SENTIMENT ANALYSIS

ABSTRACT

Imagine a world where computers can understand the emotions we express through text. This ability has vast applications in the real world, allowing machines to analyze customer reviews, social media sentiments and public opinions. This paper investigates the effectiveness of various Machine Learning (ML) and Deep Learning (DL) techniques for emotion recognition in textual data. We explore the application of traditional ML models like Naive Bayes Classifiers and Decision Trees alongside advanced DL architectures such as Recurrent Neural Networks (RNNs) and Long Short Term Memory Networks (LSTMs) for this purpose. The paper analyzes the strengths and weaknesses of each approach, comparing their performance in identifying emotions like happiness, sadness, anger, fear, love, hate, gratitude, excitement and surprise. We evaluate the models based on metrics like accuracy, precision and recall.From gauging public opinion on social media to personalizing customer experiences, emotion detection in text data holds immense potential. This comparative analysis investigates the most effective methodologies for this task, paving the way for advancements in sentiment analysis and its far-reaching applications in understanding large-scale emotions, opinions, and information extraction.

Table of Contents

INTRODUCTION

Imagine a future in which words can express feelings that computers can understand. With this skill, robots can now assess customer evaluations, social media sentiments, and public opinions, which has many real-world implications. This paper investigates how well different Deep Learning (DL) and Machine Learning (ML) methods identify emotions in textual data. We study the application of state-of-the-art DL architectures like Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs) alongside more conventional ML models like Decision Trees and Naive Bayes Classifiers.

This essay evaluates the benefits and drawbacks of each strategy, contrasting how well they identify feelings like joy, sorrow, fear, rage, love, hate, thankfulness, excitement, and surprise. We assess the models using criteria such as recall, accuracy, and precision. Emotion recognition in text data has enormous promise for applications ranging from tailoring consumer experiences to monitoring public opinion on social media. By determining which approaches work best for this kind of work, this comparative research opens the door to further developments in sentiment analysis and its many uses in comprehending large-scale emotions, views, and information extraction.

Furthermore, the benefits of effective emotion recognition go beyond commercial and social media applications. In healthcare, for example, emotion detection can aid in the early detection of mental health disorders by analysing patients' online messages and giving prompt intervention and support. Understanding student emotions through written feedback can help educators modify their teaching approaches to better meet students' needs and enhance learning outcomes.

Moreover, in the field of entertainment, emotion detection can improve user experiences by tailoring content recommendations to the viewer's emotional state. Sentiment analysis can help news companies evaluate public reaction to events and modify their coverage to better serve their audience's interests.

Ultimately, this study not only advances the academic field of sentiment analysis but also provides practical solutions to real-world challenges. By discovering the best successful emotion detection approaches, we can improve a variety of applications that rely on interpreting human emotions in text, thereby improving communication, interaction, and decision-making across domains. This thorough study lays the framework for future advances in emotion detection technology, pointing to a future in which robots can better comprehend and respond to human emotions, resulting in more sympathetic and responsive digital environments.

FUNDAMENTALS and LITERATURE SURVEY(THEORY):

Reference	Paper	Methodologies	Pros	Cons
[1]	Emotion detection from text. In the proceedings of the Data Mining and Knowledge Management Process	- The Authors [1] introduce an Emotion Detector Algorithm that calculates emotion scores by considering parameters such as parent-child relationships, depth in ontology, and frequency in the text document [3]. - The authors review existing methodologies, such as Keyword Spotting, Lexical Affinity, Learning-based Methods, and Hybrid Methods, identifying their limitations [2]. - The Emotion Ontology is developed using W.G. Parrot's emotion hierarchy, and the Emotion Detector Algorithm calculates	- This algorithm aims to overcome the limitations of existing methods, offering a more nuanced and context-aware approach to text-based emotion detection [2]. -Improved-performa nce. - This research paper [1] explores the domain of Emotion Detection in text documents, emphasizing its significance in human-computer interaction. - Structured approach to understanding emotions.	- Complexity in developing and maintaining the ontology. - Existing methods may not capture the context effectively. -Requires extensive domain knowledge and manual effort.

Reference	Paper	Methodologies	Pros	Cons
		emotion scores based on ontology parameters, providing a final emotion classification for the text [3].		
[4]	Natural language processing: state of the art, current trends and challenges.	- The Authors [4] have discussed the impact of CNNs and RNNs, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), in handling sequential data such as text [5]. -Attention mechanisms and transformers further enhanced NLP capabilities, with recent developments like Transformer-XL addressing the challenge of learning longer-term dependencies [6]. - The paper provides a comprehensive overview of the role played by various neural architectures in advancing NLP.	- Enhanced handling of sequential data. - Ability to learn longer-term dependencies. - Comprehensive overview of NLP advancements. - Addresses the challenges of mixed-language processing. - Effective for extracting meaningful information from social media feeds.	-High-computational resources required. - Integration of multiple tools can be complex and resource-intensive - May not generalize well to other mixed-language scenarios. - Social media data can be noisy and unstructured.

Reference	Paper	Methodologies	Pros	Cons
		- The significance of tools and systems in shaping NLP is emphasized, with specific mention of Sentiment Analyzers, Parts of Speech (POS) Taggers, Chunking, Named Entity Recognition (NER), Emotion Detection, and Semantic Role Labeling [9]. Sentiment analysis, as illustrated by Nasukawa et al., involves extracting sentiments about a given topic, utilizing sentiment lexicons and pattern databases. [9]. - Emotion detection is explored in the context of mixed-language scripts (Hinglish), showcasing the integration of machine learning and human knowledge for sentiment analysis [8].		

