Project Emoticonizer

Decoding Emotions from Text

A project work done in partial fulfilment of the "Certificate course on Data Analytics & Business Intelligence"

SHAHEED SUKHDEV
COLLEGE OF
BUSINESS STUDIES

Submitted by:

Sweety, Shridhi Chatterjee
Certificate course on Data Analytics & Business Intelligence Batch-10
Shaheed Sukhdev College of Business Studies
May 2024

DECLARATION

We, *Sweety and shridhi*, **declare that this project titled “** *Project Emoticonizer: Decoding Emotions from text* **” is the original work done by us under the guidance of** *Dr. Rishi Ranjan Sahay,* **Assistant Professor, Shaheed Sukhdev College of Business Studies, University of Delhi.**
we further declare that this work is for my certificate course in Data Analytics and Business Intelligence.

Name: Sweety
Name: Shridhi

ACKNOWLEDGEMENT

We would like to extend my heartfelt gratitude to everyone who has supported and guided me throughout the completion of this project.

First and foremost, I am deeply thankful to my advisor, **Dr. Rishi Ranjan Sahay**, whose guidance, and encouragement have been invaluable. His extensive knowledge, constructive feedback, and constant support have significantly contributed to the success of this project.

we also want to express my appreciation to my professors and peers who assisted and shared their insights during my studies. Their contributions have been crucial in shaping the direction of my research.

We are immensely grateful to my family and friends for their unwavering support and motivation, which have been a source of strength for me throughout this journey.

Lastly, we would like to acknowledge Shaheed Sukhdev College of Business Studies for providing the resources and a conducive environment necessary for this research.

Thank you all for your tremendous help and encouragement.

Sweety
Shridhi

Table of Contents

ABSTRACT / EXECUTIVE SUMMARY

Emotion detection through text is an essential aspect of various applications, from sentiment analysis in customer feedback to monitoring social media for mental health trends. This project, titled "**Emoticonizer**: Decoding Emotions Through Text," aims to develop a robust and accurate emotion detection model using both traditional machine learning approaches and modern ensemble methods. By leveraging traditional NLP techniques and integrating multiple machine learning algorithms through a Voting Classifier, the project seeks to achieve high accuracy and reliability in emotion classification. This work is conducted as part of a final project submission for **data analytics and business intelligence** course, **Shaheed Sukhdev college of business studies**. In this project we will see how to use hybrid approach by combining multiple machine learning models with ensemble model for emotions detections.

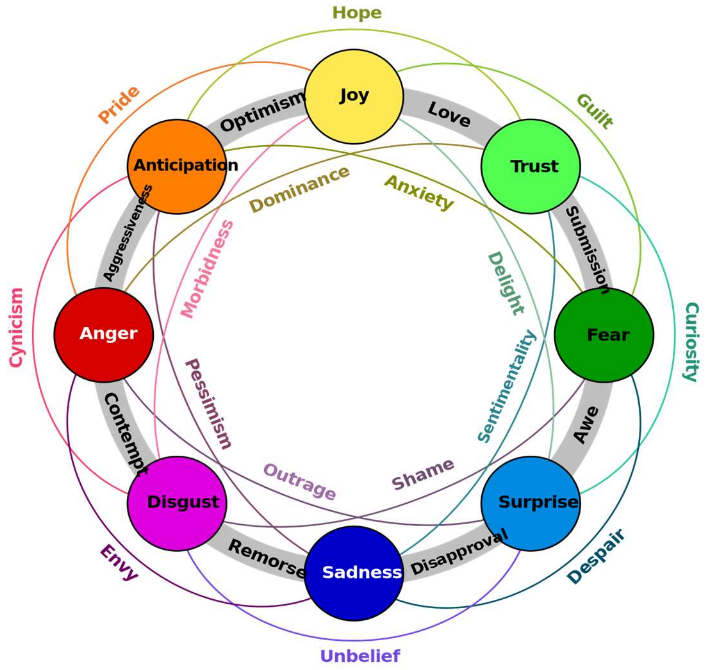
We have used natural language processing (NLP) techniques and machine learning algorithms to achieve high accuracy. The dataset includes diverse text sources like social media posts, reviews, and news articles, annotated with sentiment labels. Preprocessing involves tokenization, stop-word removal, and stemming to clean the text.

We have explored several machine learning algorithms, including vector classifier, Support Vector Machines (SVM), Random forest, multinomial naïve bayes ,logistic regression and Random Forests. Model performance is evaluated using accuracy, precision, recall, and F1-score.

The results demonstrate the model's effectiveness in accurately classifying emotions, with applications in social media monitoring, customer feedback analysis, and market research. Future work will refine the model, incorporate more diverse datasets, and explore real-time sentiment analysis to enhance robustness and applicability.

INTRODUCTION

In the realm of human experience, emotions stand out as intricate phenomena that can be perplexing to both grasp and quantify. Natural language processing (NLP), however, has ushered in a new era where machines can be trained to recognize emotions within text. This dataset serves as a valuable resource for researchers and developers invested in constructing NLP applications with the ability to comprehend human emotions.



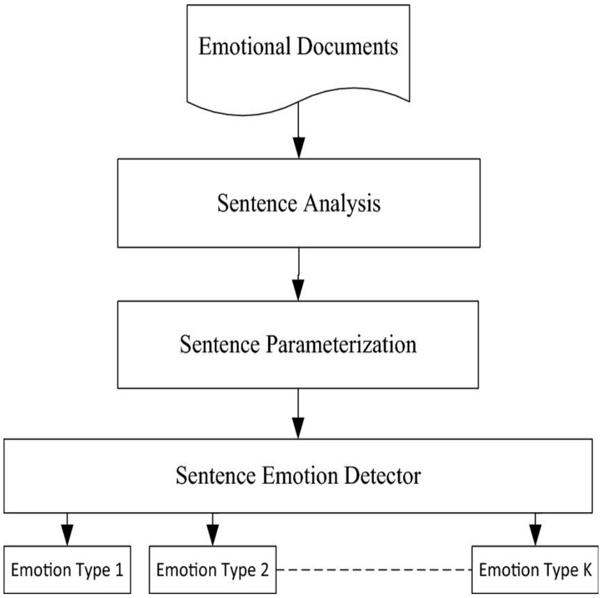
The dataset is comprised of a collection of documents that have been assigned emotional labels. This characteristic makes it possible to train machine learning models to pinpoint emotions within textual data. The dataset has the potential to be applied to a wide range of NLP tasks, including sentiment analysis, the extraction of opinions, and the development of chatbots.

Understanding Emotions Through Text

Emotions are fundamental to human interaction and communication, and they significantly influence how we perceive and respond to the world around us. They can be expressed verbally, through facial expressions, body language, and even tone of voice. However, pinpointing emotions within written text can be a challenging task, as the nuances of human language can often be ambiguous.

Here's where NLP comes into play. NLP is a subfield of artificial intelligence (AI) that deals with the interaction between computers and human language. NLP researchers have developed techniques for machines to process and understand human language, including the ability to identify emotions.

The Role of the Emotions Dataset for NLP



The emotions dataset for NLP plays a pivotal role in training machines to recognize emotions within the text. The dataset consists of a collection of documents that have been annotated with emotional labels. These labels can be specific emotions, such as happiness, sadness, anger, or fear, or they can be more general categories, such as positive or negative sentiment.

By training machine learning models on this labelled data, researchers can enable the models to learn the patterns and associations between specific words, phrases, and sentence structures, and the

Applications of the Emotions Dataset for NLP

The emotions dataset for NLP has the potential to be applied to a wide range of NLP tasks. Here are a few examples:

Sentiment Analysis: Sentiment analysis is the task of identifying the emotional tone of a piece of text. This can be used to gauge customer satisfaction in product reviews, understand public opinion on social media, or track brand sentiment online.

Opinion Mining: Opinion mining is the task of extracting opinions and sentiments from text data. This can be used to identify the opinions of customers about products or services or to understand public opinion on current events.

Chatbot Development: Chatbots are computer programs that are designed to simulate conversation with human users. By incorporating emotion recognition capabilities, chatbots can provide more natural and engaging interactions with users.

Mental Health Analysis: The emotions dataset could potentially be used to develop tools for mental health analysis. For example, it could be used to analyze social media posts or other forms of text data to identify individuals who may be at risk of suicide or other mental health problems.

RESEARCH OBJECTIVE

The primary objective of this project is to develop an emotion detection model using both traditional machine learning techniques and modern ensemble methods. By comparing these approaches, we aim to identify the most effective method for accurately detecting emotions from textual data.

1. Model Development and Evaluation:

We will explore and compare different machine learning algorithms, such as Support Vector Machines (SVMs), Naive Bayes, and deep learning architectures, to determine the most effective approach for emotion classification in this dataset.
The chosen model will be rigorously trained and evaluated on a designated portion of the dataset. Metrics like accuracy, precision, recall, and F1-score will be employed to assess the model's performance in recognizing various emotions.

2. Understanding Language Patterns of Emotion:

We will delve deeper into the dataset to identify linguistic features and patterns associated with specific emotions. This analysis might involve exploring the use of sentiment-laden words, negation terms, punctuation styles, and sentence structures in conveying emotions.
The insights gained from this analysis will be used to refine the model and potentially develop new feature engineering techniques to enhance emotion recognition accuracy.

3. Exploring Domain-Specific Applications:

We will investigate the potential of applying the trained model to real-world scenarios within specific domains. This might involve testing the model's performance on data from customer reviews, social media posts, or dialogue systems.

Analyzing the model's effectiveness in these domains will provide valuable insights into its generalizability and potential need for further adaptation.

4. Addressing Bias and Limitations:

We acknowledge the potential for bias within the dataset, as emotional expressions can vary culturally and linguistically. We will explore techniques to mitigate bias and ensure the model's generalizability across diverse datasets.
Additionally, we will investigate the limitations of the model in terms of recognizing complex emotions, sarcasm, and nuanced language usage. Exploring methods to address these limitations will be a key focus of our research.

By achieving these objectives, this project aims to contribute significantly to the field of NLP by developing a more robust and nuanced approach to emotion recognition in text data. This research has the potential to pave the way for the creation of advanced NLP applications that can better understand human sentiment and emotion, leading to more engaging and effective human-computer interactions.

RESEARCH PROBLEM

Despite significant advancements in NLP, accurately recognizing emotions in written text remains a challenging task. This research problem stems from the inherent complexities of human language and the subtle ways emotions are conveyed.

