Exploring Contextual Embeddings for Twitter Sentiment Analysis: A Comparative Study of NLP (Natural Language Processing)

PROJECT REPORT
OF MINI PROJECT

BACHELOR OF TECHNOLOGY
Data Science Dept **(VIth Semester)**

SUBMITTED BY
Ishu (2101331540047)
Mohammad Arman (2101331540063)
Mohd Shoyeb Sheakh (2101331540065)

NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY,
GREATER NOIDA, UTTAR PRADESH

STUDENT'S DECLARATION

We hereby certify that the work which is being presented in the Mini project report entitled" "Exploring Contextual Embeddings for Twitter Sentiment Analysis: A Comparative Study of NLP (Natural Language Processing)" in fulfillment of the requirement for the award of the Degree of Bachelor of Technology in Department of Data Science of Noida Institute of Engineering and Technology is an authentic record of our own work carried out during 6th semester. Each member of our group (Ishu , Mohammad Arman , Mohd Shoyeb Sheakh) has actively participated in the planning , execution and documentation phase of the project.

Date: 18th April 2024

Name of the Students

Ishu

Mohammad Arman

Mohd Shoyeb Sheakh

The Mini project viva-voce examination of Ishu (2101331540047), Mohammad Arman (2101331540063), Mohd Shoyeb Sheakh (2101331540065) of B.TECH Data Science has been held on 18th April 2024

Signature of:

Project Guide: _____

Head of Dept:
(Stamp of organization)

External Examiner: _____

Internal Examiner _____

ACKNOWLEDGEMENT

We are highly grateful to the Dr. Vinod M Kapse Director, **Noida Institute of Engineering and Technology**, Graeter Noida, for providing this opportunity.

The constant guidance and encouragement received from Dr. Manali Gupta, HOD (Data Science dept.), NIET, Greater Noida has been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to Dr. Raju and Dr. Shakeel Ahmad for their project guide, without the wise counsel and able guidance, it would have been impossible to complete the report in this manner.

We express gratitude to other faculty members of Data Science Department of NIET for their intellectual support throughout the course of this work.

Finally, the authors are indebted to all whosoever have contributed in this report work.

Ishu, Arman and Shoyeb

ABSTRACT

This research investigates the effectiveness of contextual embeddings for sentiment analysis specifically tailored for Twitter data, presenting a comprehensive comparative study within the domain of Natural Language Processing (NLP). The study encompasses a diverse range of contextual embedding models to capture the intricate nuances and sentiments prevalent in the informal and dynamic nature of Twitter communication. Through rigorous experimentation and evaluation, we meticulously assess the performance metrics such as accuracy, robustness, and computational efficiency of these models in sentiment analysis tasks.

The comparative analysis provides invaluable insights into the strengths and limitations of different contextual embeddings, shedding light on their applicability and effectiveness in understanding and analyzing sentiments expressed in social media discourse. Our findings contribute significantly to advancing the field of NLP by offering a detailed evaluation framework for contextual embeddings, thus facilitating informed decisions in choosing appropriate techniques for sentiment analysis in social media data. This research bridges the gap between theoretical advancements and practical applicability, providing a solid foundation for enhancing sentiment analysis methodologies tailored for the complex and expressive context of Twitter.

INDEX

INTRODUCTION

- Project Scope: The project delves into the realm of sentiment analysis on Twitter, a microblogging platform renowned for its dynamic and expressive nature. We aim to leverage advanced Natural Language Processing (NLP) techniques to analyze sentiments expressed in tweets.
- Focus on Contextual Embeddings: Our focus is on exploring contextual embeddings, a cutting-edge approach in NLP, to capture the nuanced meanings and sentiments embedded within the context of Twitter conversations. Contextual embeddings have shown promising results in understanding language semantics and context, making them ideal for sentiment analysis tasks.
- Comparative Study Framework: The project adopts a comparative study framework to evaluate the effectiveness of various contextual embedding models. By comparing these models against traditional sentiment analysis methods, we seek to identify the strengths and limitations of each approach in handling Twitter data's unique characteristics.
- Contributions and Impact: Through this research endeavor, we aim to contribute to the advancement of sentiment analysis methodologies tailored for social media data. Our findings will provide valuable insights for researchers and practitioners in choosing appropriate NLP techniques for sentiment analysis tasks in the context of Twitter and other social media platforms.