Reference	Paper	Methodologies	Pros	Cons
		- Semantic Role Labeling and Event Discovery in social media feeds are also discussed, showcasing the diverse applications of NLP [7].		
[10]	Performance Evaluation of Supervised Machine Learning Techniques for Efficient Detection of Emotions from Online Content.	- The Authors [10] focus on classifying text-based emotions using various machine learning classifiers, including Naïve Bayesian, Decision Tree, KNN, Support Vector Machine, and logistic regression, implemented in an NLTK-based Python framework [12] - Feature engineering involves count vector creation and TF-IDF calculation to represent the importance of terms in documents [13]. - The study evaluates the classifiers' performance based on precision, recall, F1-score, and accuracy using a dataset divided into training and testing blocks [13].	-Logistic-Regression outperformed other classifiers, achieving higher accuracy (66.58%), recall (avg) (67), and precision (avg) (67) [13]. -Logistic-Regression performs well across multiple metrics, with the highest accuracy (66.58%) and recall for specific emotion tags [11]. - Effective feature engineering approaches.	-K Nearest Neighbour (KNN) demonstrated the lowest performance with precision (avg) (0.58), recall (avg) (0.58), F1-score (avg) (0.57), and accuracy (avg) (57.81%) [13]. - Some classifiers may not handle large feature spaces well. - KNN consistently performed poorly, indicating limited applicability for this ta

Reference	Paper	Methodologies	Pros	Cons
		-Further Experiments: Additional classifiers such as Random Forest, XGBoost, Stochastic Gradient, BPN. - It discusses the accuracy of classifiers concerning different emotion signals and presents detailed experimental results, showcasing Logistics Regression as the classifier with the best performance and KNN with the worst performance [14].		

PROBLEM STATEMENT AND PROBLEM OBJECTIVES

Problem Statement

In today's world, where we're constantly connected online, it's really important to understand how people feel when they write things. Sentiment analysis, a part of understanding how computers interpret human language, helps determine the emotions behind what people write. This research project aims to explore sentiment analysis in detail. We'll look at why it's important, how it works, and how we can use it in today's world where data is ubiquitous.

The exponential rise of social media platforms, e-commerce, and online communication has ushered in an unprecedented volume of textual data. Within this vast expanse of text lie invaluable insights into human emotions, opinions, and attitudes. Sentiment analysis serves as the compass to navigate through this trove of information, unveiling sentiments ranging from joy, anger, and sadness to neutrality.

Despite the vast potential of sentiment analysis, several challenges impede its effective implementation. One major issue is the ambiguity and complexity of human language. Sarcasm, idioms, and context-dependent expressions often confound sentiment analysis models, leading to inaccurate results. For instance, a sarcastic remark like "Great job!" following a failure can be misinterpreted as positive feedback by a simplistic sentiment analysis model.

Another significant problem is the variability in language use across different demographics and regions. Slang, dialects, and cultural references can drastically alter the meaning of text, posing a challenge for sentiment analysis systems trained on standard language datasets. Additionally, multilingual sentiment analysis remains a daunting task, as models need to understand and process texts in various languages with equal proficiency.The presence of mixed emotions in a single piece of text further complicates the analysis. A review that expresses both appreciation and criticism, for instance, can be challenging to classify accurately. Current models often struggle to handle such nuances, leading to oversimplified and sometimes misleading sentiment classifications.

A critical challenge in sentiment analysis is the machine's inability to fully understand context. Contextual information is crucial in interpreting the true meaning of a sentence. For example, the phrase "I love this movie, it's a classic!" is straightforwardly positive. However, the phrase "I love how this always happens to me" could be negative if it’s said in a context of frequent misfortune. Without understanding the surrounding context, sentiment analysis models may produce incorrect sentiment classifications, reducing their effectiveness. This lack of context comprehension becomes even more problematic when dealing with complex and longer texts, where the sentiment can shift multiple times within the same passage.

Additionally, there is the challenge of maintaining model relevance over time. Language and expression evolve continuously, with new slang and trends emerging regularly. Models trained on older data may become less effective as language use changes. This requires ongoing updates and retraining of sentiment analysis models to ensure they remain accurate and relevant in capturing current sentiment expressions.

Project Objectives

This research project sets out to address these challenges by developing a more nuanced and comprehensive approach to sentiment analysis. The objectives of this project include:

1. **Enhancing Model Accuracy**: Improve the accuracy of sentiment analysis models by incorporating advanced natural language processing (NLP) techniques. This includes leveraging context-aware models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), which can better understand the subtleties of human language.

2. **Handling Sarcasm and Contextual Nuances**: Develop algorithms capable of detecting sarcasm and other context-dependent expressions. This will involve training models on a diverse dataset that includes examples of sarcasm, idioms, and other complex linguistic constructs. Additionally, implementing methods to enhance contextual understanding in

AI models will be crucial. Techniques such as attention mechanisms and transformer architectures will be explored to improve context comprehension in sentiment analysis.

3. **Managing Mixed Emotions**: Implement techniques to accurately identify and classify mixed emotions within a single text. This might include multi-label classification approaches that allow a text to be tagged with multiple sentiments simultaneously. For example, advanced machine learning techniques, such as hierarchical attention networks, can help discern the primary sentiment while still acknowledging secondary emotions present in the text.

