TESSA

Text Emotion System Sentiment Analysis

Georgios Ioannou
Farhanul Thouship
Meng Wai Chan
{gioanno000, fthoush000, mchan004}@citymail.cuny.edu

Department of Computer Science, The City College of New York
CSC 44800 Artificial Intelligence
Professor Zhao Yunhua

https://github.com/mengwaichan/CSc448_Final_Project

Abstract

This project explores machine learning models for text emotion detection and classification. We aim to develop an efficient model for identifying emotions in text data, employing natural language processing. The models trained on labeled Twitter(X) datasets find applications in customer service, social media monitoring, marketing analysis, and various other uses. We mainly hope to utilize this emotion detection as a marketing tool to analyze customer perceptions gathered from reviews, tweets, and customer service interactions. This approach will help clients understand their sentiments within their customer base, enabling targeted strategies aligned with their experiences and expectations. We examine and create eight models to determine the most effective approach for emotion detection. We successfully developed our business tool, with the Bidirectional LSTM emerging as the top-performing model with an accuracy of 92.88%.

Introduction

We all deal with emotions. Emotions are pivotal in shaping consciousness and influencing mental processes, with various types corresponding to distinct levels of consciousness. In our daily life, emotions like joy, fear, anger, and sadness are commonly perceived as inner states, creating a deeply personal and complex experience. Understanding our emotions are important but nuances in expressing emotions presents a challenge for machines in comprehending emotions. Understanding emotions can be challenging, especially in the modern era where communication often occurs through screens. Emotions are typically not observed in isolation; they are often intertwined with contextual information, providing additional insights. One source of contextual information is textual content. Analyzing text to discern sentiment allows us to capture and understand the conveyed emotions. Our objective is to develop a machine learning tool that can analyze and understand emotions based on the text received from various sources, such as phone messages, Instagram DMs, or emails, encountered in our daily lives.

In machine learning, recognizing emotions in text poses a content-based classification problem in natural language processing. The complexity arises from diverse human expressions, categorizing emotions as joy, sadness, anger, surprise, hate, and fear. Understanding and identifying these emotions are crucial in applications like chatbots, customer support forums, and various other uses. To address the challenge of sentiment detection, we utilize eight machine learning models, comprising of Bidirectional LSTM, CNN+LSTM, CNN, Multinomial Naive Bayes (MNB), Decision Tree, Random Forest, Support Vector Machine (SVM), and Logistic Regression.

Data

Dataset Link: https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp

```
i wrote two years ago so many things i feel unsure of maybe;fear
i feel suspicious of informality and a lack of credentials;fear
i receive every month make me proud and feel appreciative;joy
i feel that third situation pretty much sums up my feelings toward this title;joy
i wanted to feel him in my hands and reached out to take him into my waiting eager mouth;joy
i feel more gentle that way wth;love
i got home feeling hot tired and great;love
i feel more creative;joy
i feel so talented i can use a computer;joy
i feel unfathomably rich in having had a healthy pregnancy so far;joy
```

The dataset, sourced from the Kaggle, comprises six emotion classes: Anger, Fear, Joy, Love, Sadness, and Surprise. It consists of 16,000 unique text documents for training, 2,000 for validation, and another 2,000 for testing. The dataset offers a diverse range of emotions captured from posts obtained through the Twitter API, making it a valuable resource for natural language processing (NLP) projects focused on emotion analysis.

Preprocessing

In this project we decided to merge all training, testing, and validation dataset and split the dataset in 90:10 ratio to obtain more data for training purposes, hoping to increase the accuracy of our machine learning model. In order to prepare the text data for effective emotion analysis, a series of preprocessing steps were applied. Each step serves a specific purpose to enhance the quality and relevance of the text features for machine learning models.

Text Cleaning - Text data often contains a variety of characters, including special symbols and numbers, which may not contribute significantly to the understanding of emotions. Therefore, the text was converted to lowercase to ensure uniformity and to facilitate subsequent processing steps.

Stop Word Removal - Common words that occur frequently in a language, known as stop words, were removed from the text. These words, such as "and," "the," and "is," typically do not carry significant meaning in the context of emotion analysis and can be safely omitted.

Number Removal - Numeric characters were removed from the text as they may not contribute meaningfully to emotion analysis. By eliminating numbers, the focus remains on the emotional content conveyed through textual information.

Punctuation Removal - Punctuation marks were removed from the text to simplify the language and ensure that the machine learning models focus on the emotional content of the text rather than being influenced by punctuation.

URL Removal - Any URLs present in the text were removed as they are unlikely to contribute meaningful information to the emotion analysis. This step ensures that the models are trained on the emotional expressions conveyed through the textual content.

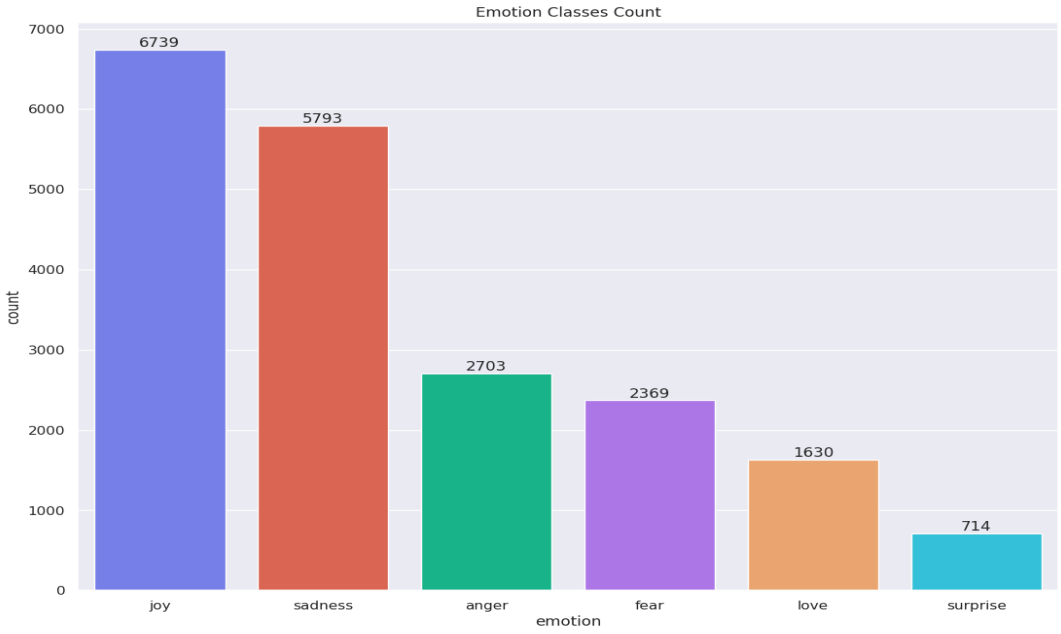
Lemmatization - Lemmatization was applied to reduce words to their base or root form. This step aids in standardizing the text and capturing the core meaning of words, thus enhancing the effectiveness of the subsequent machine learning models.