PROBLEM DEFINATION

- Twitter Sentiment Analysis Challenges: The project addresses the challenges posed by sentiment analysis on Twitter, where the brevity, informality, and diverse language expressions present unique hurdles for traditional sentiment analysis techniques.
- Need for Contextual Understanding: One of the key issues is the need for a deeper contextual understanding of tweets to accurately capture the intended sentiment. Traditional methods often struggle to grasp the nuances and subtleties present in informal social media language.
- Effectiveness of Contextual Embeddings: The project seeks to investigate whether leveraging contextual embeddings can significantly enhance sentiment analysis accuracy in the context of Twitter. This involves evaluating the performance of contextual embedding models against traditional sentiment analysis approaches.
- Comparative Evaluation Framework: To address these challenges, the project establishes a comparative evaluation framework that systematically compares the performance of different NLP techniques, including contextual embeddings, in sentiment analysis tasks. This framework aims to provide insights into the suitability and effectiveness of various techniques for sentiment analysis on Twitter data.

PURPOSE OF THE PROJECT

- Advancing Sentiment Analysis Techniques: The primary purpose of this project is to advance the state-of-the-art in sentiment analysis methodologies, particularly in the context of social media platforms like Twitter. By exploring and evaluating cutting-edge NLP techniques such as contextual embeddings, we aim to enhance the accuracy and reliability of sentiment analysis results.
- Addressing Real-World Challenges: The project is driven by the need to address real-world challenges faced in understanding and analyzing sentiments expressed in social media data. By focusing on Twitter sentiment analysis, we aim to contribute valuable insights and solutions to the broader field of social media analytics and opinion mining.
- Informing Decision-Making Processes: Another key purpose is to provide valuable insights that can inform decision-making processes for businesses, organizations, and researchers. Accurate sentiment analysis on social media data can help in understanding public opinions, customer feedback, and market trends, leading to informed strategies and actions.

TOOLS AND TECHNOLGY USED

- Python: A versatile and widely-used programming language known for its simplicity and extensive libraries, ideal for data analysis and machine learning tasks.
- Naive Bayes: A simple yet effective probabilistic classifier based on Bayes' theorem, commonly used for text classification tasks like sentiment analysis.
- LSTM (Long Short-Term Memory): A type of recurrent neural network (RNN) architecture designed to capture long-term dependencies in sequential data, suitable for analyzing text sequences.
- Random Forest: A machine learning ensemble method that uses multiple decision trees to make predictions, known for its robustness and ability to handle complex datasets.
- RNN (Recurrent Neural Network): A class of neural networks designed for sequential data processing, often used in natural language processing (NLP) tasks due to their ability to capture temporal dependencies.
- Google Colab: A cloud-based platform provided by Google that allows users to write and execute Python code in a browser environment, with access to GPU and TPU resources for machine learning tasks.
- Jupyter Notebooks: An interactive computing environment that enables users to create and share documents containing live code, equations, visualizations, and explanatory text, widely used for data exploration and analysis in a collaborative manner.

Naive Bayes

- Probabilistic Classifier: Naive Bayes is a probabilistic classifier based on Bayes' theorem, which calculates the probability of a label given some observed features.
- Assumption of Independence: It assumes that the features are independent of each other, given the class label, even though this assumption is rarely met in real-world data.
- Simple and Efficient: Naive Bayes is simple to implement and computationally efficient, making it suitable for large datasets and real-time applications.
- Applicability to Text Classification: It is commonly used in text classification tasks, such as sentiment analysis and spam detection, where each word or feature contributes to the overall classification.
- Handling Categorical Data: Naive Bayes works well with categorical features and is robust to noise in the data.
- Naive Bayes Variants: There are different variants of Naive Bayes, including Gaussian Naive Bayes (for continuous features), Multinomial Naive Bayes (for count-based features), and Bernoulli Naive Bayes (for binary features), each suited to different types of data distributions.