4. **Maintaining Model Relevance**: Develop strategies to keep sentiment analysis models up-to-date with evolving language trends. This involves continuously retraining models on recent data to ensure they capture new slang, expressions, and trends accurately. Implementing automated processes for data collection and model updating will be key to maintaining the relevance and accuracy of sentiment analysis tools.

5. **Evaluating Performance**: Conduct comprehensive evaluations of the proposed models using various metrics such as accuracy, precision, recall, and F1 score. This will help in identifying the strengths and weaknesses of each approach and refining the models accordingly. Detailed performance analysis will include cross-validation techniques and robust statistical testing to ensure reliability.

6. **To Assess Model Performance**: Another aim is to rigorously evaluate the performance of the enhanced sentiment analysis models to determine their effectiveness in handling the unique challenges presented by textual data. This evaluation extends beyond traditional metrics, seeking to comprehensively understand how well these models navigate and overcome the unique challenges presented by textual data, especially within the dynamic and evolving landscape of social media.

7. **To Provide Data-Driven Insights**: This research aims to provide businesses and organizations with data-driven insights into consumer attitudes, opinions, and sentiment towards their brands, products, or services as expressed on social media apps. This facet of the study seeks to unravel patterns and trends within the vast pool of user-generated content, offering valuable information that can inform strategic decision-making processes for brands, products, or services.

8. **To Facilitate Informed Decision-Making**: By achieving the above objectives, this study aims to empower businesses and organizations to make more informed decisions based on a deeper understanding of the sentiments and emotions prevalent in social media discussions.

By addressing these objectives, this project aims to overcome the limitations of current sentiment analysis schemes and contribute to the development of more effective, accurate, and ethical sentiment analysis technologies. Through rigorous research and innovative methodologies, we seek to unlock the full potential of sentiment analysis in understanding and interpreting the vast array of human emotions expressed in digital communication. This project will not only advance the academic field of sentiment analysis but also offer practical solutions to real-world challenges, fostering more empathetic and effective interactions in an increasingly connected world. Ultimately, this research aspires to bridge the gap between human emotional complexity and machine understanding, paving the way for more sophisticated and human-centered AI applications.

PROPOSED WORK

The Methodology encompasses the following : 1) Data collection, 2) Pre-processing, 3) Data balancing 4) Modeling of different classifying models 5) deployment.

1. **Data collection** :
The data is collected from some kaggle datasets which consist of twitter tweets, Go_emotions dataset (Hugging_face dataset having reddit comments) [15].

The datasets have variegated lengths and classes. We have stored them in separated csv and text files on github and later we merged them into a single dataframe of 328694 data points and 37 emotion categories. Table 1 provides a comprehensive overview of the obtained dataset.

Table 1 : Dataset Detail

Number of Reviews	Total Emotion Categories	Number of Instances per Emotion Category
328694	37 (admiration, amusement, anger, annoyance, approval, boredom, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, empty, enthusiasm, excitement, fear, fun, gratitude, grief, happiness, happy, hate, joy, love, nervousness, neutral, optimism, pride, realization, relief,	admiration(17135), amusement(8866), anger(18070), annoyance(11928), approval(15529), boredom(179), caring(5150), confusion(6601), curiosity(7708), desire(3001), disappointment(6768), disapproval(8916), disgust(4282), embarrassment(1720), empty(827), enthusiasm(759), excitement(4376), fear(12951), fun(1776), gratitude(8436), grief(494), happiness(5209), happy(7029), hate(6358), joy(22929), love(12454), nervousness(947), neutral(65731), optimism(4994), pride(714), realization(5125),

	remorse, sadness, shame, surprise, worry)	relief(2340), remorse(1648), sadness(27814), shame(146), surprise(11325), worry(8459).

2. Preprocessing:

Dropped all the columns except the text and its corresponding emotion.

1) We allowed only Latin script alphabets, digits from the decimal numeral system, single whitespace and some commonly used symbols which are - ['!' , '?', '.' , '@' ,'#' , "'" , ',' , '"' , '/' , ':' , '(' , ')'] Everything else was removed in the first phase.
2) Lowercase all the text.
3) Then we removed emojis, URLs, mentions, leading and trailing spaces.
4) Further we removed outliers of text data having length more than 200 words.
5) Then we removed stopwords.
6) The words are turned into tokens by tokenizer.
7) We applied stemming and lemmatization

2.1 Data Balancing :

In our work, a 2,50,000 dataset (after sampling) is used with 10 emotions that are derived from 32 emotions having 2,98,151 data points before sampling and preprocessing. Firstly the insignificant emotions that are not so broadly categorized are removed like ‘realization’, ‘approval’, ‘disapproval’, ‘shame’. Then outliers are removed which are text data having length more than 200 words. Lastly the sampling is done to balance the dataset which includes undersampling and oversampling.