1. Ambiguity and Nuance:

Human language is rife with ambiguity. Words can have multiple meanings depending on context, and emotions can be expressed implicitly through sarcasm, figurative language, or sentence structure.
This ambiguity makes it difficult for machines to accurately interpret the intended emotional tone of a text.

2. Limited Training Data:

Building robust emotion recognition models requires substantial amounts of labelled data, where text is categorized with the specific emotions it conveys.
However, creating high-quality, emotionally labelled data is a time-consuming and expensive process. Additionally, existing datasets might not capture the full spectrum of emotions, or the nuances of informal language used in online communication.

3. Domain Specificity:

Emotional expressions can vary significantly between different domains, such as customer reviews, social media interactions, or formal documents.
A model trained on one domain might struggle to generalize and accurately recognize emotions in another.

4. Bias and Cultural Variations:

Existing datasets and models might exhibit bias based on the demographics of the data used for training. This can lead to inaccurate emotion recognition for users from different cultures or backgrounds.
Emotions themselves can be culturally specific, making it challenging to develop universal models for emotion recognition.
These research problems hinder the ability of NLP to fully grasp the emotional nuances of human communication. This project aims to address these challenges by developing a more robust and adaptable approach to emotion recognition in text data.

RESEARCH SCOPE AND METHODOLOGY

This research focuses on leveraging the "Emotions Dataset for NLP" to develop a powerful emotion recognition system for textual data. The breakdown of our scope and methodology are given below:

Emotions: We will focus on recognizing primary emotions like happiness, sadness, anger, fear, disgust, surprise, and neutral sentiment.
Textual Data: We will primarily work with written text formats from the provided dataset and explore potential applications to specific domains like customer reviews or social media posts in the later stages.
Machine Learning Techniques: We will investigate various machine learning algorithms like SVMs, Naive Bayes, and deep learning architectures for model development.
Methodology:

Our research approach will follow a systematic process:

Data Preprocessing: We will clean the data by removing noise and inconsistencies. Feature engineering techniques will be employed to extract relevant features like word n-grams, sentiment lexicons, and part-of-speech tags.

Model Development and Training: We will explore and compare different machine learning algorithms like SVMs, Naive Bayes, and deep learning architectures. The chosen model will be trained on a designated portion of the dataset, with the remaining portion reserved for testing and validation.

Model Evaluation: We will evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score to identify the strengths and weaknesses, guiding further development.

Analysis and Interpretation: We will analyze the model's performance to understand the linguistic features and patterns associated with different emotions, refining the model and potentially developing new feature engineering approaches.

Domain-Specific Exploration: We will explore the model's generalizability by testing it on data from specific domains, potentially including customer reviews or social media posts. This analysis will provide insights into the need for domain-specific adaptations.

Bias Mitigation: We will be mindful of potential biases within the dataset and explore techniques like data augmentation or fairness-aware training methodologies to mitigate their impact.

By addressing these research challenges and following our outlined methodology, this project aims to develop a more robust and adaptable system for emotion recognition in text, paving the way for advanced NLP applications that can better understand human sentiment and emotion.

LITERATURE REVIEW

Emotion detection from text is a rapidly evolving field within Natural Language Processing (NLP) and has garnered significant attention due to its wide range of applications, including sentiment analysis, customer service, mental health monitoring, and social media analysis. This review highlights recent advancements, compares traditional methods with modern approaches, and discusses current trends in text-based emotion detection using machine learning.

Traditional Approaches in Emotion Detection

Early research in emotion detection relied heavily on lexicon-based methods, which used predefined lists of emotion words to analyze text. These methods were limited by their inability to handle context and polysemy effectively. The introduction of machine learning algorithms, such as Naive Bayes and Support Vector Machines, marked a significant improvement by allowing models to learn from data and generalize to new inputs.

1. Rule-Based Methods

Early emotion detection systems relied heavily on rule-based approaches, utilizing predefined lexicons and linguistic rules. These methods involved:

- **Emotion Lexicons:** Dictionaries like WordNet-Affect, where words are associated with specific emotions.
- **Pattern Matching:** Using regular expressions and syntactic patterns to identify emotion-bearing phrases.

Limitations: Rule-based methods, while straightforward, often lacked generalization capabilities and were not robust to the nuances of natural language, leading to poor performance on diverse datasets.

2. Statistical Methods

With the advent of more computational power, statistical methods became popular. These include:

- **Naive Bayes Classifiers:** Simple probabilistic models used for text classification.
- **Support Vector Machines (SVM):** Employed for their robustness in high-dimensional spaces.

Limitations: These methods required extensive feature engineering and were often outperformed by more sophisticated models in capturing context and semantics.

Modern Approaches

In recent years, deep learning models like Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Transformer-based models (e.g., BERT and GPT) have shown remarkable performance in NLP tasks. These models can capture complex patterns and contextual information, making them highly effective for emotion detection. However, they require substantial computational resources and large datasets.

1. Machine Learning Techniques

- Bag-of-Words (BoW) and TF-IDF: These methods convert text into numerical vectors, which can then be used by traditional machine learning models like Logistic Regression, SVMs, and Random Forests.

Pros: Simple to implement and interpret.

Cons: Fail to capture the semantic meaning and context of words.

2. Deep Learning Techniques

- Recurrent Neural Networks **(RNNs) and** Long Short-Term Memory Networks **(LSTMs):** These architectures are designed to handle sequential data and have been used effectively for sentiment and emotion analysis.

Pros: Capable of learning temporal dependencies in text.

Cons: Computationally intensive and require large datasets for training.

- Convolutional Neural Networks **(CNNs):** Initially popular in image processing, CNNs have been adapted for text classification tasks, leveraging their ability to capture local features.

Pros: Efficient in capturing local patterns in text.

Cons: Limited in capturing long-range dependencies without significant modifications.

3. Transfer Learning and Pre-trained Models

- Word Embeddings: Techniques like Word2Vec and GloVe provided dense vector representations of words, significantly improving the performance of downstream NLP tasks by capturing semantic relationships.

- Transformer-based Models: The introduction of transformer architectures, particularly **BERT** (Bidirectional Encoder Representations from Transformers), revolutionized **NLP. BERT** and its successors **(RoBERTa, GPT-3**, etc.) have demonstrated state-of-the-art performance in various NLP tasks, including emotion detection.

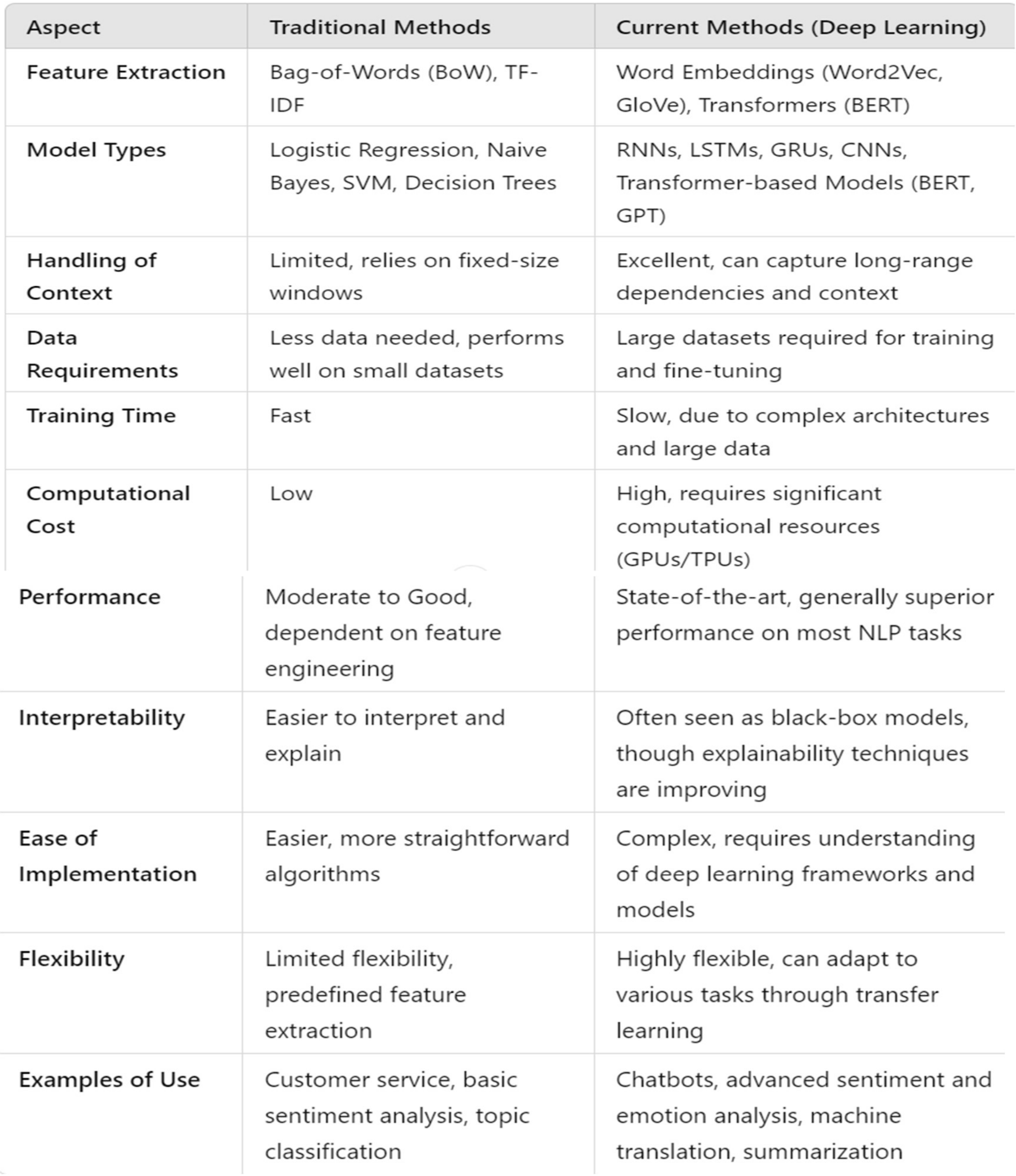
Pros: Pre-trained on massive corpora, capturing rich semantic and syntactic information.
Cons: Resource-intensive and require fine-tuning for specific tasks.

Hybrid Approaches

Combining traditional machine learning models with ensemble techniques has proven to be a promising approach. Ensemble methods like Voting Classifiers leverage the strengths of multiple algorithms to create more robust models. This hybrid approach balances the simplicity and efficiency of traditional methods with the improved performance of modern techniques.