LSTM (Long Short-Term Memory)

- Sequential Data Handling: LSTM is a type of recurrent neural network (RNN) designed to handle and process sequential data, such as time series or natural language text.
- Long-Term Dependencies: Unlike traditional RNNs, LSTM is capable of capturing long-term dependencies in sequences, making it suitable for tasks where context from distant past steps is crucial.
- Memory Cells: It incorporates memory cells with self-gating mechanisms, such as input gates, forget gates, and output gates, allowing it to selectively remember or forget information over time.
- Prevention of Vanishing/Exploding Gradients: LSTM addresses the vanishing and exploding gradient problems often encountered in training deep neural networks by controlling the flow of gradients during backpropagation through time.
- Applications in NLP: LSTM is widely used in natural language processing (NLP) tasks, including machine translation, sentiment analysis, text generation, and speech recognition, due to its ability to capture semantic meaning and context in textual data.
- Variants and Extensions: Over time, various LSTM variants and extensions have been developed, such as Bidirectional LSTMs (Bi-LSTMs), which process sequences in both forward and backward directions, enhancing context understanding in tasks like sentiment analysis and named entity recognition.

SOFTWARE REQUIREMENT SPECIFICATIONS

❖ Introduction

Exploring Contextual Embeddings for Twitter Sentiment Analysis: Comparative Study of NLP techniques for accurate sentiment understanding in tweets.

❖ Functional Requirements

Data Collection: Gather a diverse dataset of Twitter posts with labeled sentiment annotations for training and testing purposes.

Preprocessing: Implement preprocessing steps such as tokenization, stemming, and stop-word removal to clean and prepare the textual data for analysis.

Embedding Models Integration: Integrate various contextual embedding models like Word2Vec, GloVe, and BERT for the semantic representation of tweets**..**

Model Training: Train machine learning and deep learning models including Naive Bayes, LSTM, and random forest on the preprocessed and embedded data**.**

Sentiment Analysis: Develop algorithms to perform sentiment analysis on tweets, categorizing them into positive, negative, or neutral sentiments.

Evaluation Metrics: Implement evaluation metrics such as accuracy, precision, recall, and F1-score to assess the performance of sentiment analysis models**.**

❖ Non-functional Requirements

Performance: The system should exhibit fast response times and efficient processing of sentiment analysis tasks, even with large volumes of Twitter data..

Accuracy: Ensure high accuracy in sentiment analysis predictions to provide reliable insights into the sentiments expressed in tweets.

Robustness: Handle noisy and diverse Twitter data effectively, maintaining consistent performance across different types of tweets and linguistic variations.

Reliability Ensure the system's reliability by minimizing downtime, errors, and disruptions, thus enabling continuous sentiment analysis operations.

❖ **System Architecture**

Utilizing a layered architecture with data collection, preprocessing, embedding integration, model training, and sentiment analysis modules for Twitter sentiment analysis.