Emotion before sampling -

Emotions	**count**
admiration	17135
amusement	8866
anger	18070

Emotions	**count**
annoyance	11928
boredom	179
caring	5150
confusion	6601
curiosity	7708
desire	3001
disappointment	6768
disgust	4282
embarrassment	1720
enthusiasm	759
excitement	4376
fear	12951
fun	1776
gratitude	8436
grief	494
happiness	5209
happy	7029
hate	6358
joy	22929
love	12454
nervousness	947
neutral	65731
optimism	4994
pride	714

Emotions	count
relief	2340
remorse	1648
sadness	27814
surprise	11325
worry	8459

Emotion after derivation -

Emotion	Count
anger	29998
excitement	17002
fear	22357
gratitude	33619
happy	36943
hate	10640
love	17604
neutral	65731
surprise-curiosity	25634
sadness	38623

Emotion after sampling -

Emotion	Count
neutral	25000
happy	25000
sadness	25000
gratitude	25000
fear	25000
love	25000
excitement	25000
hate	25000
anger	25000
surprise-curiosity	25000

3. Modeling of different classifying models :

In the pursuit of accurately predicting sentiment within text data, We employed a diverse array of classification models. The choice of models spans traditional machine learning approaches and advanced deep learning techniques, enabling a comprehensive comparison of their performance in sentiment analysis.

Initially, logistic regression was implemented as a baseline model due to its simplicity and efficiency in text classification tasks. Logistic regression, a linear model, is effective for binary classification and provides probabilistic outputs, making it suitable for sentiment analysis. It offers a straightforward approach to modeling the relationship between input features and the sentiment labels.

Naive Bayes was also used as another baseline model, leveraging the probabilistic relationships between features. This model is particularly well-suited for text data, where the assumption of feature independence often holds reasonably well. Naive Bayes classifiers are known for their effectiveness in handling large vocabularies and their efficiency in training and prediction.

To explore non-linear decision boundaries, a Decision Tree classifier was incorporated. This model, while intuitive and easy to interpret, can capture complex patterns in the data. However, its tendency to overfit necessitates careful tuning and evaluation. Despite its simplicity, the decision tree provided valuable insights into the feature importance and structure of the data.

Support Vector Classifier (SVC) was then applied to leverage its ability to handle high-dimensional spaces and construct hyperplanes that maximize the margin between classes. SVC's effectiveness in text classification stems from its robustness to overfitting and ability to handle the sparse nature of text data. The use of kernel functions in SVC allowed for capturing non-linear relationships, enhancing the model's flexibility and performance.

Transitioning to deep learning, Convolutional Neural Networks (CNN) combined with Long Short-Term Memory (LSTM) networks were utilized to capture both spatial and sequential dependencies in the text. The CNN layers excel at extracting local features, while the LSTM layers are adept at capturing long-term dependencies, making this hybrid model highly effective for sentiment analysis. The combination of CNN and LSTM leveraged the strengths of both architectures, improving the model's ability to understand complex textual patterns.

Then we explored using a standalone Recurrent Neural Network (RNN). RNNs, designed to handle sequential data, are capable of maintaining a form of memory over previous inputs, thus making them suitable for tasks like sentiment analysis where the order of words is crucial. The RNN's ability to capture temporal dependencies in the text was instrumental in modeling the flow and structure of the language, albeit with challenges related to long-term dependencies and gradient vanishing.

After observing the capabilities and limitations of the RNN, we decided to enhance the temporal dynamics further by incorporating a variant combining CNN with bidirectional-LSTM. This model processes the text in both forward and backward directions, thus capturing a more comprehensive context. Such bidirectional processing is particularly beneficial in sentiment analysis, where context from both preceding and succeeding words can significantly impact the sentiment classification. The bidirectional-LSTM added a layer of depth to the understanding of text sequences, contributing to improved sentiment prediction accuracy.

Each model was evaluated based on its accuracy, precision, and recall to determine its effectiveness in sentiment classification. The comparison provided insights into the trade-offs between model complexity, interpretability, and performance, guiding the selection of the most appropriate model for sentiment analysis in various applications.

4. Deployment :

To bring the sentiment analysis model to practical use, we deployed the pickle file of our best-performing model, the bi-directional LSTM with CNN, to a Streamlit server. This deployment enables users to interact with the model through a user-friendly web interface, making the advanced sentiment analysis capabilities accessible and easy to use.

The Streamlit application includes a functionality that allows users to test real-time sentiment on any text entered into the interface. The model not only predicts the overall sentiment but also provides per-class confidence scores, offering detailed insights into the probabilities of different sentiment classes. This feature enhances user understanding by breaking down the sentiment prediction into its constituent probabilities, making the results more transparent and interpretable.

In addition to real-time sentiment testing, we added a "search by emotion" functionality. This feature allows users to search through the dataset used for training the model by specifying an emotion. Users can explore examples from the dataset that correspond to each sentiment class, providing a deeper understanding of the data and the model's training context. This functionality is particularly useful for educational purposes and for users who wish to see concrete examples of how the model interprets different emotions.

Overall, the deployment on the Streamlit server transforms the sentiment analysis model into a practical tool, enabling real-time interaction and providing valuable insights through advanced functionalities.

EXPERIMENTAL RESULT ANALYSIS AND OUTCOMES OF PROJECT

- **Introduction to Experiments**

  The primary goal of our experiments was to evaluate and compare the effectiveness of various ML and DL techniques for emotion recognition in textual data. We aimed to test the hypothesis that advanced DL models, such as CNNs and LSTMs, would outperform traditional ML models in terms of accuracy and overall performance.

- **Experimental Setup:**

  Models and Techniques: The models evaluated included Logistic Regression, Naive Bayes Classifiers, Decision Trees,SVC, Recurrent Neural Networks (RNNs), and Long Short Term Memory Networks (LSTMs).

1. **Naive Bayes** : A simple algorithm that is often used for text classification tasks. Naive Bayes works by calculating the probability of each word in a text belonging to a particular emotion category and then selecting the category with the highest probability.

