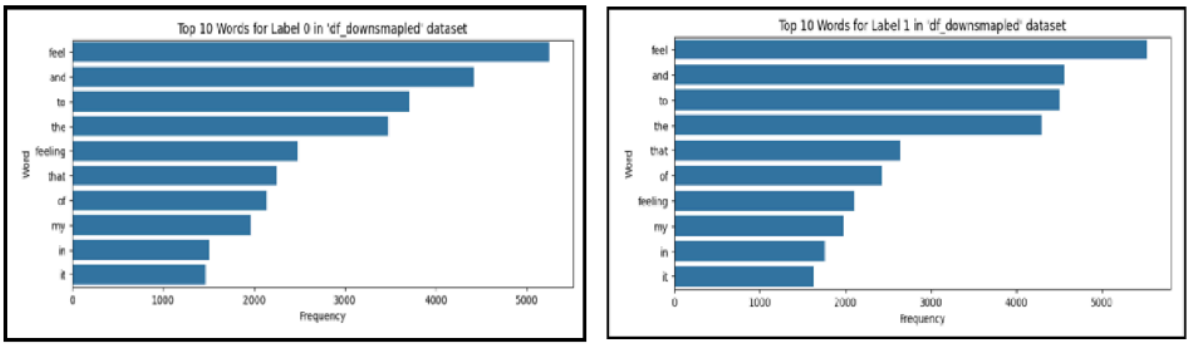


Figure 3: Bar charts showing the 10 most frequent words in each class. Many of these words (e.g., *feel*, *to*, *the*) appear in all classes and are not emotion-specific.

```
custom_stopwords = {"i","im","like", "feel", "feeling", "my", "the", "to", "still"
                    "for","know","littl","think","time","thing","would","go",
                    "really","feel","am","so","get", "one", "to", "and", "at", "can",
                    "day","way", "make", "me", "want", "could" , "would", "tri", "u", "href", "http", "www", "com", "https"
                    }
```

Figure 4: A list of common words that were removed because they don't help the model detect emotions.

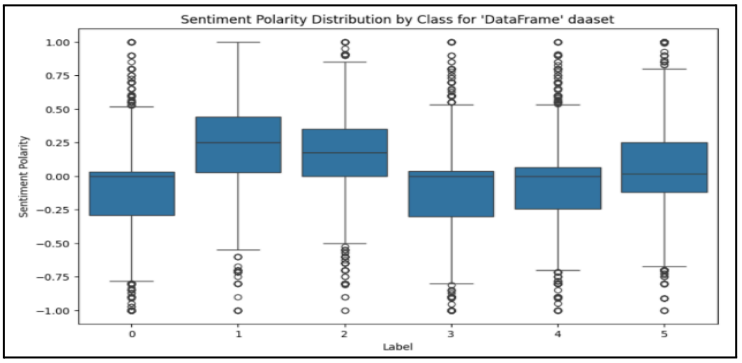


Figure 5: Boxplot showing the sentiment polarity for each emotion class. Positive emotions (e.g., joy, love) have higher median values, while negative emotions (e.g., sadness, anger, fear) have lower values. The dots show extreme sentiment scores (outliers).

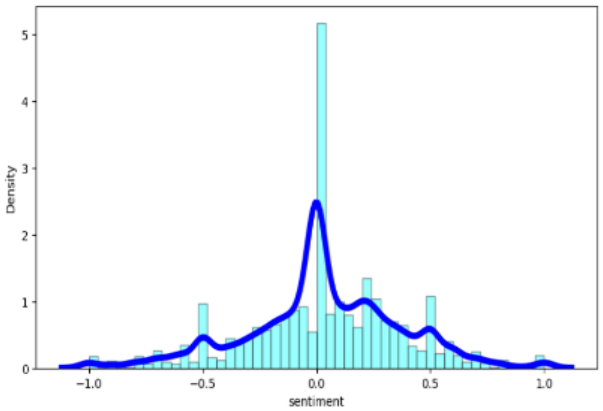


Figure 6: Sentiment values are mostly close to zero, meaning tweets are generally neutral. The histogram shows how often each sentiment score appears, and the blue line shows how they are spread out.