Comparison of Methods

Traditional NLP vs. Deep Learning



Aspect	Traditional Methods	Current Methods (Deep Learning)
Feature Extraction	Bag-of-Words (BoW), TF-IDF	Word Embeddings (Word2Vec, GloVe), Transformers (BERT)
Model Types	Logistic Regression, Naive Bayes, SVM, Decision Trees	RNNs, LSTMs, GRUs, CNNs, Transformer-based Models (BERT, GPT)
Handling of Context	Limited, relies on fixed-size windows	Excellent, can capture long-range dependencies and context
Data Requirements	Less data needed, performs well on small datasets	Large datasets required for training and fine-tuning
Training Time	Fast	Slow, due to complex architectures and large data
Computational Cost	Low	High, requires significant computational resources (GPUs/TPUs)
Performance	Moderate to Good, dependent on feature engineering	State-of-the-art, generally superior performance on most NLP tasks
Interpretability	Easier to interpret and explain	Often seen as black-box models, though explainability techniques are improving
Ease of Implementation	Easier, more straightforward algorithms	Complex, requires understanding of deep learning frameworks and models
Flexibility	Limited flexibility, predefined feature extraction	Highly flexible, can adapt to various tasks through transfer learning
Examples of Use	Customer service, basic sentiment analysis, topic classification	Chatbots, advanced sentiment and emotion analysis, machine translation, summarization

Conclusion

Text-based emotion detection has progressed significantly from rule-based and statistical methods to sophisticated deep-learning techniques. The advent of pre-trained models like BERT has set new benchmarks in the field, achieving high accuracy and robustness. Current trends indicate a move towards more comprehensive, multimodal, and real-time emotion detection systems, with a focus on interpretability and extending capabilities to a wider range of languages. As the field continues to evolve, these advancements promise to enhance the effectiveness and applicability of emotion detection systems across various domains.

Main Work and Contribution

In this project, combination of NLP and machine learning algorithms has been used in order to classify as well as predict the various Emotions associated with particular text or set of text.

Initially, the data is explored and **visualized**, followed by **preprocessing** and cleaning of data which include converting the text to lowercase, removing the non-alphabetic characters, Tokenize the text, remove stopwords, and apply stemming and lemmatization.

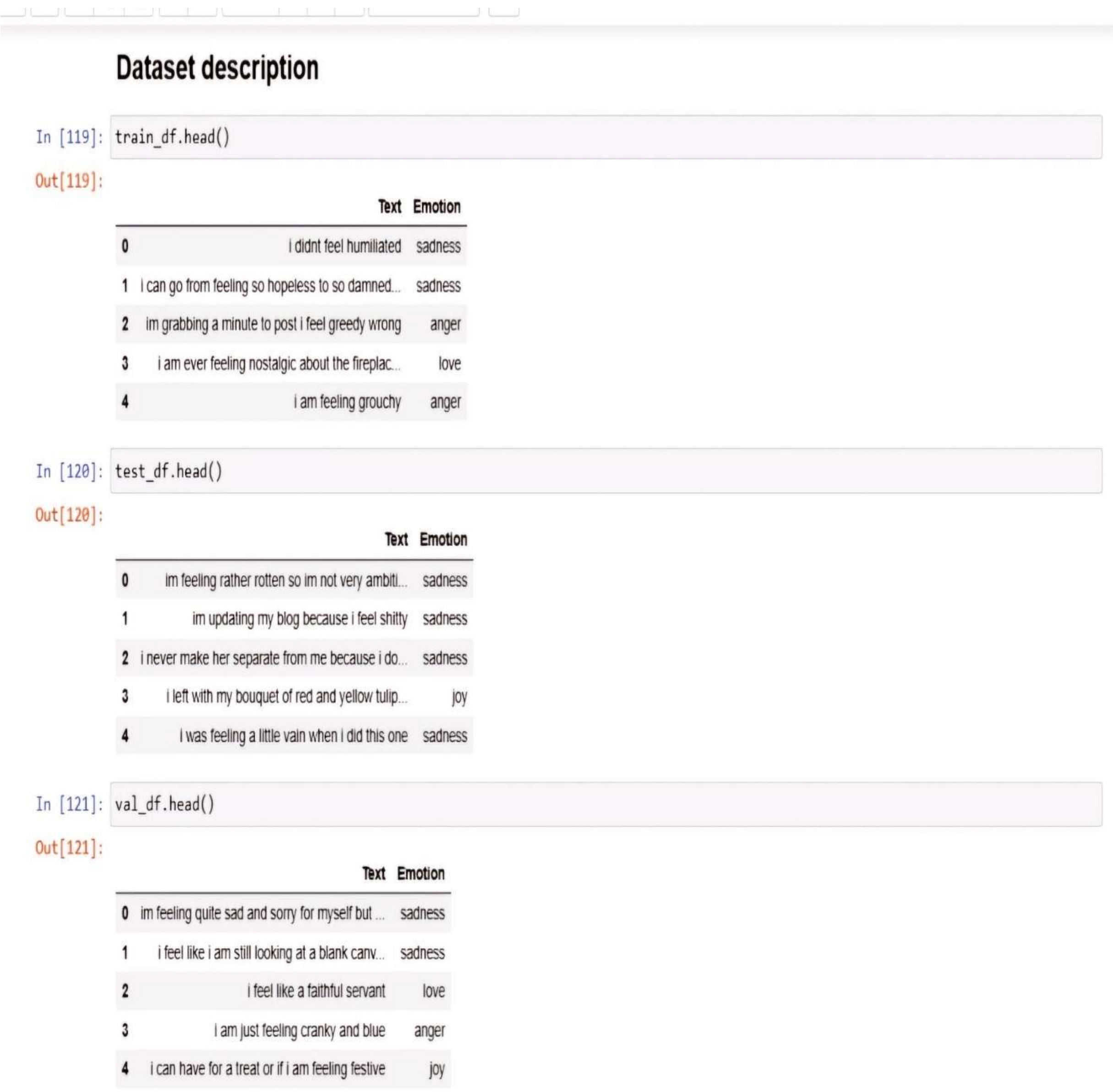
After that, **Features Extraction** is done by using vectorizer, TF-IDF or bag-of-words. Then, appropriate **Model selection and Training** is done followed by fitting the model and finally, **model evaluation**. Since the data is qualitative and so are personality and behavior, this project has endeavored to predict using this algorithm, the resulting accuracy of the model is satisfactory.

TD-IDF
TF-IDF, TF stands for Term Frequency, and IDF refers to Inverse Document Frequency. The TF-IDF model can estimate the importance of a word to a document in a document set. The more times a word appears in a document, the more important the word is to the document, and the more frequently it appears in the document set, the less important the word is.

However, same as the **bag of words model**, TF-IDF ignores the relationship between each word in the document and fails to consider important information such as word order, semantics and syntax. "Bag-of-words," where each feature represents a word in the vocabulary, and its value indicates how many times that word appears in the text.

Data Set

In this project, we considered the Emotion detection Dataset from Kaggle. The "Emotion Detection" dataset is depicted in is freely accessible on Kaggle and includes 3 datasets training set, testing set, and validation set having 16000, 2000,2000 records of data respectively. There are two columns in each row, one for the Text / sentence and the other for emotion of individuals. This information was gathered from the twitter social media since it has a big variety of users.



Dataset description

```
In [119]: train_df.head()
Out[119]:
```

	Text	Emotion
0	i didnt feel humiliated	sadness
1	i can go from feeling so hopeless to so damned...	sadness
2	im grabbing a minute to post i feel greedy wrong	anger
3	i am ever feeling nostalgic about the fireplac...	love
4	i am feeling grouchy	anger

```
In [120]: test_df.head()
Out[120]:
```

	Text	Emotion
0	im feeling rather rotten so im not very ambiti...	sadness
1	im updating my blog because i feel shitty	sadness
2	i never make her separate from me because i do...	sadness
3	i left with my bouquet of red and yellow tulip...	joy
4	i was feeling a little vain when i did this one	sadness

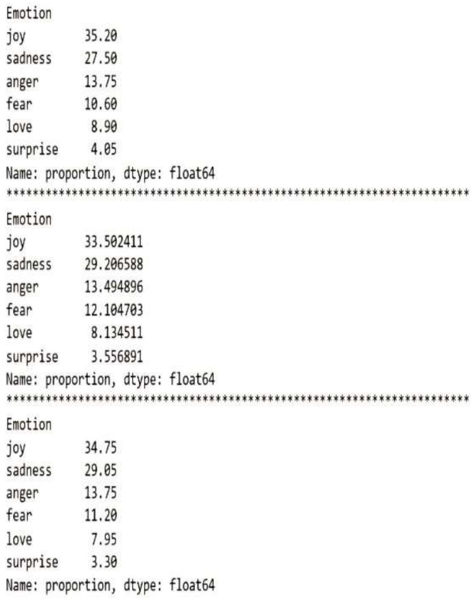
```
In [121]: val_df.head()
Out[121]:
```

	Text	Emotion
0	im feeling quite sad and sorry for myself but ...	sadness
1	i feel like i am still looking at a blank canv...	sadness
2	i feel like a faithful servant	love
3	i am just feeling cranky and blue	anger
4	i can have for a treat or if i am feeling festive	joy

Exploratory Data Analysis

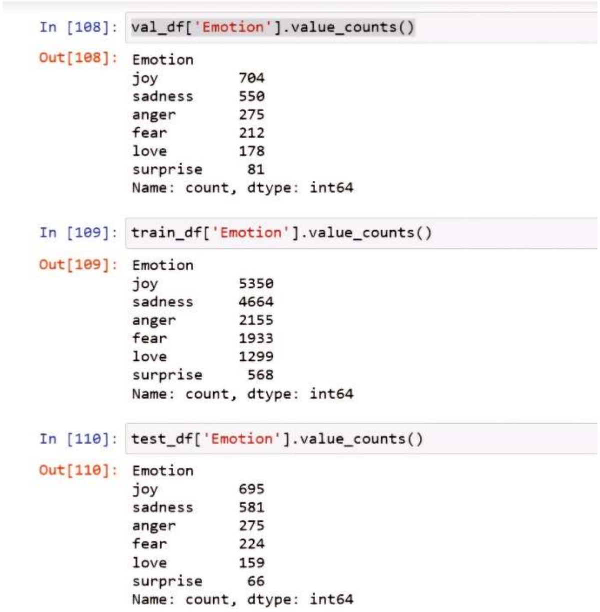
Exploratory Data Analysis involved checking the data, finding out and eliminating any null values, checking the data types involved and the relative as well as absolute figures for every emotion. This enabled the Visualization Stage.