❖ **Testing**

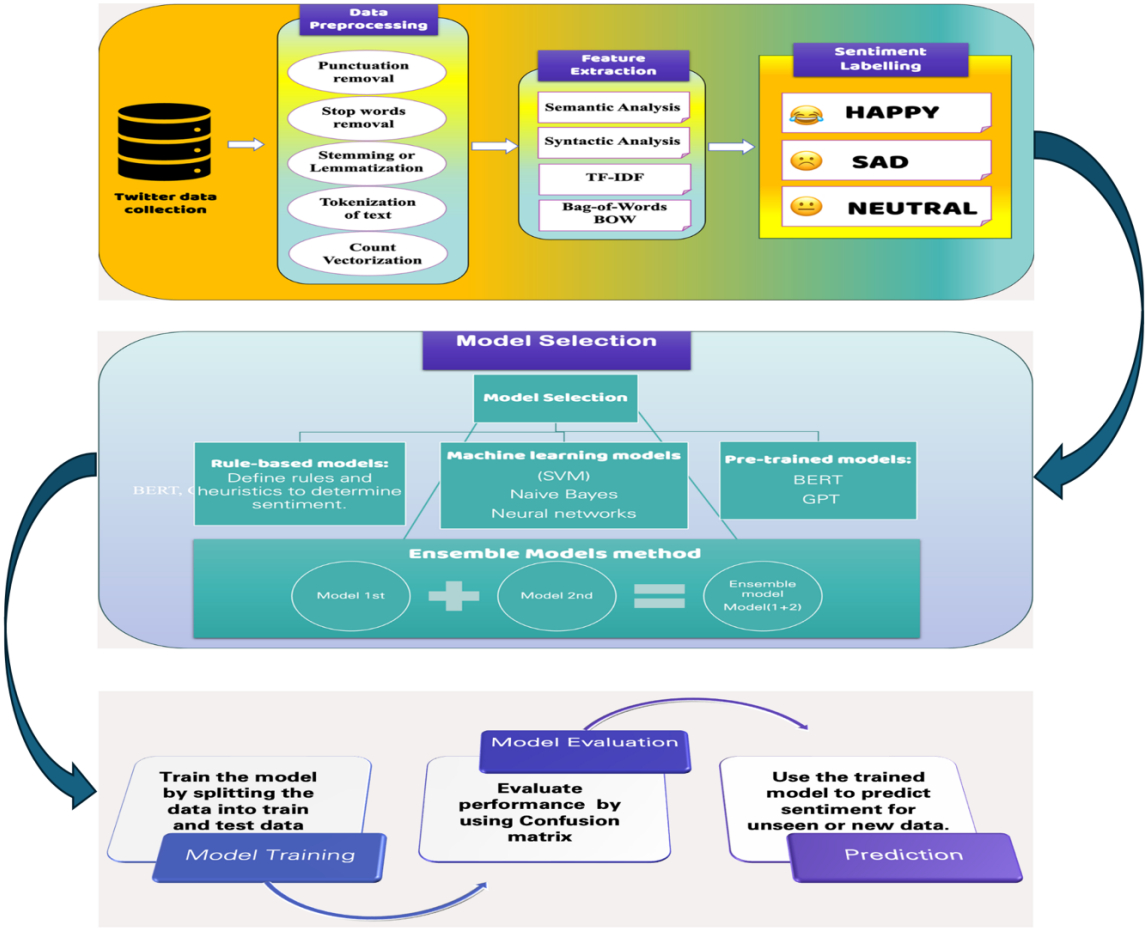
The system undergoes rigorous testing to ensure accuracy and reliability in sentiment analysis.

❖ **Maintenance and Support**

Maintenance and support services are provided to ensure the ongoing functionality, performance, and usability of the sentiment analysis system.

SYSTEM DESIGN

Flowchart



1. **Preprocessing**:
    - Preprocessing refers to the initial steps taken to prepare raw data for further analysis. It involves cleaning, organizing, and transforming data to make it suitable for machine learning or other analytical tasks.
    - **Techniques**:

        1. **Tokenization**: Breaking down text into smaller units, such as words or phrases.
        2. **Stemming and Lemmatization**: Reducing words to their root or base form to standardize variations.
        3. **Named Entity Recognition (NER)**: Identifying entities (e.g., names, locations, organizations) in the text.
        4. **Part-of-Speech Tagging**: Assigning grammatical categories to words (e.g., noun, verb, adjective).
        5. **Text Clustering and Classification**: Grouping similar documents or assigning predefined categories to texts.
        6. **Topic Modeling**: Identifying topics or themes within a collection of documents.

2. **Semantic Analysis**:
    - Semantic analysis is a process in machine learning that interprets the meaning of text. It involves understanding the logical meaning behind words, phrases, and sentences.
    - For example, sentiment analysis relies on semantic analysis to determine whether a piece of text expresses positive, negative, or neutral sentiment.
3. **Sentiment Labeling**:
    - Sentiment labeling involves assigning a sentiment value to a piece of text. Common sentiment labels include “happy,” “sad,” “neutral,” etc.
    - Sentiment analysis models use this labeling to classify text based on its emotional tone.
4. **Model Selection**:
    - Model selection is the process of choosing an appropriate algorithm or model for data analysis.
    - Examples of models include:
        - Rule-based Models: These models follow predefined rules to make predictions.
        - Machine Learning Models (e.g., Naive Bayes, SVM, Neural Networks): These learn from data to make predictions.
        - Pre-trained Models (e.g., BERT): These are already trained on large datasets and can be fine-tuned for specific tasks.
    - Ensemble Methods combine multiple models to improve prediction accuracy.
    - **Techniques:**