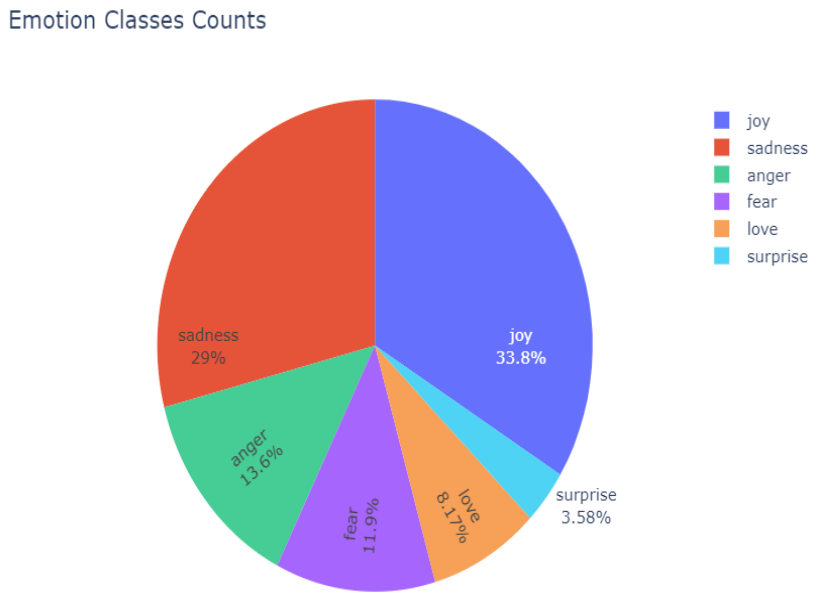
By implementing these preprocessing steps, the text data has been refined to emphasize emotional content while removing noise and irrelevant information. This clean and standardized textual input is now ready for further analysis using machine learning models for emotion classification. The following is a sample of the processed dataset.

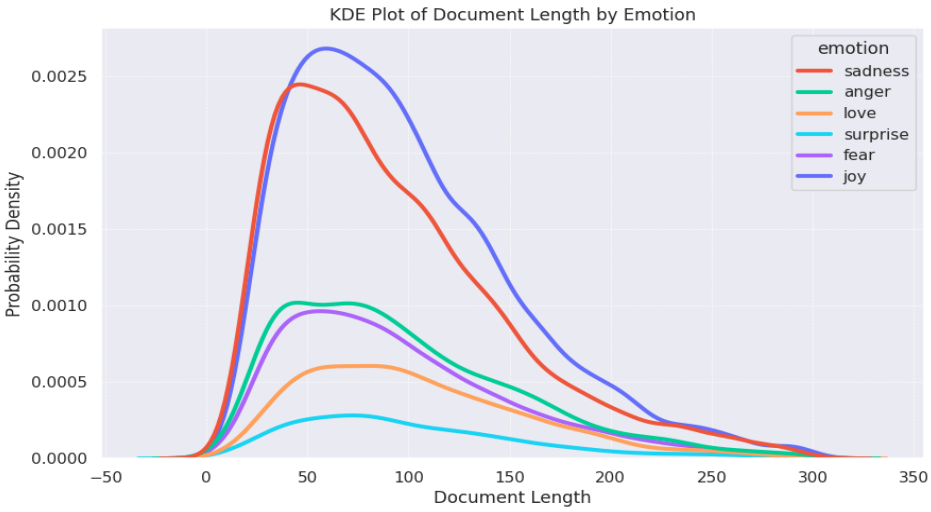
	document	emotion	document_length	document_clean
0	i didnt feel humiliated	sadness	23	didnt feel humiliated
1	i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	sadness	108	go feeling hopeless damned hopeful around someone care awake
2	im grabbing a minute to post i feel greedy wrong	anger	48	im grabbing minute post feel greedy wrong
3	i am ever feeling nostalgic about the fireplace i will know that it is still on the property	love	92	ever feeling nostalgic fireplace know still property
4	i am feeling grouchy	anger	20	feeling grouchy
...	...	...	...	...
1995	i just keep feeling like someone is being unkind to me and doing me wrong and then all i can think of doing is to get back at them and the people they are close to	anger	163	keep feeling like someone unkind wrong think get back people close
1996	im feeling a little cranky negative after this doctors appointment	anger	66	im feeling little cranky negative doctor appointment
1997	i feel that i am useful to my people and that gives me a great feeling of achievement	joy	85	feel useful people give great feeling achievement
1998	im feeling more comfortable with derby i feel as though i can start to step out my shell	joy	88	im feeling comfortable derby feel though start step shell
1999	i feel all weird when i have to meet w people i text but like dont talk face to face w	fear	86	feel weird meet w people text like dont talk face face w