Percentage of emotions distributions in the datasets:

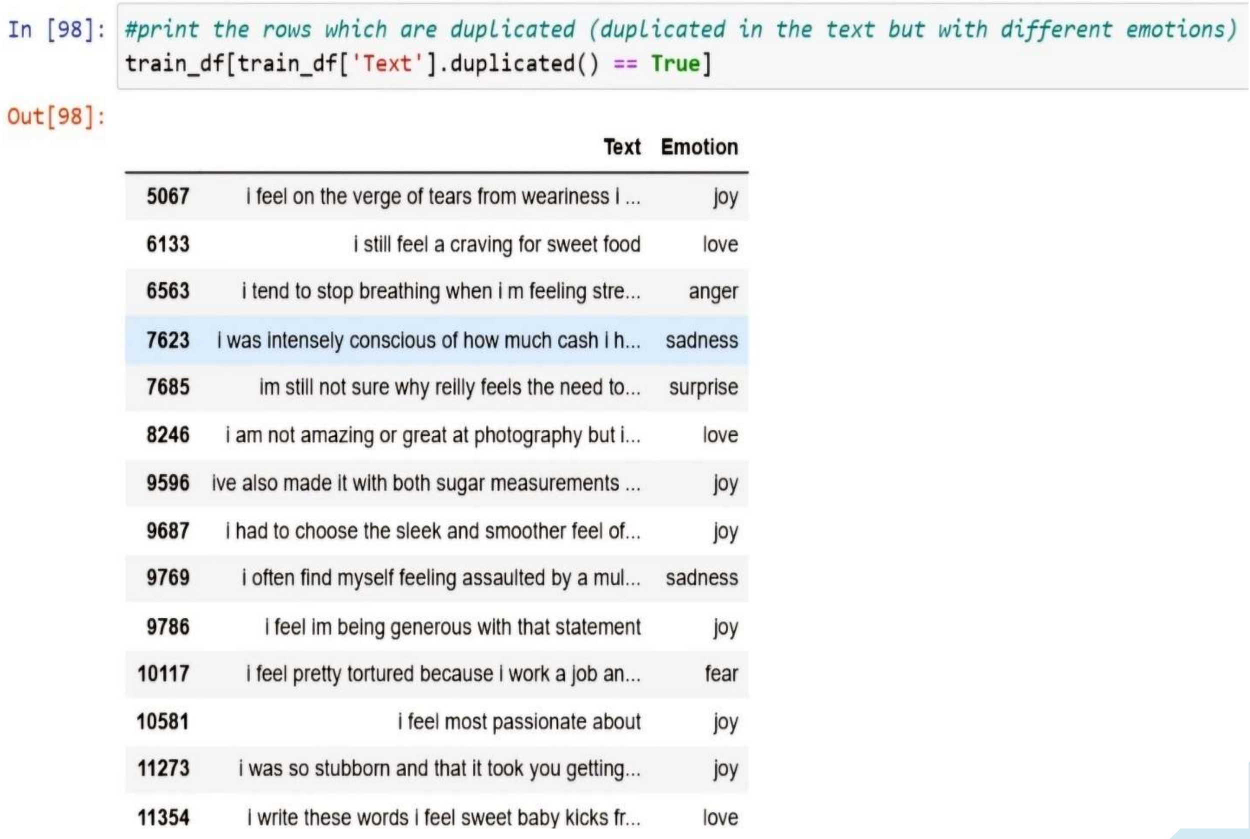


```
Emotion
joy         35.20
sadness     27.50
anger       13.75
fear        10.60
love         8.90
surprise     4.05
Name: proportion, dtype: float64
**********************************************************************
Emotion
joy         33.502411
sadness     29.206588
anger       13.494896
fear        12.104703
love         8.134511
surprise     3.556891
Name: proportion, dtype: float64
**********************************************************************
Emotion
joy         34.75
sadness     29.05
anger       13.75
fear        11.20
love         7.95
surprise     3.30
Name: proportion, dtype: float64
```

Label counts on each dataset:



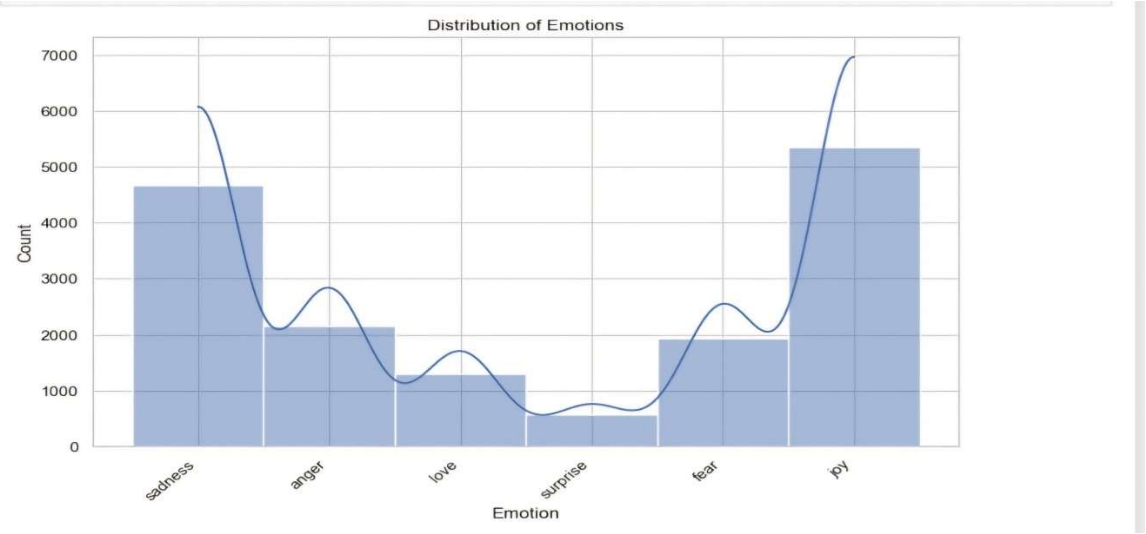
```
In [108]: val_df['Emotion'].value_counts()

Out[108]: Emotion
          joy         704
          sadness     550
          anger       275
          fear        212
          love        178
          surprise     81
          Name: count, dtype: int64

In [109]: train_df['Emotion'].value_counts()

Out[109]: Emotion
          joy         5350
          sadness     4664
          anger       2155
          fear        1933
          love        1299
          surprise     568
          Name: count, dtype: int64

In [110]: test_df['Emotion'].value_counts()

Out[110]: Emotion
          joy         695
          sadness     581
          anger       275
          fear        224
          love        159
          surprise     66
          Name: count, dtype: int64
```



```
In [98]: #print the rows which are duplicated (duplicated in the text but with different emotions)
         train_df[train_df['Text'].duplicated() == True]
```

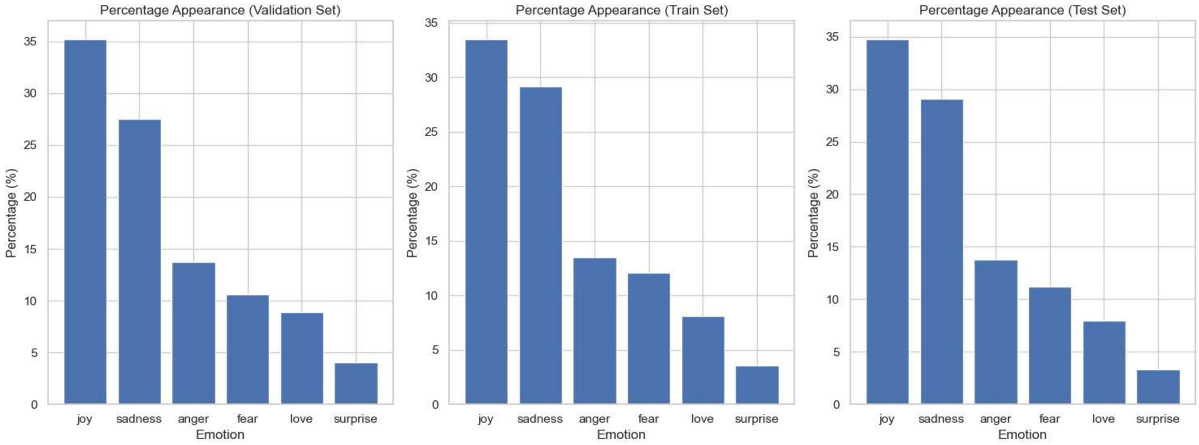
Out[98]:

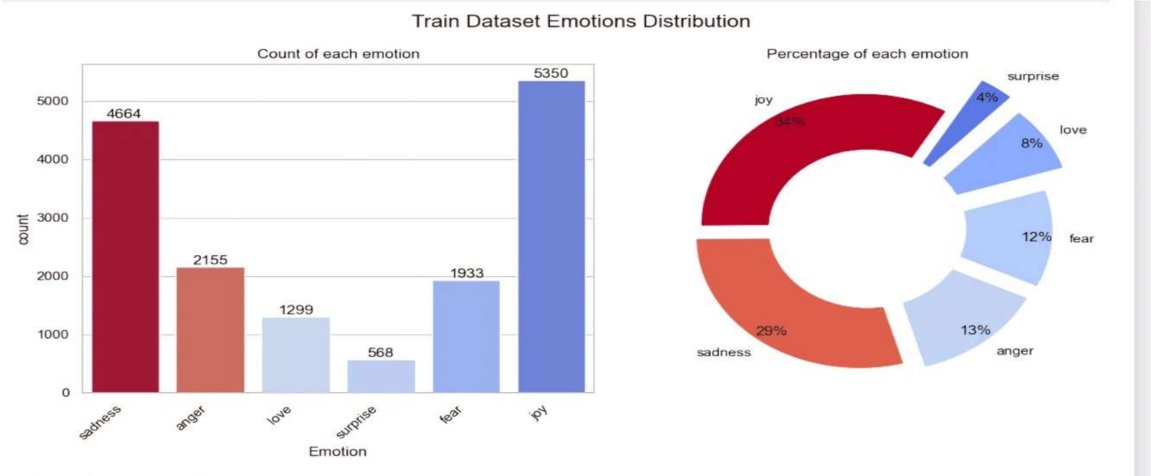
	Text	Emotion
5067	i feel on the verge of tears from weariness i ...	joy
6133	i still feel a craving for sweet food	love
6563	i tend to stop breathing when i m feeling stre...	anger
7623	i was intensely conscious of how much cash i h...	sadness
7685	im still not sure why reilly feels the need to...	surprise
8246	i am not amazing or great at photography but i...	love
9596	ive also made it with both sugar measurements ...	joy
9687	i had to choose the sleek and smoother feel of...	joy
9769	i often find myself feeling assaulted by a mul...	sadness
9786	i feel im being generous with that statement	joy
10117	i feel pretty tortured because i work a job an...	fear
10581	i feel most passionate about	joy
11273	i was so stubborn and that it took you getting...	joy
11354	i write these words i feel sweet baby kicks fr...	love