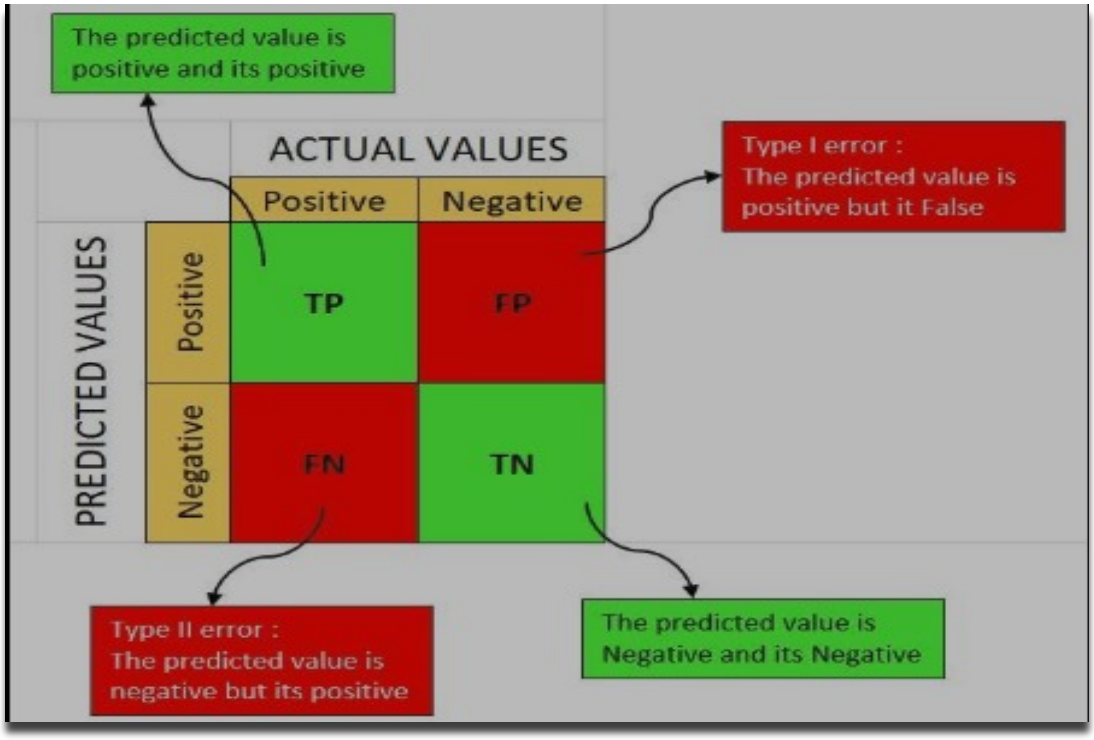
        1. **Rule-based Approaches**: Defining rules and heuristics to classify sentiment based on predefined criteria.
        2. **Machine Learning Approaches**: Training models, often using supervised learning, on labeled datasets to predict sentiment.
        3. **Lexicon-based Approaches:** Using sentiment lexicons or dictionaries to assign sentiment scores to words and aggregating them to determine overall sentiment.

4. **Deep Learning Approaches**: Utilizing deep neural networks, such as recurrent neural networks (RNNs) or transformers, for more sophisticated sentiment analysis.

5. **Model Evaluation**:
   - Model evaluation assesses the performance of a trained model. Common evaluation metrics include accuracy, precision, recall, and F1-score.
   - The confusion matrix is often used to visualize model performance.
   - Once a model is evaluated, it can be used for making predictions on new data.

1. Confusion Matrix:

- A confusion matrix is a table that summarizes the performance of a machine learning model on a set of test data.
- It helps us understand how well the model's predictions align with the actual outcomes.
- The matrix consists of four components:
  - **True Positives (TP)**: Instances correctly predicted as positive.
  - **True Negatives (TN)**: Instances correctly predicted as negative.
  - **False Positives (FP)**: Instances predicted as positive but actually negative (Type I error).
  - **False Negatives (FN)**: Instances predicted as negative but actually



2. Precision:

- Precision measures the proportion of true positive predictions among all positive predictions made by the model.
- It answers the question: "Of all instances predicted as positive, how many were actually positive?"
- Formula:
    - Precision = TP / (TP + FP)

3. Recall (Sensitivity):

- Recall (also known as sensitivity or true positive rate) measures the proportion of actual positive instances correctly predicted by the model.
- It answers the question: "Of all actual positive instances, how many did the model correctly predict?"
- Formula:
    - Recall = TP / (TP + FN)

4. F1-Score:

- The F1-score is the harmonic mean of precision and recall.
- It balances precision and recall, especially when there is an uneven class distribution.
- Formula:
    - F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

Example:

Let's consider a binary classification problem (e.g., identifying whether an email is spam or not):

- True Positives (TP): 100
- True Negatives (TN): 800
- False Positives (FP): 20
- False Negatives (FN): 30

Using the formulas:

- Precision = 100 / (100 + 20) = 0.83
- Recall = 100 / (100 + 30) = 0.77
- F1-Score = 2 * (0.83 * 0.77) / (0.83 + 0.77) = 0.80

Remember that F1-score combines both precision and recall, providing a more comprehensive evaluation of the model's performance.