   Below is the code used and the scores that we got :

```
pipe_nb=Pipeline(steps[('cv',CountVectorizer()),('nb',MultinomialNB(
))])

pipe_nb.fit(x_train,y_train)

print("Multinomial Naive Bayes -")

print(f"Accuracy:{pipe_nb.score(x_test,y_test)}")
```

Output :

Multinomial Naive Bayes -
Accuracy: 0.4608825953397767 (46% approx.)

2. **Logistic Regression** : Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Below is the code used and the scores that we got :

```
from sklearn.pipeline import Pipeline

pipe_lr=Pipeline(steps=[('cv',CountVectorizer()),('lr',LogisticRegression(
)))

pipe_lr.fit(x_train,y_train)

print("Logistic Regression -")

print(f"Accuracy : {pipe_lr.score(x_test,y_test)}")
```

Output :

Logistic Regression -
Accuracy: 0.5041732224469918 (50.4% approx.)

3. **Decision tree classifier** : Decision tree classifiers use pre labeled data in order to train an algorithm that can be used to make a prediction. Decision trees can also be used for regression problems.

```
from sklearn.tree import DecisionTreeClassifier

pipe_tree                                                              =
Pipeline(steps=[('cv',CountVectorizer()),('dtree',DecisionTreeClassi
fier())])

pipe_tree.fit(x_train,y_train)

print("Decision Tree -")

print(f"Test score : {pipe_tree.score(x_test,y_test)")
```

Output:

```
[85] print("Decision Tree -")
     print(f"Test score : {pipe_tree.score(x_test,y_test)}\nTrain score : {pipe_tree.score(x_train,y_train)}")

Decision Tree -
Test score : 0.6295905943922516
Train score : 0.93463548359462
```

```
[86] y_pred_dtree = pipe_tree.predict(x_test)
```

```
[87] print('recall score = ',recall_score(y_test,y_pred_dtree,average='macro'))

recall score =  0.5723137077639558
```

```
[88] print('accuracy = ',accuracy_score(y_test,y_pred_dtree))
     print('precision = ',precision_score(y_test,y_pred_dtree,average = 'macro'))

accuracy =  0.6295905943922516
precision =  0.6022583280242809
```

```
from sklearn.metrics import f1_score
print("F1-Score = ",f1_score(y_test,y_pred,average='macro'))

F1-Score =  0.6102633050907867
```

4: **SVC** : A supervised learning algorithm called Support Vector Classification (SVC) was developed from Support Vector Machines (SVM). It finds the hyperplane that best divides the data points of several classes in the feature space and is applied to classification jobs.

```
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import recall_score, accuracy_score, precision_score, f1_score

# Assuming you have x_train, y_train, and x_test data ready

pipe_svm = Pipeline(steps=[('cv', CountVectorizer()), ('svm', SVC())])
pipe_svm.fit(x_train, y_train)

print("Support Vector Machine -")
print(f"Test score : {pipe_svm.score(x_test, y_test)}\nTrain score : {pipe_svm.score(x_train, y_train)}")
y_pred_svm = pipe_svm.predict(x_test)
print('recall score = ', recall_score(y_test, y_pred_svm, average='macro'))
print('accuracy = ', accuracy_score(y_test, y_pred_svm))
print('precision = ', precision_score(y_test, y_pred_svm, average='macro'))
print("F1-Score = ", f1_score(y_test, y_pred_svm, average='macro'))

✓ 178m 22.4s

Support Vector Machine -
Test score : 0.6658792317057658
Train score : 0.8037039633030381
recall score =  0.5447073652023533
accuracy =  0.6658792317057658
precision =  0.7180880180612016
F1-Score =  0.5999745954735262
```

5. **CNN With LSTM:** Combining Convolutional Neural Networks (CNNs) with Long Short Term Memory (LSTM) networks leverages CNNs for spatial feature extraction and LSTMs for capturing temporal dependencies. This hybrid architecture is particularly effective for tasks like text analysis and video classification, where both spatial and sequential information are crucial.