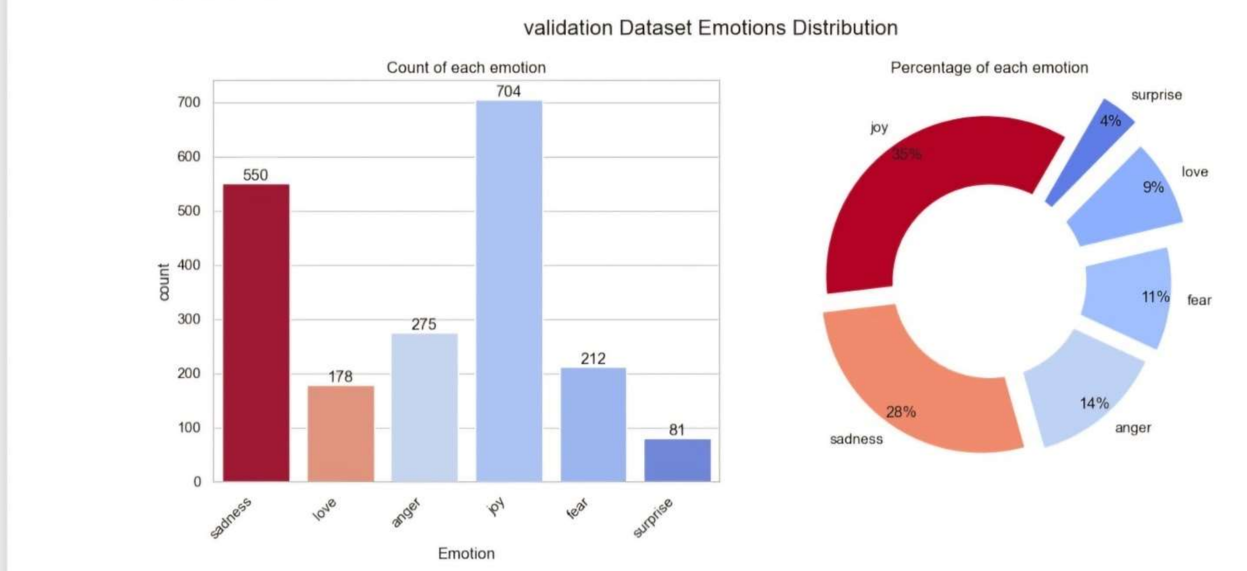
Data Visualization

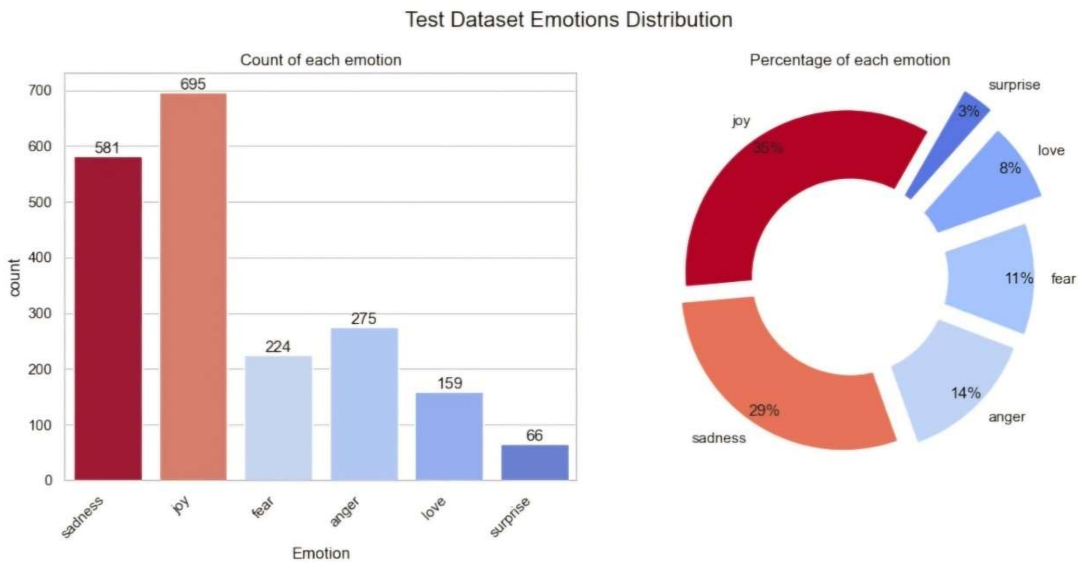
Various visualizations have been created to facilitate an understanding of the data using various parameters such as words per comment. Visualization has been carried out before preprocessing and cleaning, to view the authentic data. The various visualizations created are:




Train Dataset Emotions Distribution
Count of each emotion
count
Emotion
0
1000
2000
3000
4000
5000
4664
2155
1299
568
1933
5350
sadness
anger
love
surprise
fear
joy
Percentage of each emotion
joy
34%
surprise
4%
love
8%
fear
12%
anger
13%
sadness
29%

```
<Figure size 640x480 with 0 Axes>
```


validation Dataset Emotions Distribution
Count of each emotion
count
Emotion
0
100
200
300
400
500
600
700
550
178
275
704
212
81
sadness
love
anger
joy
fear
surprise
Percentage of each emotion
joy
35%
surprise
4%
love
9%
fear
11%
anger
14%
sadness
28%


Test Dataset Emotions Distribution
Count of each emotion
count
Emotion
0
100
200
300
400
500
600
700
581
695
224
275
159
66
sadness
joy
fear
anger
love
surprise
Percentage of each emotion
joy
35%
surprise
3%
love
8%
fear
11%
anger
14%
sadness
29%

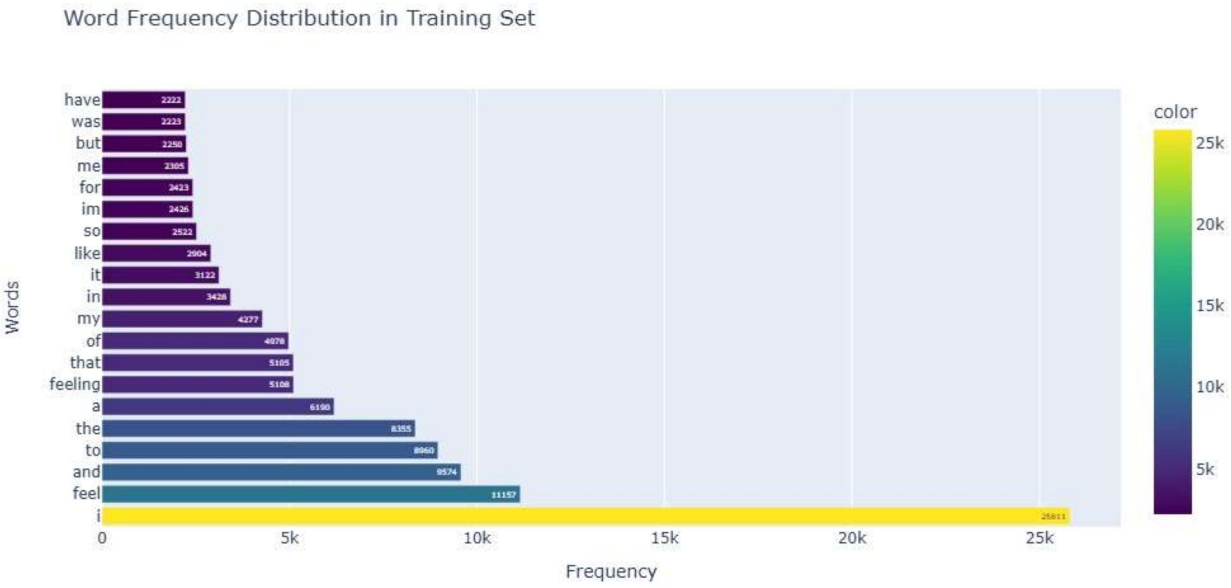
Data Preprocessing and Cleaning

First, data preprocessing and feature extraction are necessary for the data used. To fit the format of the Emotion detection dataset, we filter out punctuation, stopwords, emoticons, numeric and alphanumeric data, repetitive words, very long and very short words, URLs, etc. using Regular Expressions.

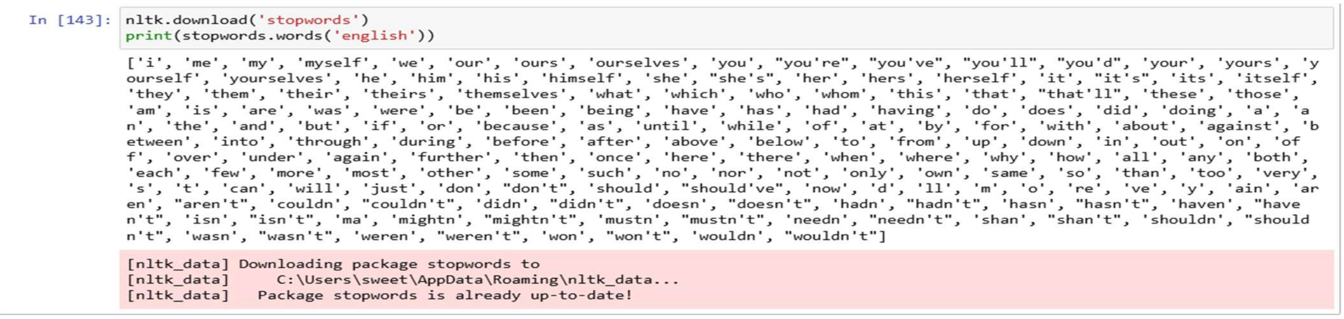
□ **Objective**: Clean and prepare the text data for modeling.