SYSTEM IMPLEMENTATION

DATASET

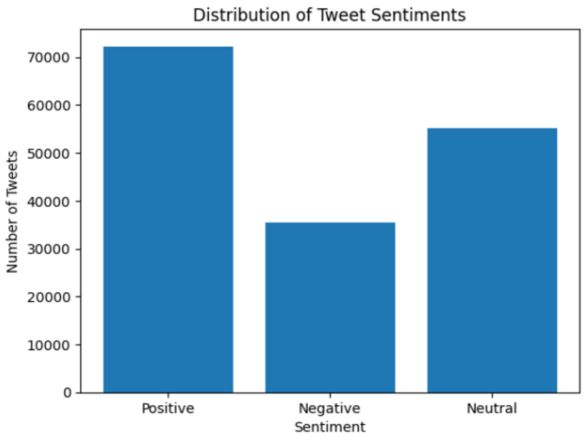
The dataset is basically downloaded from Kaggle which contains three different types labels category(Neutral, Negative, Positive) and people opinions as “clean_text” namely it contains 160k tweets extracted.

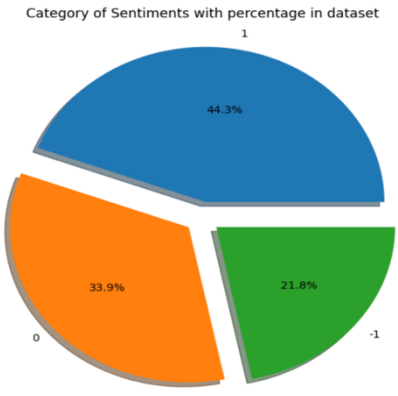
Dataset instance

clean_text	category
when modi promised “minimum government maximum governance” expected him begin the difficult job reforming the state why does take years get justice state should and not business and should exit psus and temples	-1
talk all the nonsense and continue all the drama will vote for modi	0
what did just say vote for modi welcome bjp told you rahul the main campaigner for modi think modi should just relax	1
asking his supporters prefix chowkidar their names modi did great service now there confusion what read what not now crustal clear what will crass filthy nonsensical see how most abuses are coming from chowkidars	1
answer who among these the most powerful world leader today trump putin modi may	1
kiya tho refresh maarkefir comment karo	0
surat women perform yagna seeks divine grace for narendra modi become again	0
this comes from cabinet which has scholars like modi smriti and hema time introspect	0
with upcoming election india saga going important pair look current modi leads govt elected with deal brexit combination this weekly looks juicy bears imho	1
gandhi was gay does modi	1
things like demonetisation gst goods and services tax…the upper castes would sort either view favourably say that need give this more time other castes like dalits the muslims were more against because that’ just not modi’ constituency2	1
hope tuthukudi people would prefer honest well behaved nationalist courageous likly minister modi cabinet vote benifit thuthukudi	1
calm waters wheres the modi wave	1
one vote can make all the difference anil kapoor answers modis election 2019 clarion call extends support his vote kar campaign	0
one vote can make all the difference anil kapoor answers modis election 2019 clarion call extends support his campaign	0
vote such party and leadershipwho can take fast and firm action none other than narendra damodardas modi and bjp party	-1
vote modi who has not created jobs	0
through our vote ensure govt need and deserve anupam kher responds modis appeal for the 2019 elections	0
dont play with the words was talking about the modi swamy relation guru saying what good and chowkidar protecting the good mind you tweeted dark side terrorism there any brighter side you better know there any	1
didn’ write chowkidar does mean ’ anti modi try visit the plz not all who haven’ used are anti	-1
was the one who recently said that people who vote against modi are anti national that put gen hooda all congress supporters and those jawans who not support modi anti national what great things did you hear about him	1
with firm belief the leadership shri narendra modi bjp entering into politics given form file nomination for the khammam parliamentary seat proceeding khammam today	-1