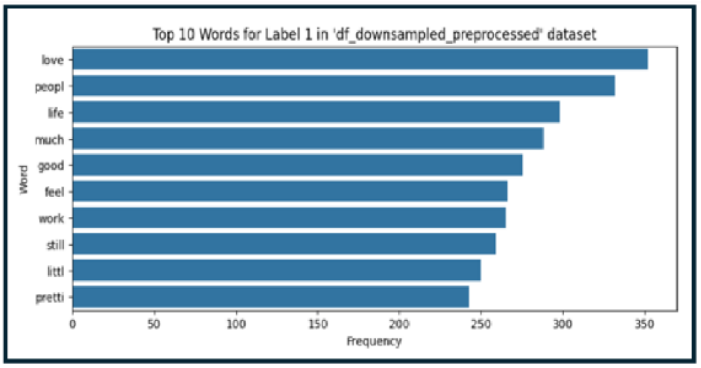


Figure 7: After cleaning the data, the most frequent words in each class are more related to the emotion—like *love* in joy.

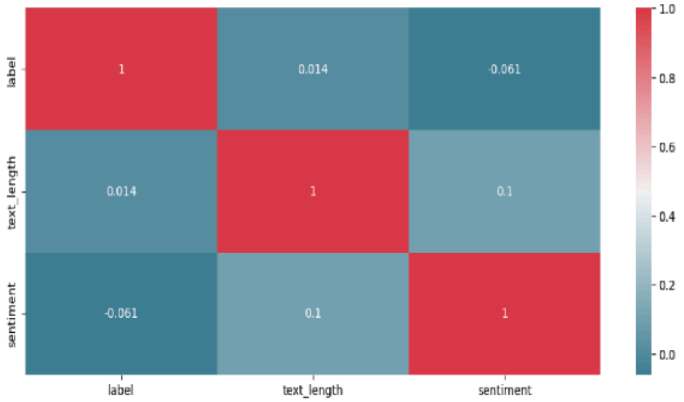


Figure 8: Correlation heatmap showing weak relationships between label, text length, and sentiment. None of the variables strongly predict the others.

Algorithms used in the project

Models, Tuned Parameters and Embedding Types
(Parameters selected via Grid Search, Random Search or validation-based tuning)

Model	Best Parameters (After Tuning)	Embedding Type
Dummy	Predicts most frequent or random class	-
KNN	22 neighbors, weighted by distance	GloVe / CountVectorizer
SVM	C=1.3, linear kernel, degree=4	GloVe / CountVectorizer
Random Forest	119 trees, depth=15, min split=5	GloVe / CountVectorizer
XGBoost	95 trees, depth=8, learning rate=0.05	GloVe / CountVectorizer
AdaBoost	160 trees, learning rate=0.3	GloVe / CountVectorizer
CNN	Embedding + Conv1D + MaxPooling + Dense	Self-trained

Accuracy Comparison Between Models

In our experiments with classical machine learning models, we evaluated two different text embedding methods: **GloVe** (pre-trained word embeddings) and **CountVectorizer** (frequency-based embedding). These two approaches were chosen after reviewing common embedding techniques in the literature, as they were found to be particularly suited for short, informal text like tweets.

We observed that models trained with **CountVectorizer** consistently outperformed those using GloVe. This is likely because CountVectorizer better captures word frequency patterns in domain-specific or informal language, such as slang, abbreviations, and emojis commonly found in tweets.

While GloVe provides rich semantic relationships from large general corpora, it may not reflect the unique vocabulary used on social media platforms. Our results suggest that simple frequency-based methods can be more effective when the data distribution differs significantly from the GloVe training corpus.

It is worth noting that other embedding methods such as **Word2Vec**, or **TF-IDF**,were considered but not tested due to scope limitations. These alternatives may be explored in future work to further optimize performance and compare their ability to handle noisy, short-form text.