□ **Steps**:

- **Remove Noise**: Strip unwanted characters like HTML tags, punctuation, and numbers.
- **Lowercase Conversion**: Convert all text to lowercase for uniformity.
- **Tokenization**: Split text into individual words (tokens).



- **Stemming/Lemmatization**: Reduce words to their base or root form (e.g., 'running' to 'run').
- **Stopwords Removal**: Remove common words that do not contribute to the emotion (e.g., 'the', 'is').

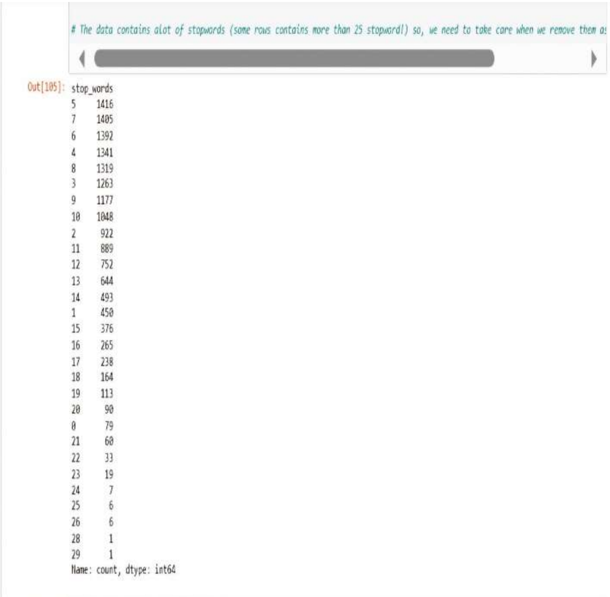


In [143]:
```
nltk.download('stopwords')
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'y
ourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself',
'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those',
'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'a
n', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'b
etween', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'of
f', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very',
's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'ar
en', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "have
n't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "should
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

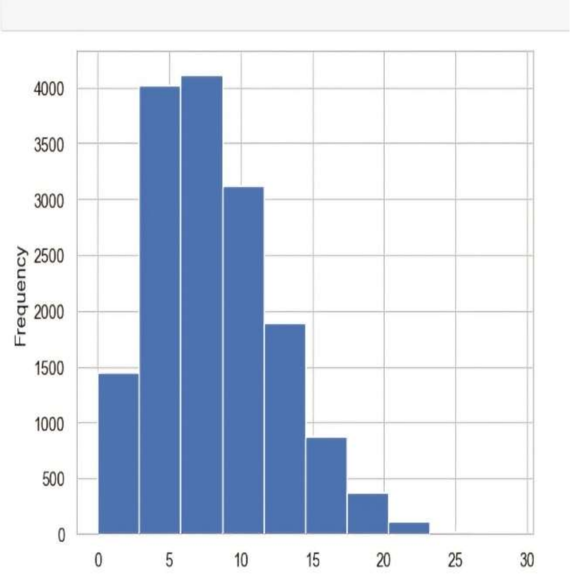
```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\sweet\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

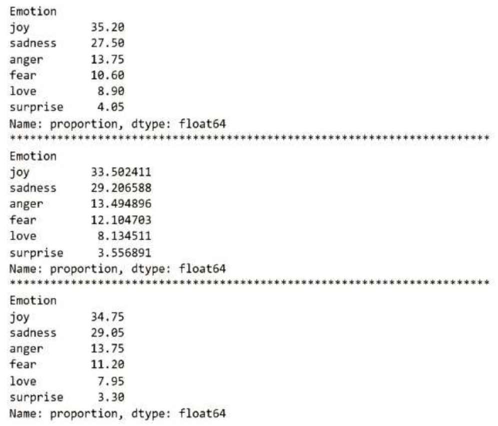
This is a Text Processing class that take any text and convert it to a lower case and stemming it as the user is need.



```
# The data contains alot of stopwords (some rows contains more than 25 stopword!) so, we need to take care when we remove them a:
```

```
Out[105]: stop_words
5     1416
7     1405
6     1392
4     1341
8     1319
3     1263
9     1177
10    1048
2      922
11     889
12     752
13     644
14     493
1      450
15     376
16     265
17     238
18     164
19     113
20      90
0       79
21      60
22      33
23      19
24       7
25       6
26       6
28       1
29       1
Name: count, dtype: int64
```

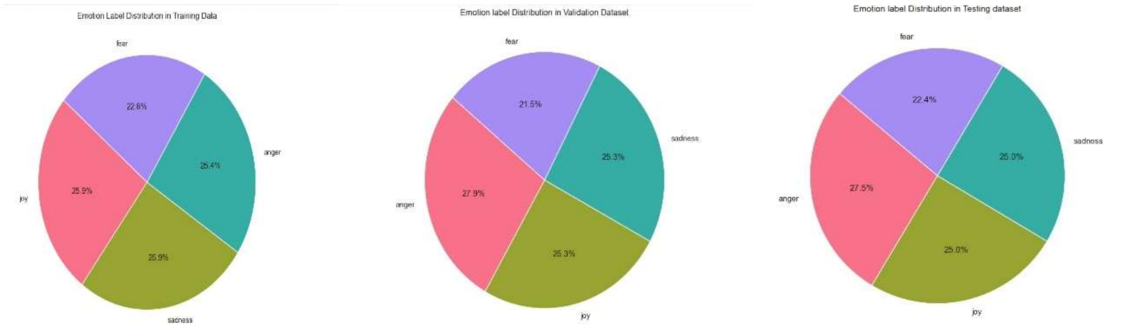




```
Emotion
joy         35.20
sadness     27.50
anger       13.75
fear        10.60
love         8.90
surprise     4.05
Name: proportion, dtype: float64
***********************************************************************
Emotion
joy         33.502411
sadness     29.206588
anger       13.494896
fear        12.104703
love         8.134511
surprise     3.556891
Name: proportion, dtype: float64
***********************************************************************
Emotion
joy         34.75
sadness     29.05
anger       13.75
fear        11.20
love         7.95
surprise     3.30
Name: proportion, dtype: float64
```

It seems like "love" and "surprise" have low representation, likely due to data scarcity. Removing these emotions could enhance model performance and make this unbalanced dataset to balanced

Data distribution after feature engineering and making it balanced by using sampling

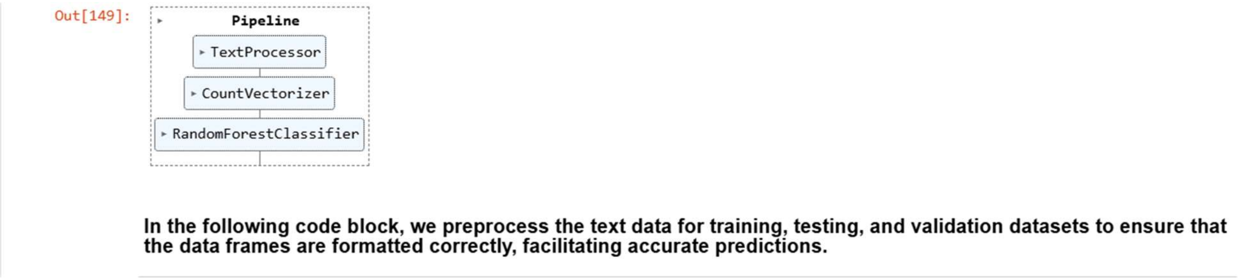


Feature extraction

It was carried out through word embedding techniques, with a focus on experimenting with Bag-of-Words and TF-IDF. Tokens generated by this method are followed by the matrix of TF-IDF features. TF-IDF Vectorization has regulations that can accommodate the weight of tokens that occur frequently and tokens that occur rarely. Therefore, it can extract primary features better than a normal vectorizer. These techniques returned vectors of numerical values reflecting various linguistic properties of the text. There are a lot of repetitive processes for training and testing the dataset separately. To avoid these, pipeline was used.

 Text **Preprocessing**: The script uses regular expressions and NLTK's stopwords to clean and preprocess the text data.

Here we create a pipline that contains:
 1.Text Processing for the input data
 2.Victorizing the text column
 3.Take an object of Random Forest Classifier
We create pipeline for each classifier for text preprocessing.
After the piplines is created We fit the model



In the following code block, we preprocess the text data for training, testing, and validation datasets to ensure that the data frames are formatted correctly, facilitating accurate predictions.

 Feature Extraction: The script uses TF-IDF vectorization to transform the cleaned text data into numerical features suitable for machine learning models.

Training and Testing the Machine Learning Model

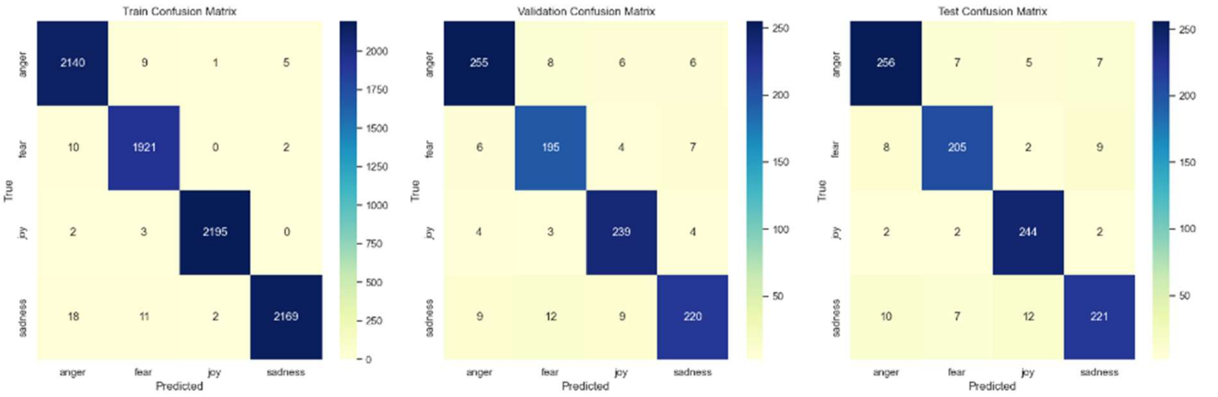
The classification algorithm used in this step are Random Forest, Logistic Regression, Multinominal naïve bayes, SVM etc. It belongs to the Supervised Learning category. Moving on to the training phase, the dataset was split into a training and testing sets already, validation set for evaluating model accuracy. Because we do text classification, the input of the text document should convert into a numerical representation that can be adapted to machine algorithms to make predictions. Model Building

- **Objective**: Train machine learning models to classify emotions based on text features.