VISUALIZATION OF THE DATA:

S.No.	Sentiment Type	Label used	Total Value Counts
1.	Positive Tweets	1	72250
2.	Neutral Tweets	0	55213
3.	Negative Tweets	-1	35510





VECTORIZATION OF THE TEXTUAL DATA

- **TF-IDF:** TF-IDF, stands for Term Frequency-Inverse Document Frequency, it is a statistical method that is mostly used in NLP(natural language processing) for to retrieve the information. It measures that how important a particular word is in a document with respect to the collection of documents (corpus). Words that are basically present in the textual documents are converted to importance numbers with the help of text vectorization.
  In TF-IDF
  TF is the ratio of frequency of a particular term "**t**" present in the document "**d**" to the total number of words in document "**d**".

$$TF(t,d) = \frac{frequency\ (t,d)}{total\ word\ count(d)} \qquad (1)$$

  IDF determines the importance of the word or a term in all the documents present in the corpus, total number of documents containing that term in corpus. If a term is rare so it has high IDF score so it depicts that term is very important(*technical words, jargons*) but if a term has low IDF score it depicts that term is common in the corpus(e.g.: it, we, they, the, etc.).

$$IDF(t,d) = (\frac{total\ number\ of\ documents(d) in\ corpus}{total\ number\ of\ documents\ in\ corpus\ containing\ term(t,d)}) \qquad (2)$$

  TF-IDF is the term we get after multiplying the **TF** and **IDF** scores.

$$TF\text{-}IDF = tf_{(t,d)} * log(idf_{(t,d)}) \qquad (3)$$

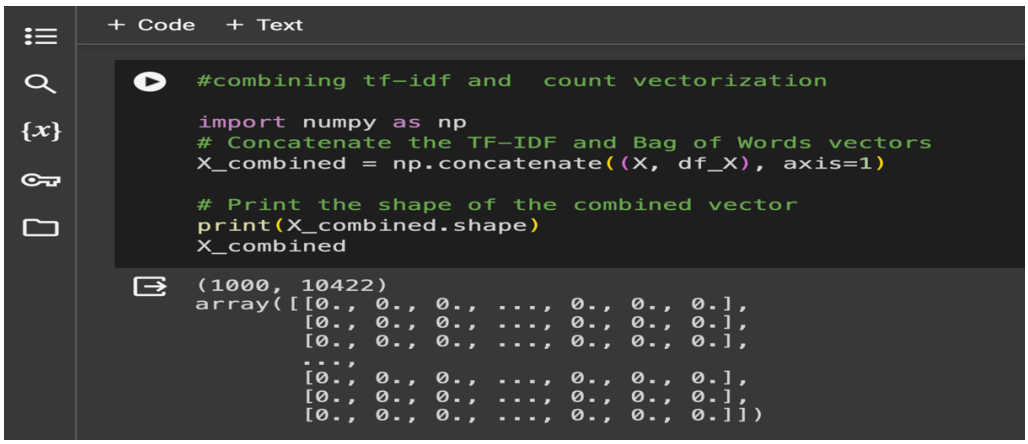
- **Bow:** Bow stands for Bag of Words, here the text of document is broken into the tokens by the process called tokenization and all the punctuations and stop words are excluded from the text, then in each document the value of each element represents the frequency of the word in that document. Once the vocabulary is built then.

BoW

1. **Hello and welcome friends**
2. **Arman, Ishu and Shoaib love teaching**
3. **Arman, Ishu and Shoaib love to play cricket**

D.No.	Hello	and	welcome	friends	Arman	Ishu	and	Shoaib	love	teaching	to	play	cricket
1.	1	1	1	1	0	0	0	0	0	0	0	0	0
2.	0	0	0	0	1	1	1	1	1	1	0	0	0
3.	0	0	0	0	1	1	1	1	1	0	1	1	1

We are using the Vectorization By Concatenating the Both Techniques:
TF-IDF + CountVectorization(BoW)



Dependant Variable "y":

Copy of MODI_Twitter_Sentiment_Analysis.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

(10000, 41318)

DEPENDANT VARIABLE

```
[ ] y = df['category']
    print("Y SHAPE : ",y.shape[0])
    print()
    y
```

```
Y SHAPE :  10000

102593    1
148753    1
2704      0
138001    0
73506     1
         ..
27748     1
107598   -1
151666    0
77114     1
62391     1
Name: category, Length: 10000, dtype: int64
```