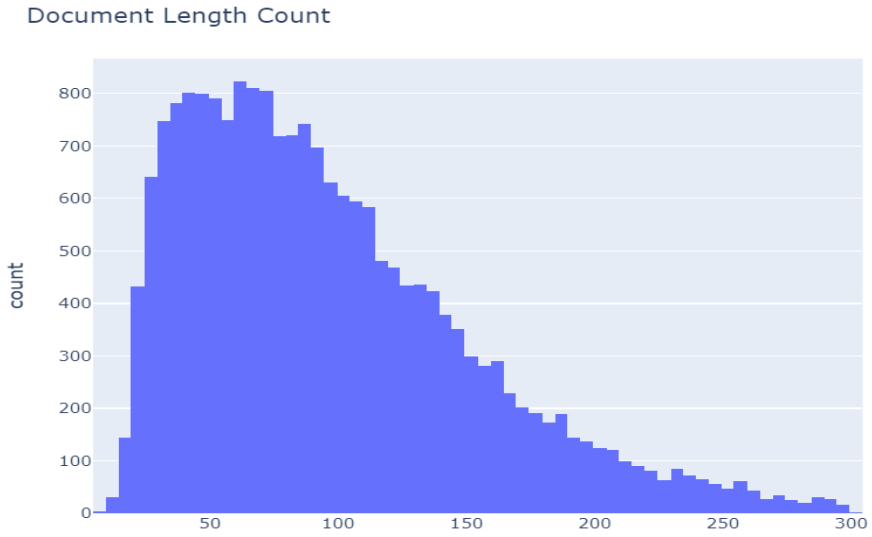
In order to transform the processed text into a format suitable for machine learning models, text vectorization was employed. This process involves converting the data into numerical vectors that can be understood and utilized by machine learning algorithms.

Visualization / EDA:





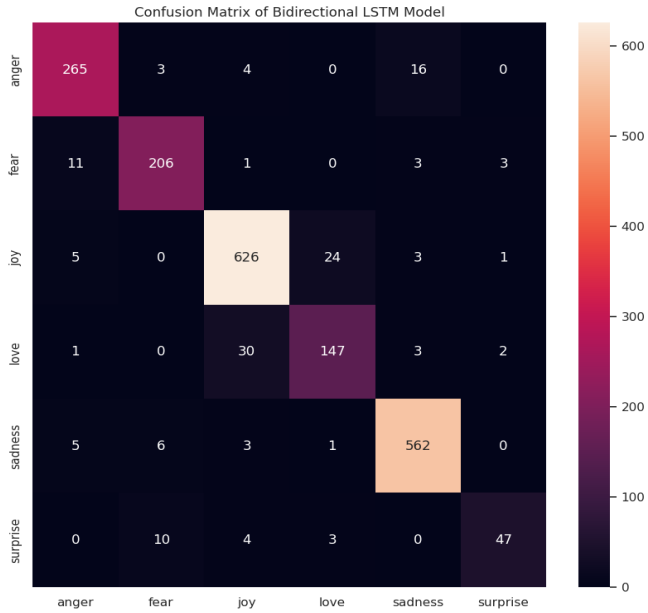




Machine Learning Models:

With the preprocessed data, we apply eight machine learning models to understand and classify emotions in text, by using eight models we are able to analyze which machine learning model best fit for this emotions dataset.

Bidirectional Long Short-Term Memory (BiLSTM)

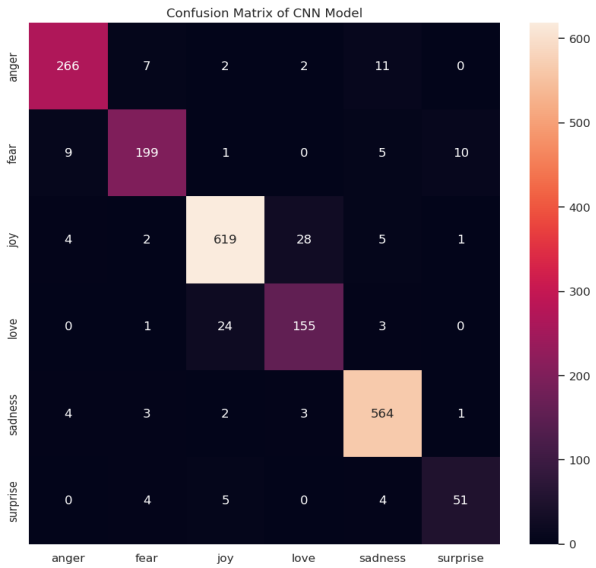


Model Classification Report

	precision	recall	f1-score	support
anger	0.92	0.92	0.92	288
fear	0.92	0.92	0.92	224
joy	0.94	0.95	0.94	659
love	0.84	0.80	0.82	183
sadness	0.96	0.97	0.97	577
surprise	0.89	0.73	0.80	64
accuracy			0.93	1995
macro avg	0.91	0.88	0.90	1995
weighted avg	0.93	0.93	0.93	1995

Bidirectional Long Short-Term Memory (BiLSTM) is a sequential data processing architecture that utilizes both forward and backward information flow to capture contextual dependencies effectively, making it well-suited for tasks like language processing and time series analysis.

Convolutional Neural Network (CNN)

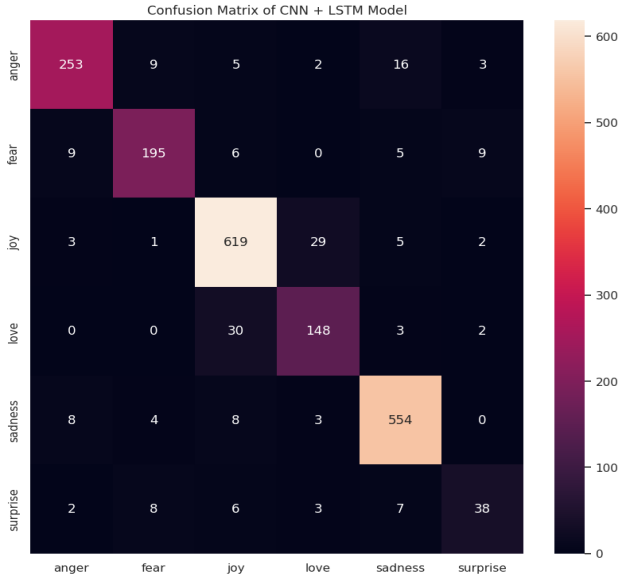


Model Classification Report

	precision	recall	f1-score	support
anger	0.94	0.92	0.93	288
fear	0.92	0.89	0.90	224
joy	0.95	0.94	0.94	659
love	0.82	0.85	0.84	183
sadness	0.95	0.98	0.96	577
surprise	0.81	0.80	0.80	64
accuracy			0.93	1995
macro avg	0.90	0.90	0.90	1995
weighted avg	0.93	0.93	0.93	1995

Convolutional Neural Network (CNN) is a deep learning architecture designed for processing grid-like data. It uses convolutional layers to automatically learn hierarchical representations, enabling effective feature extraction and pattern recognition.