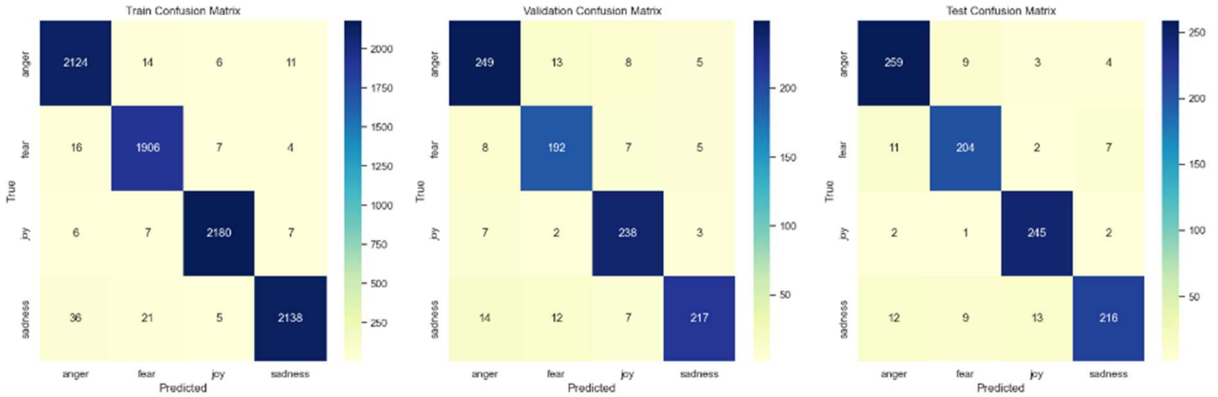
-

Models:

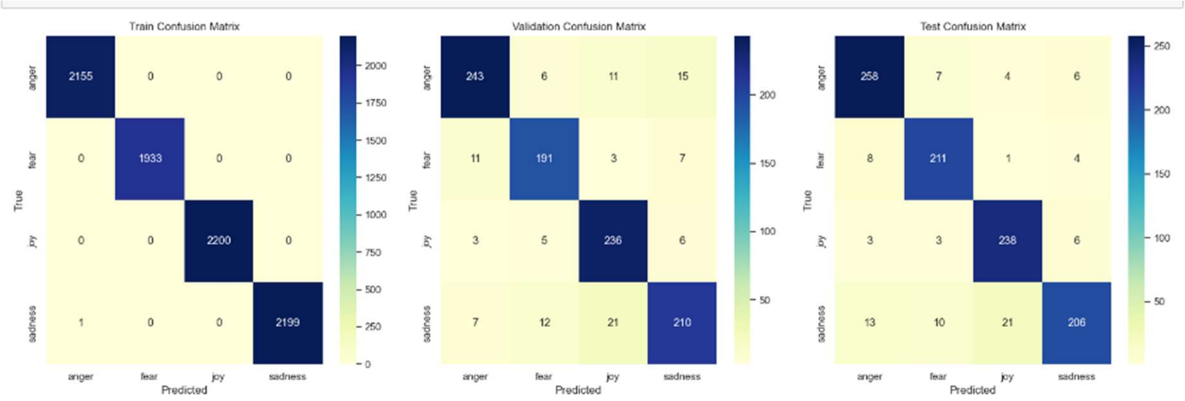
- **Logistic Regression:** A simple linear model for binary classification, extendable to multi-class.



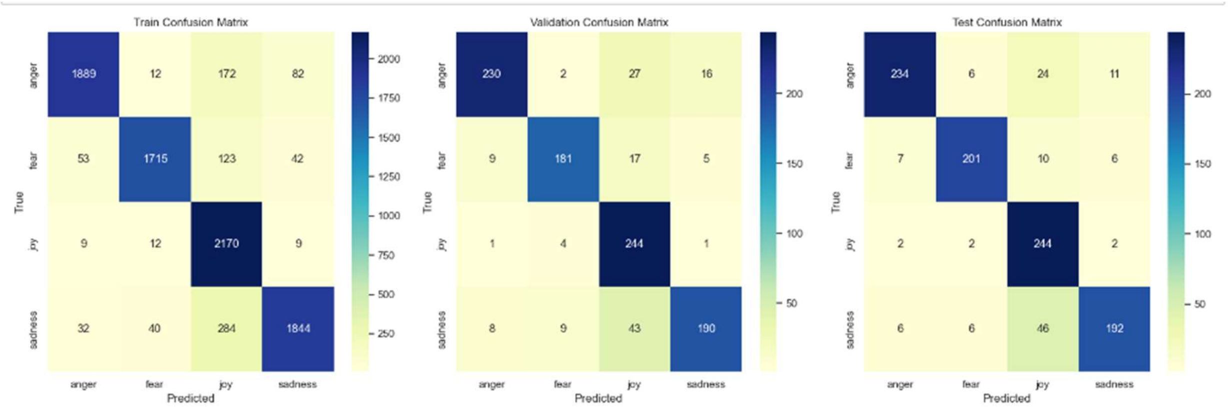
- **Support Vector Machine (SVM):** A powerful classifier that finds the optimal hyperplane to separate classes.



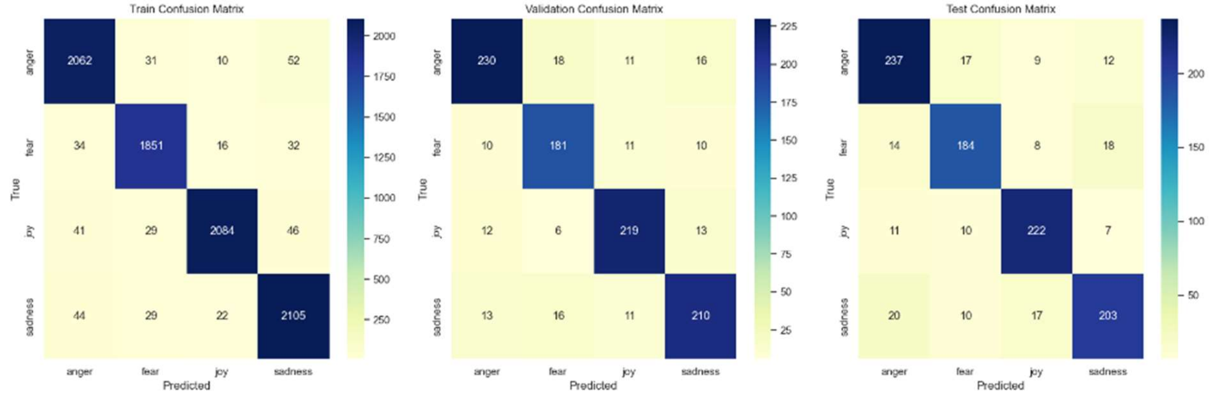
- **Random Forest**: An ensemble method using multiple decision trees to improve classification accuracy.



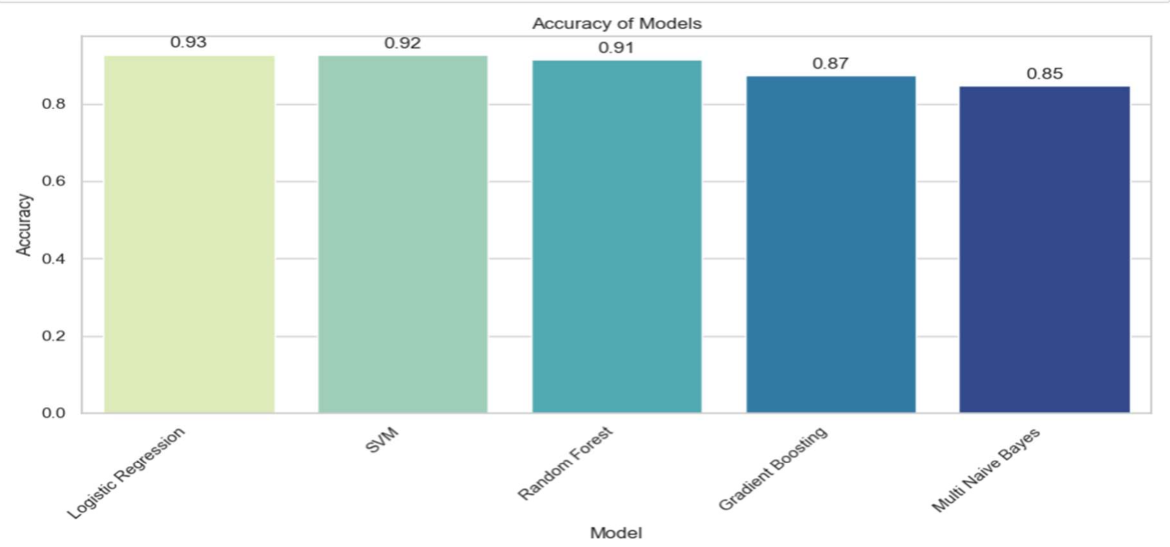
- **Gradient Boosting:** Another ensemble method that builds models sequentially to correct errors of previous models.



- **Multinomial Naive Bayes**: A probabilistic classifier often used for text classification tasks.



Accuracy % of all the machine learning models.



```
<Figure size 640x480 with 0 Axes>
```

The code uses a technique called "Voting Classifier" to combine predictions from multiple machine learning models. It includes three models: Random Forest Classifier, Logistic Regression, and Support Vector Machine. These models are trained on textual data and corresponding emotions. The Voting Classifier learns from these individual models to make a final prediction.

After checking the confusion matrix and classification reports, we find out the Logistic Regression gives the highest Accuracy score of 93% followed by SVM and Random Forest. We will use these 3 models in Ensemble method for better prediction.