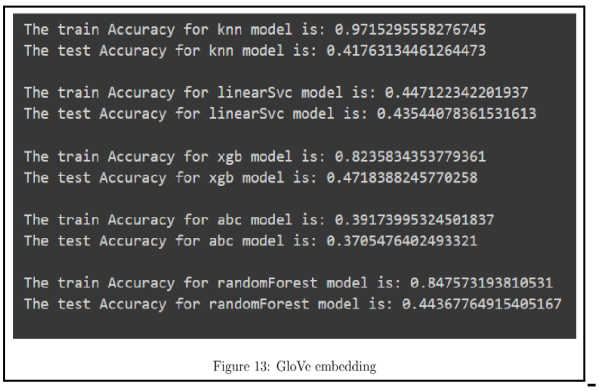


Figure 13: GloVe embedding

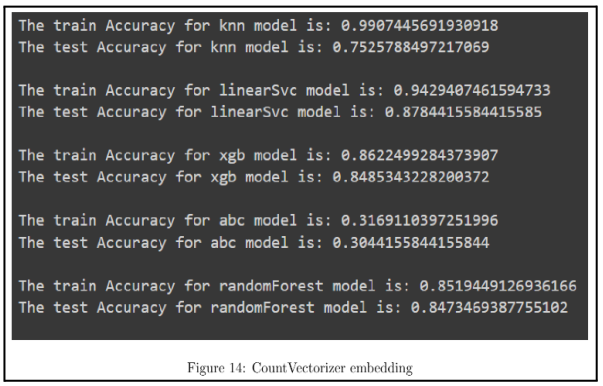
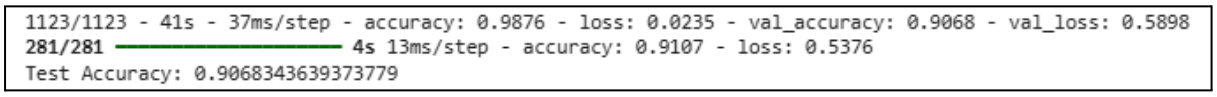


Figure 14: CountVectorizer embedding

As shown above, models trained with CountVectorizer embedding consistently achieved higher test accuracy compared to GloVe, highlighting the impact of embedding choice on model performance.



The CNN model achieved the highest test accuracy (0.9068), demonstrating the advantage of deep learning with learned embeddings over traditional models with static embeddings.