CNN+LSTM

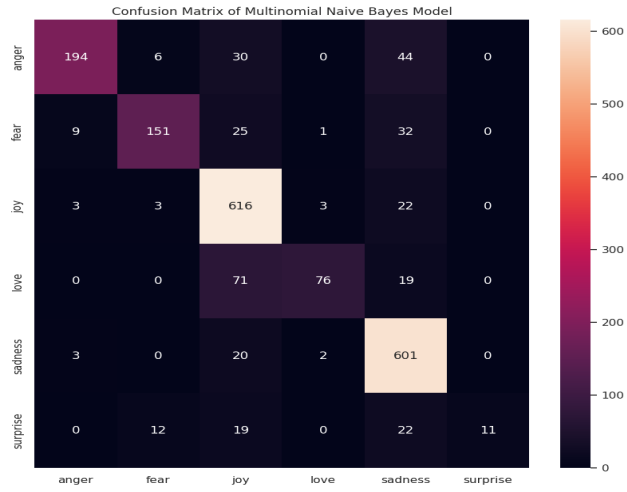


Model Classification Report

	precision	recall	f1-score	support
anger	0.92	0.88	0.90	288
fear	0.90	0.87	0.88	224
joy	0.92	0.94	0.93	659
love	0.80	0.81	0.80	183
sadness	0.94	0.96	0.95	577
surprise	0.70	0.59	0.64	64
accuracy			0.91	1995
macro avg	0.86	0.84	0.85	1995
weighted avg	0.90	0.91	0.90	1995

CNN+LSTM is a hybrid neural network architecture combining Convolutional Neural Networks (CNNs) for spatial feature extraction with Long Short-Term Memory networks (LSTMs) for sequential information processing, making it effective for tasks involving both spatial and temporal dependencies.

MNB

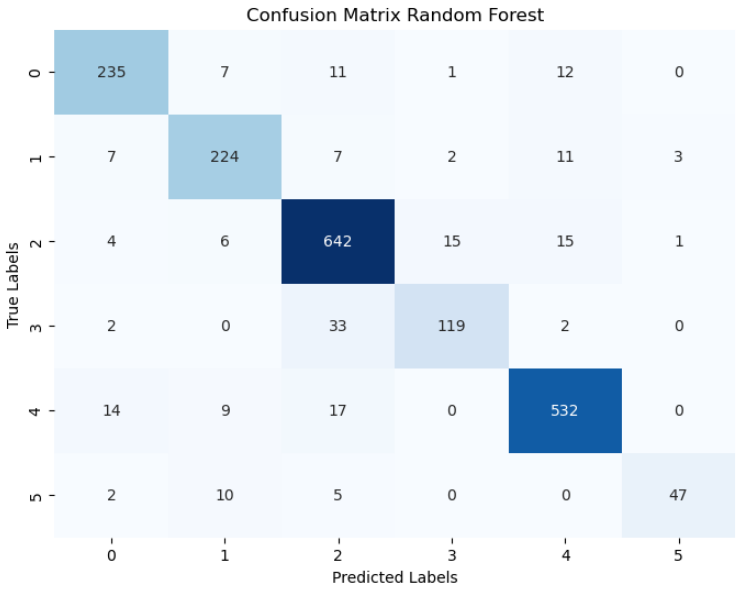


Model Classification Report

	precision	recall	f1-score	support
anger	0.93	0.71	0.80	274
fear	0.88	0.69	0.77	218
joy	0.79	0.95	0.86	647
love	0.93	0.46	0.61	166
sadness	0.81	0.96	0.88	626
surprise	1.00	0.17	0.29	64
accuracy			0.83	1995
macro avg	0.89	0.66	0.70	1995
weighted avg	0.84	0.83	0.81	1995

Multinomial Naive Bayes (MNB) is a probabilistic classification algorithm commonly used for text categorization. It models the likelihood of observing a particular set of words in a document given its class and makes predictions based on the maximum likelihood estimation of class probabilities.

Random Forest

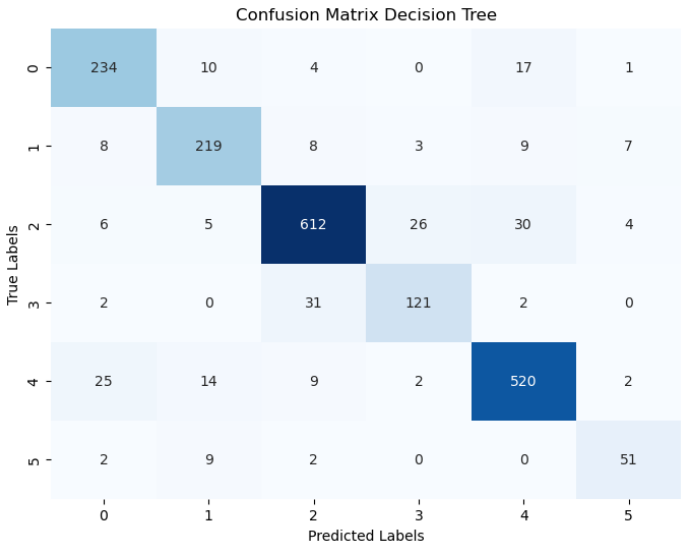


	precision	recall	f1-score	support
0	0.89	0.88	0.89	266
1	0.88	0.88	0.88	254
2	0.90	0.94	0.92	683
3	0.87	0.76	0.81	156
4	0.93	0.93	0.93	572
5	0.92	0.73	0.82	64
accuracy			0.90	1995
macro avg	0.90	0.86	0.87	1995
weighted avg	0.90	0.90	0.90	1995

Name: Random Forest
Accuracy: 0.9017543859649123
Precision: 0.9016448509720978
Recall: 0.9017543859649123
F1: 0.900923733903796

Random Forest is an ensemble learning algorithm that operates by constructing a multitude of decision trees during training and outputting the mode of the classes or mean prediction of the individual trees. Each tree in the Random Forest is constructed using a random subset of the training data and a random subset of features, introducing diversity and reducing overfitting. This ensemble approach results in a robust and accurate model.