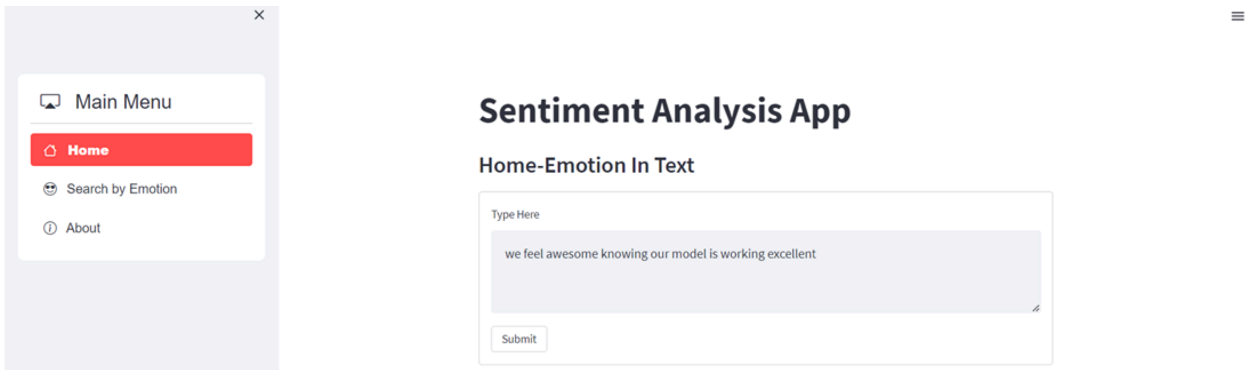
Below is the code used and the score

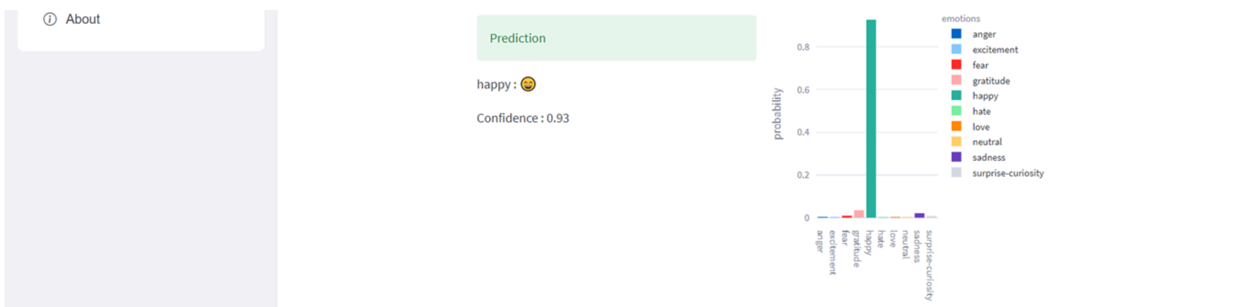
6. **CNN With Bi LSTM:** For tasks requiring both spatial and temporal input, Convolutional Neural Networks (CNNs) and Bidirectional Long Short Term Memory (BiLSTM) networks combine to form a potent design. This hybrid model is very useful for complicated sequence-based tasks that we got since it combines the advantages of both CNNs and BiLSTMs.

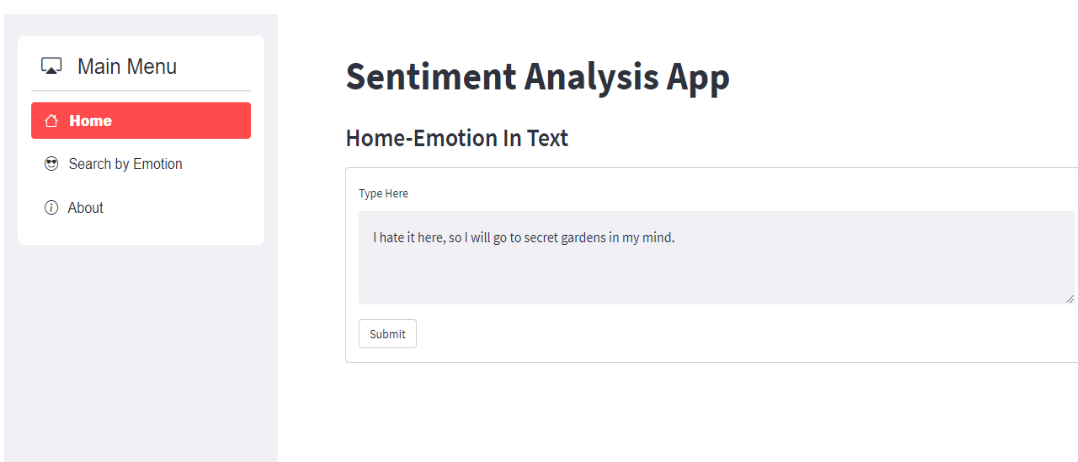
Model	Training Accuracy	Validation Accuracy	Precision	Recall	F1-Score
Logistic Regression	**75%**	**66%**	**66%**	**57%**	**61%**
Naive Bayes	**66%**	**46%**	**65%**	**41%**	**50%**
Decision Tree	**93%**	**62%**	**60%**	**57%**	**61%**
SVC	**80%**	**66%**	**71%**	**54.4%**	**59.9%**
CNN With LSTM	**83%**	**62%**	**61%**	**62%**	**61%**
CNN With Bi-LSTM	**94%**	**73%**	**73%**	**73%**	**73%**

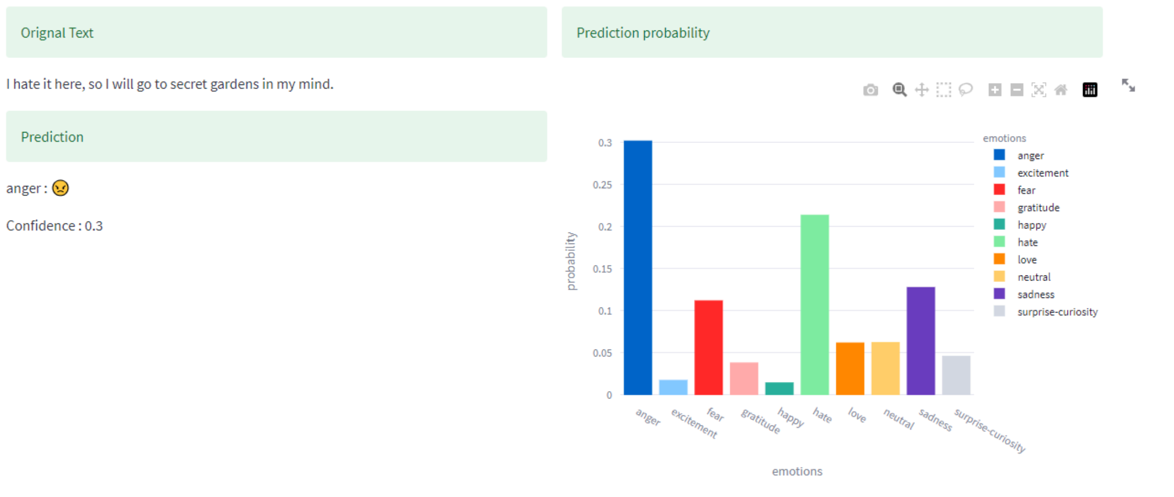
- **Tools and Frameworks:** All models were implemented using Python. For ML models, we used Scikit-learn, and for DL models, we utilized TensorFlow.
- **Interface:**
  We chose to use Streamlit for creating a machine learning app. Streamlit is an excellent tool for rapid prototyping because it allows developers to quickly create interactive demos and test various machine learning models without spending significant amounts of time on code. Streamlit's fast feedback loop enables developers to iterate quickly, making