Confusion Matrices for Classical Models

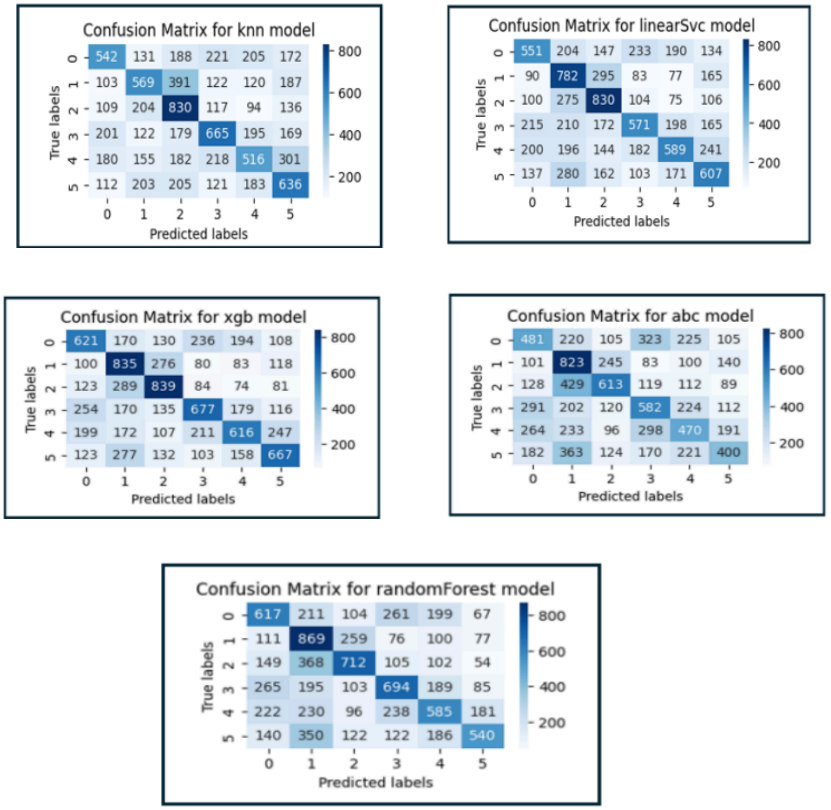


Figure 16: shows the confusion matrices for the classical models.
SVM and XGBoost had strong diagonal lines, meaning good accuracy across emotions.
AdaBoost, however, made more mistakes, especially mixing **anger with fear**.
These errors show the challenge of telling apart emotions that sound or feel similar.

Fine-Tuning BERT for Emotion Classification

To leverage recent advancements in Natural Language Processing, we performed fine-tuning on BERT, a powerful pre-trained transformer model, widely recognized for its superior language understanding capabilities. We randomly selected **3,000 tweets** from the original dataset and split them into 80% training and 20% testing subsets. Each emotion was encoded as an integer (0–5) for classification purposes. Tweets were tokenized using BERT's built-in tokenizer with a maximum length of **64 tokens**, suitable for short-form texts.

The fine-tuning process involved training the model for 3 epochs (complete training cycles), using a batch size of 4 tweets per step due to computational constraints, and employing standard fine tuning hyperparameters (learning rate and weight decay) recommended by HuggingFace guidelines.

The fine-tuned BERT model achieved an impressive accuracy of **89.33%** on the test dataset. close to our best-performing CNN model (**90.68%**) and significantly higher than all classical machine learning methods tested.

This result clearly demonstrates that fine-tuning transformer-based models, even briefly and on limited datasets, can significantly improve emotion classification performance, effectively capturing subtle emotional cues within tweets.

```
fine tuned BERT Accuracy: 89.33 %
```

Summary of Main Findings

We compared different methods for emotion detection in short tweets: classical ML, CNN, and fine-tuned BERT.

- **CNN and BERT** gave the best results, showing strong ability to understand emotions in short, informal text.
- **LinearSVC and XGBoost** did fairly well, but **AdaBoost** struggled with noisy and sparse data.
- **Learned embeddings** (from CNN and BERT) worked better than basic ones like CountVectorizer.

CNN reached the highest accuracy (90.68%), with BERT close behind (89.33%), showing the power of deep learning for this task.

Insights from Error Analysis

The confusion matrices showed where the models made mistakes:

- **Fear and Anger** were often confused because they use similar words.
- **Sadness and Love** were sometimes mixed up due to subtle context in short texts.

These errors suggest that more data or better text processing could help improve accuracy in future work.

Practical Implications Of This Project

Emotion detection in tweets can be useful in many areas:

- **Mental health** – Detecting emotional distress online can help with early support.
- **Customer service** – Smarter chatbots can respond more emotionally and improve user experience.
- **Social media analysis** – Understanding public feelings about events or brands helps in marketing and research.

Limitations and Challenges

Although our results were promising, we faced a few limitations:

- **Limited fine-tuning data** – We trained BERT on only 3,000 examples due to time and GPU limits (even with a Tesla GPU, training took over an hour). More data could improve accuracy and stability.
- **Emotion ambiguity** – Short tweets often express more than one emotion or unclear feelings. Using only one label may miss these emotional layers.

Future Research Directions

Based on our findings and limitations, we suggest the following directions:

- **Try more embedding methods** like Word2Vec or FastText to see if they improve emotion classification.
- **Use more training data** when fine-tuning models like BERT or RoBERTa to get better results.
- **Explore multi-label emotion detection**, where a tweet can express more than one emotion at the same time.

Work Distribution

This project was done equally by both of us – Avichay and Orel.

We worked together on all parts, supported each other, and made every decision as a team.

Avichay focused more on the preprocessing and EDA, while Orel worked more on the classical models.

We both collaborated on training the CNN and fine-tuning the BERT model, and wrote the report together.

The work was fully shared and balanced throughout the process.