Decision Tree

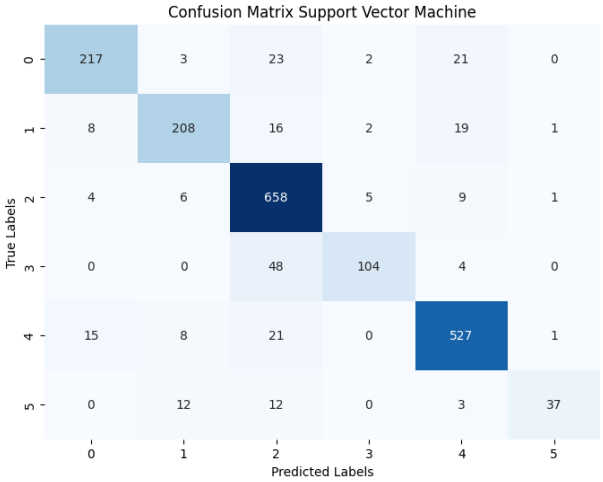


	precision	recall	f1-score	support
0	0.84	0.88	0.86	266
1	0.85	0.86	0.86	254
2	0.92	0.90	0.91	683
3	0.80	0.78	0.79	156
4	0.90	0.91	0.90	572
5	0.78	0.80	0.79	64
accuracy			0.88	1995
macro avg	0.85	0.85	0.85	1995
weighted avg	0.88	0.88	0.88	1995

Name: Decision Tree
Accuracy: 0.8807017543859649
Precision: 0.881089951820483
Recall: 0.8807017543859649
F1: 0.8807766265759057

A Decision Tree is a tree-like model that recursively splits the dataset based on feature values, enabling hierarchical decision-making and providing a transparent representation of the decision process.

Support Vector Machine (SVM)

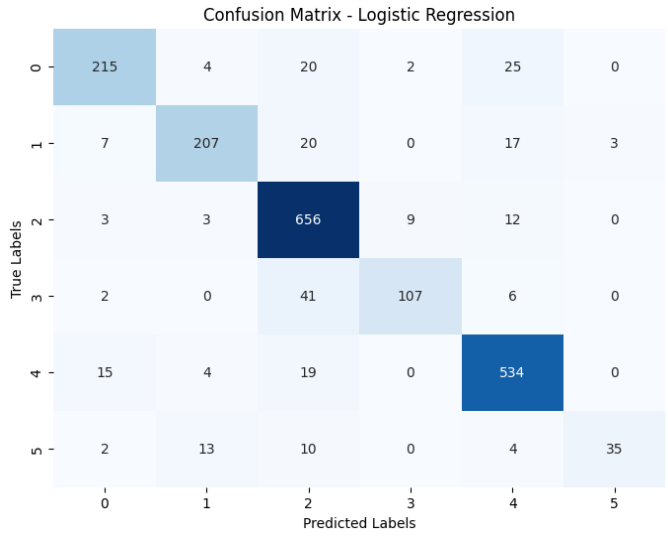


	precision	recall	f1-score	support
0	0.89	0.82	0.85	266
1	0.88	0.82	0.85	254
2	0.85	0.96	0.90	683
3	0.92	0.67	0.77	156
4	0.90	0.92	0.91	572
5	0.93	0.58	0.71	64
accuracy			0.88	1995
macro avg	0.89	0.79	0.83	1995
weighted avg	0.88	0.88	0.87	1995

```
Name: SVM
Accuracy: z 0.8776942355889724
Precision:  0.8806868101778665
Recall:  0.8776942355889724
F1:  0.8746469171434184
```

Support Vector Machine (SVM) is a supervised machine learning algorithm that seeks to find an optimal hyperplane to separate data points into different classes, maximizing the margin between classes.

Logistic Regression



	precision	recall	f1-score	support
0	0.88	0.81	0.84	266
1	0.90	0.81	0.85	254
2	0.86	0.96	0.91	683
3	0.91	0.69	0.78	156
4	0.89	0.93	0.91	572
5	0.92	0.55	0.69	64
accuracy			0.88	1995
macro avg	0.89	0.79	0.83	1995
weighted avg	0.88	0.88	0.88	1995

```
Name: Logistic Regression
Accuracy: 0.8792
Precision: 0.8813
Recall: 0.8792
F1: 0.8759
```