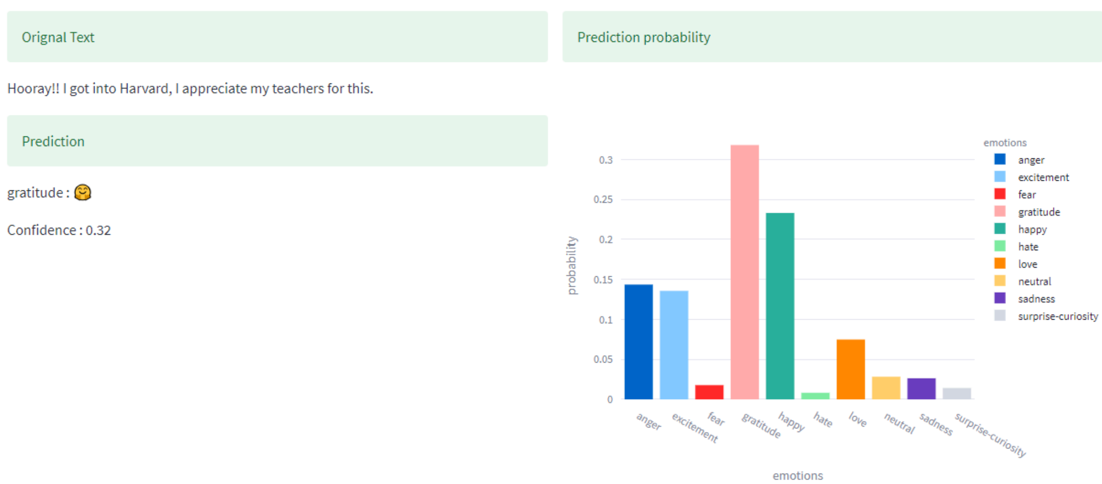
it an ideal choice for exploring different machine learning approaches and building proof-of-concept applications.

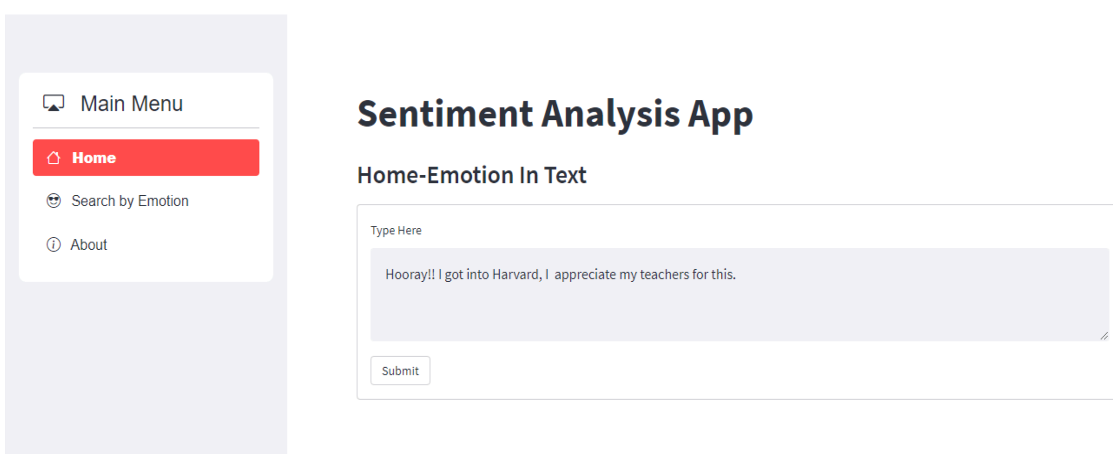







Orignal Text
Prediction probability
I hate it here, so I will go to secret gardens in my mind.
Prediction
anger :
Confidence : 0.3
emotions
anger
excitement
fear
gratitude
happy
hate
love
neutral
sadness
surprise-curiosity
probability
0.3
0.25
0.2
0.15
0.1
0.05
0
emotions


Orignal Text
Prediction probability
Hooray!! I got into Harvard, I appreciate my teachers for this.
Prediction
gratitude :
Confidence : 0.32
emotions
anger
excitement
fear
gratitude
happy
hate
love
neutral
sadness
surprise-curiosity
probability
0.3
0.25
0.2
0.15
0.1
0.05
0
emotions


Main Menu
Home
Search by Emotion
About
Sentiment Analysis App
Home-Emotion In Text
Type Here
Hooray!! I got into Harvard, I appreciate my teachers for this.
Submit

- ## Analysis of Result
    - **Performance Comparison:**

    The DL models (RNN and LSTM) demonstrated superior performance, likely due to their ability to capture temporal dependencies and contextual nuances in the text. Traditional ML models struggled with more complex sentence structures.