Ensemble Method

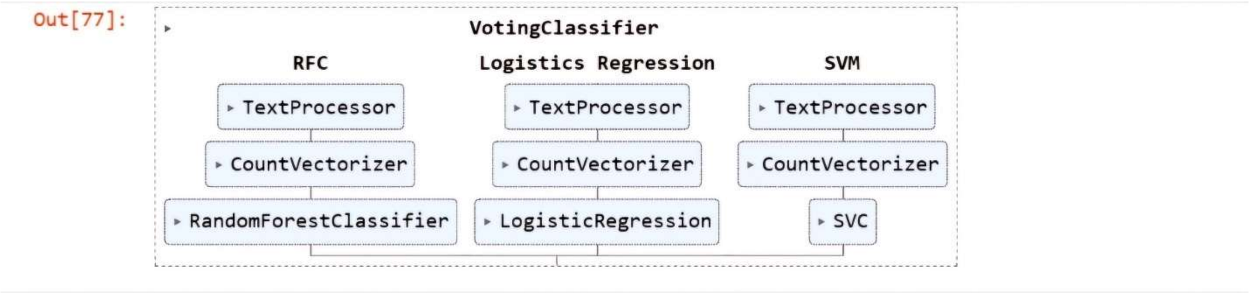
Ensemble methods enhance the performance of traditional machine learning models, offering a balance of simplicity, interpretability, and computational efficiency.

After making pipelines and building machine learning models, next step is of combining models using a Voting Classifier. It improves robustness and performance.
Objective: Combine multiple models to improve robustness and performance.

- **Technique**:
    - **Voting Classifier**: Integrate predictions from multiple models using majority voting or averaging.

- The Voting Classifier achieved high accuracy on both validation and test datasets, demonstrating the effectiveness of combining multiple models.

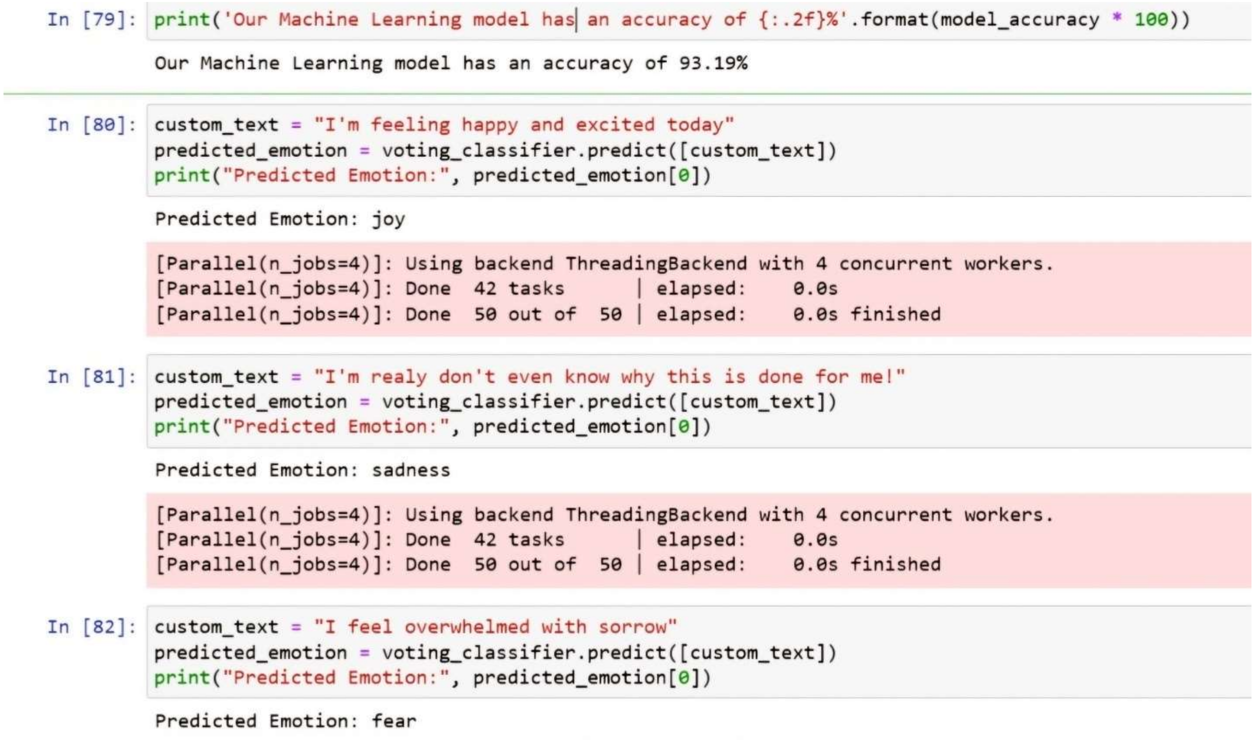
- Comparing individual models and the Voting Classifier shows the benefits of ensemble methods in achieving higher accuracy and robustness.



```
Out[77]:
                         VotingClassifier
        RFC                 Logistics Regression            SVM
    TextProcessor             TextProcessor            TextProcessor
    CountVectorizer           CountVectorizer          CountVectorizer
 RandomForestClassifier      LogisticRegression             SVC
```

Assessing the Accuracy of the Models

Initially after applying the Machine learning Algorithms and NLP techniques on the entire data and assessing its accuracy, some amendments are made to get a better model fit. The Emotions type is composed of 4 binary groups. Consequently, four prominent binary classifiers have been trained, each specializing in one of the emotion aspects. Using the model Voting classifier we predict the values for custom text with the accuracy of 93.19%.



```
In [79]: print('Our Machine Learning model has an accuracy of {:.2f}%'.format(model_accuracy * 100))

         Our Machine Learning model has an accuracy of 93.19%

In [80]: custom_text = "I'm feeling happy and excited today"
         predicted_emotion = voting_classifier.predict([custom_text])
         print("Predicted Emotion:", predicted_emotion[0])

         Predicted Emotion: joy

         [Parallel(n_jobs=4)]: Using backend ThreadingBackend with 4 concurrent workers.
         [Parallel(n_jobs=4)]: Done  42 tasks      | elapsed:    0.0s
         [Parallel(n_jobs=4)]: Done  50 out of  50 | elapsed:    0.0s finished

In [81]: custom_text = "I'm realy don't even know why this is done for me!"
         predicted_emotion = voting_classifier.predict([custom_text])
         print("Predicted Emotion:", predicted_emotion[0])

         Predicted Emotion: sadness

         [Parallel(n_jobs=4)]: Using backend ThreadingBackend with 4 concurrent workers.
         [Parallel(n_jobs=4)]: Done  42 tasks      | elapsed:    0.0s
         [Parallel(n_jobs=4)]: Done  50 out of  50 | elapsed:    0.0s finished

In [82]: custom_text = "I feel overwhelmed with sorrow"
         predicted_emotion = voting_classifier.predict([custom_text])
         print("Predicted Emotion:", predicted_emotion[0])

         Predicted Emotion: fear
```

Conclusion

Summary of Findings

This project demonstrates the application of traditional NLP techniques combined with ensemble learning for emotion detection through text. The Voting Classifier, which integrates multiple machine learning algorithms, achieves the highest accuracy and robustness.

Implications

The findings suggest that ensemble methods can effectively enhance the performance of traditional machine learning models for emotion detection. This approach balances simplicity, interpretability, and computational efficiency, making it suitable for various text classification tasks.

Future Work

Future work could explore the integration of deep learning models with traditional approaches to further improve accuracy. Additionally, expanding the dataset and experimenting with different feature extraction techniques could yield better results.

find my python file here: Emotion detection project
Connect with me on GitHub and LinkedIn:

https://github.com/sweety0423

Email:0423sweety@gmail,com
https://www.linkedin.com/in/sweetysharma04?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=android_app

REFERENCES:

1. Emotions Dataset for NLP: https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp/data

2. Bikram Sarkar, and Joydeep Mukherjee. "Improvement of Feature Engineering on Emotion Detection from Textual Data." World Journal of Advanced Engineering Technology and Sciences, vol. 12, no. 1, 30 May 2024, pp. 073–076, wjaets.com/sites/default/files/WJAETS-2024-0171.pdf, https://wjaets.com/sites/default/files/WJAETS-2024-0171.pdf Accessed 30 May 2024.

3. Dr.KavithaSubramani, et al. "Secured Text-Based Emotion Classification Using Machine Learning with NLP." Educational Administration: Theory and Practice, vol. 30, no. 5, 4 May 2024, pp. 901–910, kuey.net/index.php/kuey/article/view/2986, https://doi.org/10.53555/kuey.v30i5.2986. Accessed 30 May 2024.

4. Gonçalves, Pollyanna, et al. "Comparing and Combining Sentiment Analysis Methods." Proceedings of the First ACM Conference on Online Social Networks - COSN '13, 2013, dl.acm.org/citation.cfm?id=2512951, https://doi.org/10.1145/2512938.2512951.

5. Hung, Lai Po, and Suraya Alias. "Beyond Sentiment Analysis: A Review of Recent Trends in Text-Based Sentiment Analysis and Emotion Detection." Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 27, no. 1, 20 Jan. 2023, pp. 84–95, https://doi.org/10.20965/jaciii.2023.p0084.

6. Meler, Andrzej. In the Beginning, Let There Be the Word: Challenges and Insights in Applying Sentiment Analysis to Social Research. 13 May 2024, https://doi.org/10.1145/3589335.3651264. Accessed 30 May 2024.

7. Mohammad, S. M. (2017). Challenges in sentiment analysis. In Socio-affective computing (pp. 61–83). https://doi.org/10.1007/978-3-319-55394-8_4

8. Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. Social Network Analysis and Mining, 11(1) https://link.springer.com/article/10.1007/s13278-021-00776-6

9. "Sentiment Analysis." Kaggle.com, www.kaggle.com/code/dadaji/sentiment-analysis.

10. Wankhade, Mayur, et al. “A Survey on Sentiment Analysis Methods, Applications, and Challenges.” Artificial Intelligence Review, vol. 55, no. 55, 7 Feb. 2022, link.springer.com/article/10.1007/s10462-022-10144-1, https://doi.org/10.1007/s10462-022-10144-1.

11. Waykar, Yashwant Arjunrao, and Sucheta Yambal. “Unlocking Human Emotions: The Power of Deep Learning in Sentiment Analysis.” Www.igi-Global.com, IGI Global, 2024, www.igi-global.com/chapter/unlocking-human-emotions/347296. Accessed 30 May 2024.

12. Y. A. Solangi, Z. A. Solangi, S. Aarain, A. Abro, G. A. Mallah and A. Shah, "Review on Natural Language Processing (NLP) and Its Toolkits for Opinion Mining and Sentiment Analysis," 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Bangkok, Thailand, 2018, pp. 1-4, https://ieeexplore.ieee.org/abstract/document/8629198

13. https://www.kaggle.com/code/nextmillionaire/emotion-detection-nlp-ml-acc-95/input