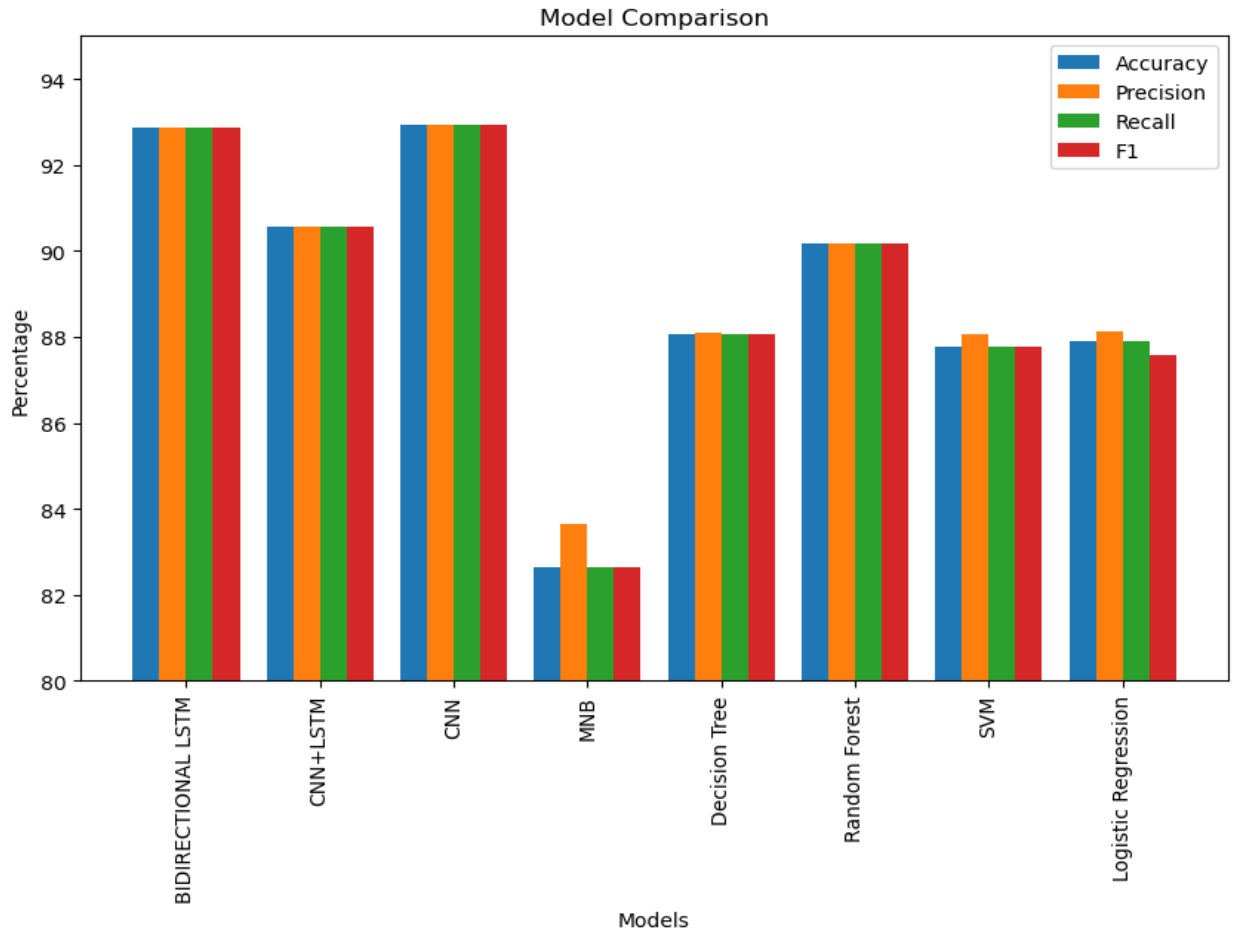
Logistic Regression is a statistical model that predicts the probability of binary outcomes by fitting the data to a logistic curve, making it suitable for classification tasks.

Model Performance

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
BiLSTM	92.882206	92.882206	92.882206	92.882206
CNN	92.932331	92.932331	92.932331	92.932331
CNN+LSTM	90.576441	90.576441	90.576441	90.576441
Decision Tree	88.070175	88.10899	88.070175	88.077662
Logistic Regression	87.92	88.13	87.92	87.59
MNB	82.6566	83.6566	82.6566	82.6566
Random Forest	90.17543	90.16448	90.175438	90.09237
SVM	87.769423	88.06868	87.76942	87.46469

The table illustrates the accuracy, precision, recall, and F1 score for all our machine learning models on sentiment analysis. The Bidirectional LSTM performed well across all metrics with an accuracy, precision, recall, and F1 score of 92.88%. The CNN+LSTM model closely follows with 90.58% across all metrics. The standalone CNN model performs the best with an accuracy, precision, recall, and F1 score of 92.93%. The Multinomial Naive Bayes (MNB) model achieves an accuracy of 82.66% with consistent precision, recall, and F1 scores. The Decision Tree and Random Forest models show strong performance, with the Decision Tree achieving 88.07% accuracy and the Random Forest reaching 90.18%. The Support Vector Machine (SVM) exhibits an accuracy of 87.77%, while the Logistic Regression model achieves

an accuracy of 87.92% with corresponding precision, recall, and F1 scores of 88.13%, 87.92%, and 87.59%, respectively. Overall, CNN stands out as our top-performing model for our project.



Analysis

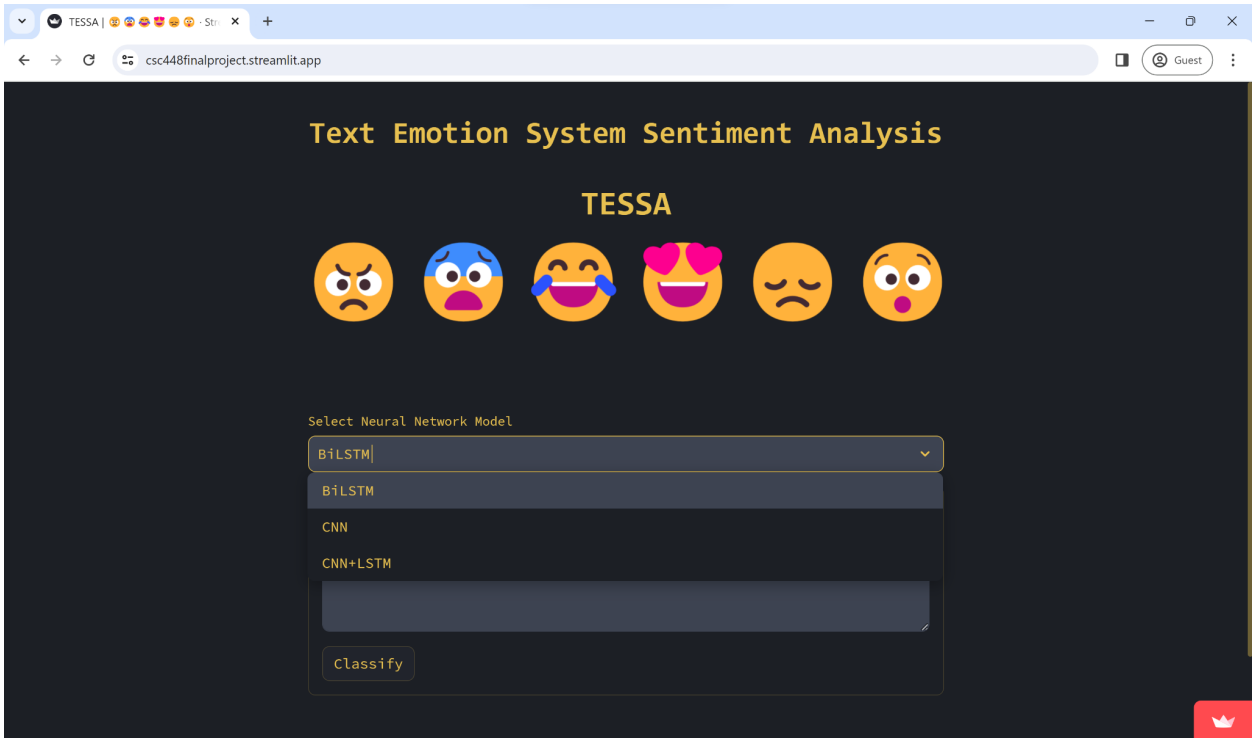
The variation in performance among the different models can be attributed to the inherent differences in their architectures and underlying algorithms. The Bidirectional LSTM excels with a well-balanced combination of accuracy, precision, recall, and F1 score at 92.88%. Its bidirectional nature enables it to capture contextual information from both past and future tokens, making it effective in understanding sequential patterns within the sentiment data. The CNN and CNN+LSTM models also perform impressively, achieving accuracies of 92.93% and 90.58%,