    - **Strengths and Weaknesses:**

    While DL models showed higher accuracy, they required significantly more computational resources and longer training times. In contrast, ML models were faster and more interpretable but less accurate.

    - **Error Analysis:**

      A common error across models was the misclassification of sarcastic comments. For example, the sentence "I just love waiting in traffic" was often misclassified, highlighting the need for a more nuanced understanding of context.

Discussion of Outcomes

- **Interpretation of Results:**

  The results validate our hypothesis that DL models, particularly LSTMs, are more effective for emotion recognition in text due to their advanced processing capabilities.

- **Practical Implications:**

  These findings suggest that for applications requiring high accuracy in emotion detection, such as customer service automation, DL models are preferable. However, for quick, interpretable results, traditional ML models still hold value.

- **Limitations:**

  This study was limited by the size and diversity of the dataset. Future work should explore larger datasets and more varied text sources to validate these findings further.

CONCLUSION AND FUTURE WORK

Future work:

1. **Brand monitoring**: Companies can use Twitter sentiment analysis to monitor the sentiment of their brand in real-time. This can help them identify any negative sentiment and take corrective actions if necessary.

2. **Crisis management**: During a crisis, Twitter sentiment analysis can be used to monitor the sentiment of the public towards the crisis. This can help government agencies and organizations take quick action to mitigate the impact of the crisis.

3. **Political analysis**: Sentiment analysis can be used to analyze public opinion on political issues. This can be useful for political parties and candidates to understand the sentiment of the public and tailor their messages accordingly.

4. **Product feedback**: Companies can use Twitter sentiment analysis to monitor feedback on their products. This can help them identify areas for improvement and make changes to their products accordingly.

5. **Event analysis**: Twitter sentiment analysis can be used to analyze the sentiment of the public towards events such as concerts, sports events, and conferences. This can help organizers make changes to improve the experience for attendees.

Conclusion:

Twitter sentiment analysis is a powerful tool for extracting valuable insights from social media data. By analyzing the sentiment of tweets, we can gain a better understanding of public opinion on various topics, such as products, brands, events, or political issues. This project aimed to showcase the effectiveness of sentiment analysis by analyzing a large dataset of tweets related to a specific topic or event. Through the use of machine learning algorithms and data analyzing techniques, we were able to accurately classify tweets into its sentiments. The results of this analysis can be used by businesses, researchers, or policymakers to make informed decisions based on public sentiment. Overall, the project demonstrated the potential of sentiment analysis as a valuable tool for social media analytics and data-driven decision making.

REFERENCES

1. Shiv Naresh Shivhare and Prof. Saritha Khethawat (2012), Emotion detection from text. In the proceedings of the Data Mining and Knowledge Management Process (DKMP 2012) conference, New Delhi. Doi: 10.5121/csit.2012.2237
2. Chun-Chieh Liu, Ting-Hao Yang, Chang-Tai Hsieh, Von-Wun Soo, ”Towards Text-based Emotion Detection: A Survey and Possible Improvements ”, in International Conference on Information Management and Engineering,2009.
3. http://www.wikipedia.org

4. Khurana, D., Koli, A., Khatter, K. et al. Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl 82, 3713–3744 (2023). https://doi.org/10.1007/s11042-022-13428-4
5. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y, (2014) On the properties of neural machine translation: encoder-decoder approaches. arXiv preprint arXiv:1409.1259
6. Dai Z, Yang Z, Yang Y, Carbonell J, Le QV, Salakhutdinov R, (2019) Transformer-xl: attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860
7. Palmer M, Gildea D, Kingsbury P (2005) The proposition bank: an annotated corpus of semantic roles. Computational linguistics 31(1):71–106.
8. Sharma S, Srinivas PYKL, Balabantaray RC (2016) Emotion Detection using Online Machine Learning Method and TLBO on Mixed Script. In Proceedings of Language Resources and Evaluation Conference 2016 (pp. 47–51).
9. Yi J, Nasukawa T, Bunescu R, Niblack W (2003) Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In data mining, 2003. ICDM 2003. Third IEEE international conference on (pp. 427-434). IEEE.
10. Muhammad Zubair Asghar, Fazli Subhan, Muhammad Imran1, Fazal Masud Kundi, Shahboddin Shamshirband, Amir Mosavi, Peter Csiba, Annamaria R. Varkonyi Koczy (2020) Performance Evaluation of Supervised Machine Learning Techniques for Efficient Detection of Emotions from Online Content. Doi: https://doi.org/10.32604/cmc.2020.07709 .
11. Cambria, E.; Grassi, M.; Hussain, A.; Havasi, C. (2012). Sentic computing for social media marketing. Multimedia tools and applications, vol. 59, no. 2, pp. 557-577.
12. Asghar, M. Z.; Khan, A.; Bibi, A.; Kundi, F. M.; Ahmad, H. (2017): Sentence-Level Emotion Detection Framework Using Rule-Based Classification. Cognitive Computation, vol. 9, no. 6, pp. 868-894.
13. Babajide, M.; I.; Saeed, F. (2016). Bioactive molecule prediction using extreme gradient boosting. Molecules, vol. 21, no. 8, pp. 983.
14. Cambria, E. (2016). Effective computing and sentiment analysis. IEEE Intelligent Systems, vol. 31, no. 2, pp. 102-107.
15. https://huggingface.co/datasets/go_emotions