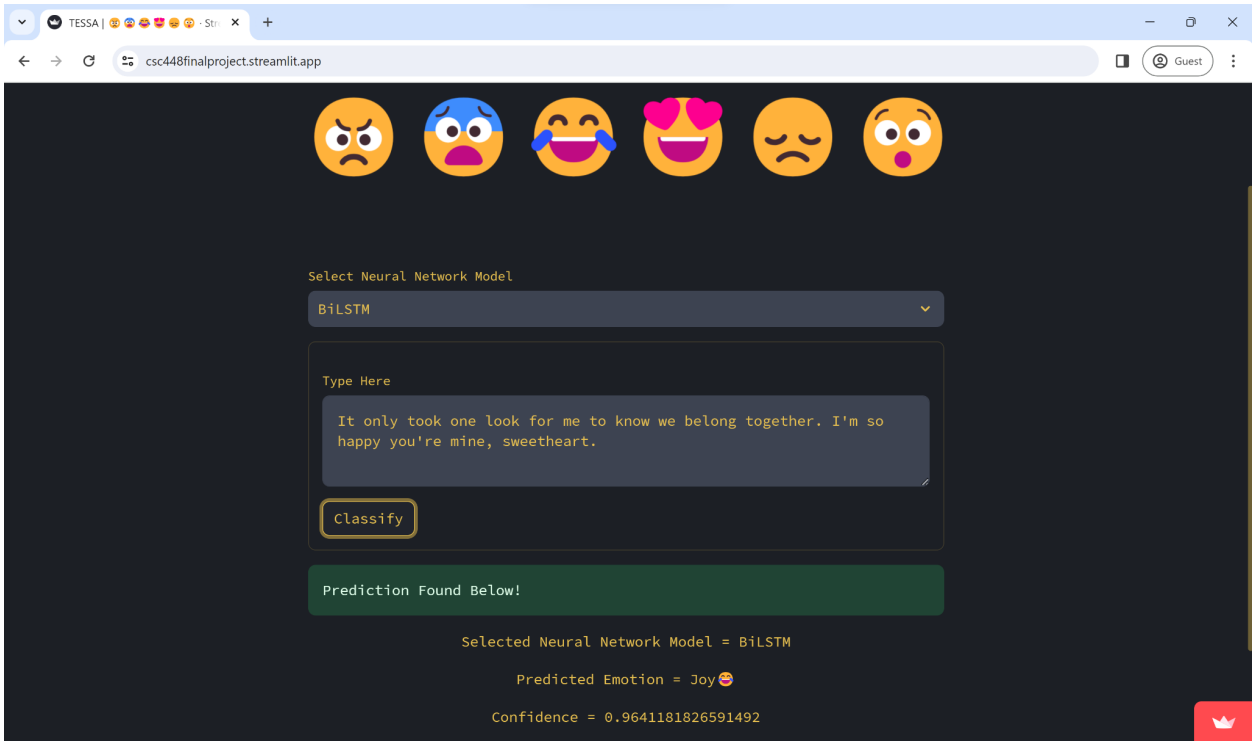
respectively. Convolutional Neural Networks (CNNs) are adept at capturing local patterns, while the combination of CNN and LSTM allows for both local and sequential feature extraction, contributing to their robust performance in sentiment analysis. Emotion detection in text is best relying on diverse neural network architectures, with outcomes tied to data characteristics. CNNs excel in capturing spatial hierarchies efficiently, while LSTMs focus on crucial temporal dependencies. Results underscore the importance of considering both spatial and temporal aspects, emphasizing the need to tailor architecture to the dataset.

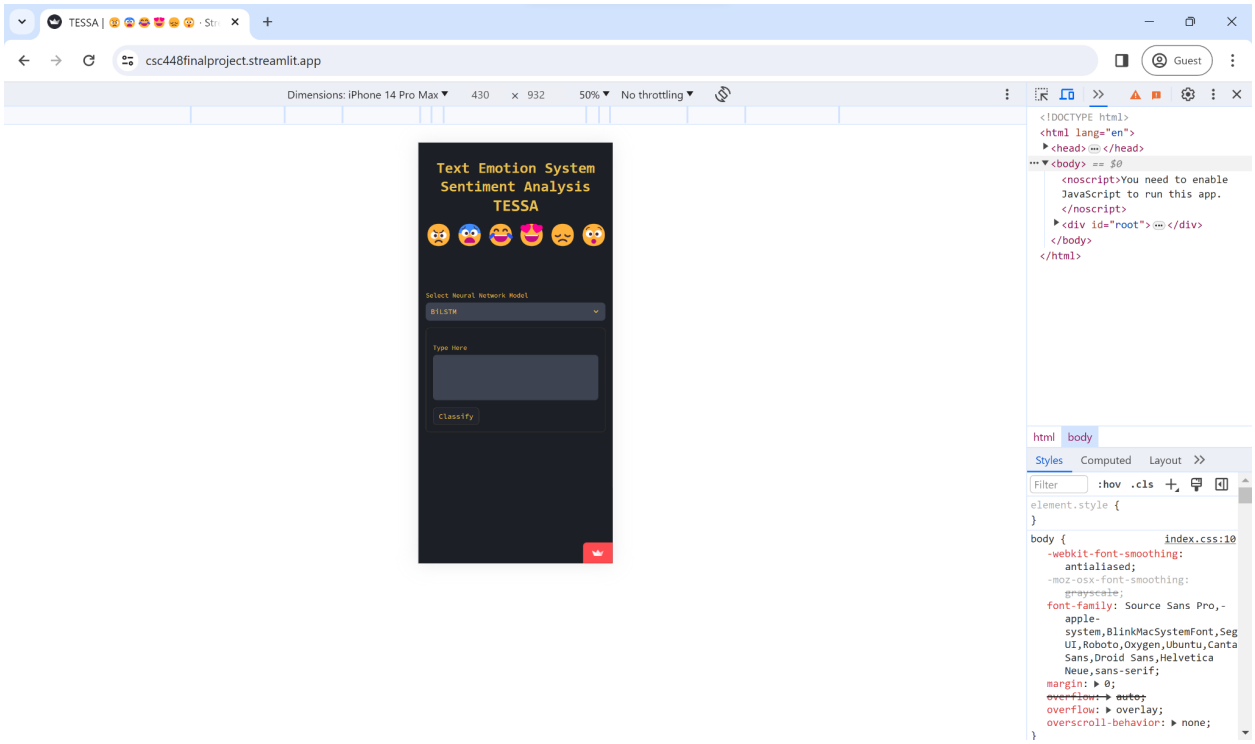
On the other hand, the Multinomial Naive Bayes (MNB) model lags behind with an accuracy of 82.66%. MNB relies on the assumption of feature independence, which may not hold true for sentiment analysis where the order and context of words are crucial. The Decision Tree and Random Forest models show strong performances at 88.07% and 90.18%, respectively, by leveraging hierarchical decision structures and ensemble learning. Support Vector Machines (SVM) and Logistic Regression, both achieving accuracies around 87-88%, may face challenges in capturing nuanced patterns in the sentiment data. Further model fine-tuning and feature engineering may be required to enhance their performance. In conclusion, the varied model performances highlight the importance of selecting models that align with the inherent characteristics of the sentiment analysis task, considering factors such as sequential dependencies, feature independence assumptions, and the ability to capture intricate patterns.

Application User Interface

Nowadays, having only state of the art models is not enough. We must have a way that will allow users to interact, test, and give feedback on our models. Therefore, we decided to build a simple web application for this Project where users can test our Neural Network models on their documents. Below you will find illustrations of how our application works. The application is also mobile responsive and it was built using Streamlit. Please feel free to test our application, here is the live deployed link: https://csc448finalproject.streamlit.app/


TESSA
csc448finalproject.streamlit.app
Guest
Text Emotion System Sentiment Analysis
TESSA
Select Neural Network Model
BiLSTM
BiLSTM
CNN
CNN+LSTM
Classify


TESSA
csc448finalproject.streamlit.app
Guest
Select Neural Network Model
BiLSTM
Type Here
It only took one look for me to know we belong together. I'm so happy you're mine, sweetheart.
Classify
Prediction Found Below!
Selected Neural Network Model = BiLSTM
Predicted Emotion = Joy
Confidence = 0.9641181826591492



Summary

Objective: The aim for our project was to explore and develop machine learning models for text emotion detection and classification, focusing on recognizing emotions like joy, sadness, anger, surprise, hate, and fear.

Dataset: The Kaggle dataset features six emotion classes and includes 16,000 unique training texts, along with 2,000 for validation and testing each. Derived from Twitter API posts, it is a valuable resource for emotion analysis in NLP projects.

Preprocessing: Text data underwent essential preprocessing steps, including cleaning, stop word removal, number removal, punctuation removal, URL removal, and lemmatization, refining it for emotion analysis. The resulting processed dataset, prepared for machine learning models, is transformed into numerical vectors through text vectorization for effective analysis.

Models: To tackle the challenge of sentiment detection, we utilize eight machine learning models, comprising Bidirectional LSTM, CNN+LSTM, CNN, Multinomial Naive Bayes (MNB), Decision Tree, Random Forest, Support Vector Machine (SVM), and Logistic Regression.

Model Performance: The Bidirectional LSTM achieves a leading performance with 92.88% accuracy, followed closely by CNN+LSTM at 90.58%. The CNN outperforms both with 92.93% accuracy, establishing CNN as the top-performing model in our project, while the rest of the models trail.

Analysis: Model performance variation arises from inherent architectural differences. The Bidirectional LSTM excels with a balanced combination of metrics, leveraging bidirectionality for effective sequential pattern understanding. CNN and CNN+LSTM also perform well, capturing local and sequential features, highlighting the importance of diverse neural network architectures. The remaining models trail these top performers, emphasizing the need for architecture tailored to dataset characteristics.

Application User Interface: Our state of the art and user friendly application serves as a playground for the users to interact with our Neural Network models and test the accuracy of our models. Our application allows the user to select from a selection of models, input the document, and finally classify the document.

Conclusion: Overall we successfully developed a highly accurate emotion detection tool for marketing, analyzing customer sentiments from reviews, tweets, and customer interactions. This tool empowers clients using high accuracy models to understand and align strategies with their customer base's sentiments, enhancing targeted approaches.