Independant Variable "x":

```
x = df_combined
print("X SHAPE : ",x.shape)
x
```

```
X SHAPE :  (10000, 41318)
```

	0	1	2	3	4	5	6	7	8	9	...	41308	41309	41310	41311	41312	41313	41314	41315	41316	41317
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9999	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

10000 rows × 41318 columns

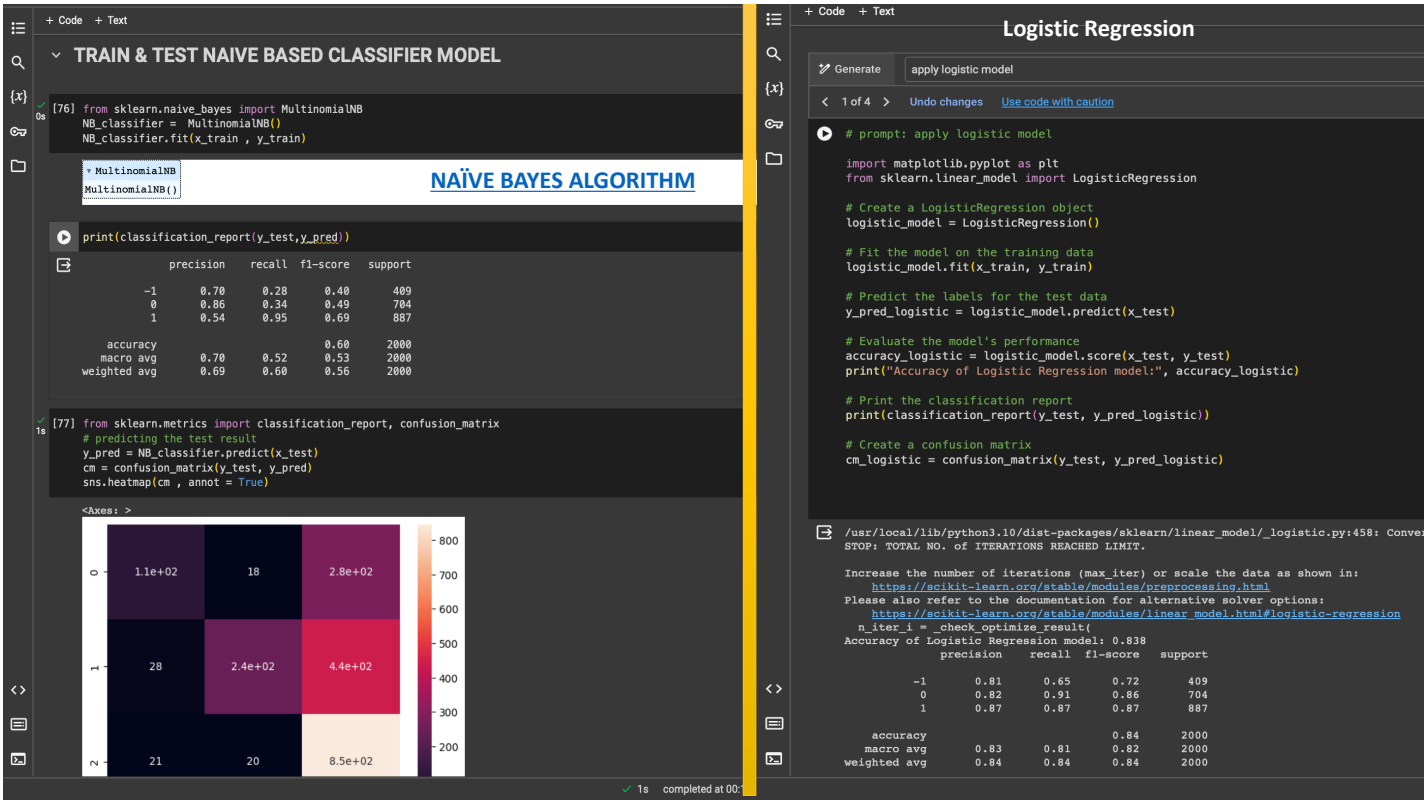
Dataset Splitting for Test/Train:

SPLITTING TRAIN TEST DATASETS

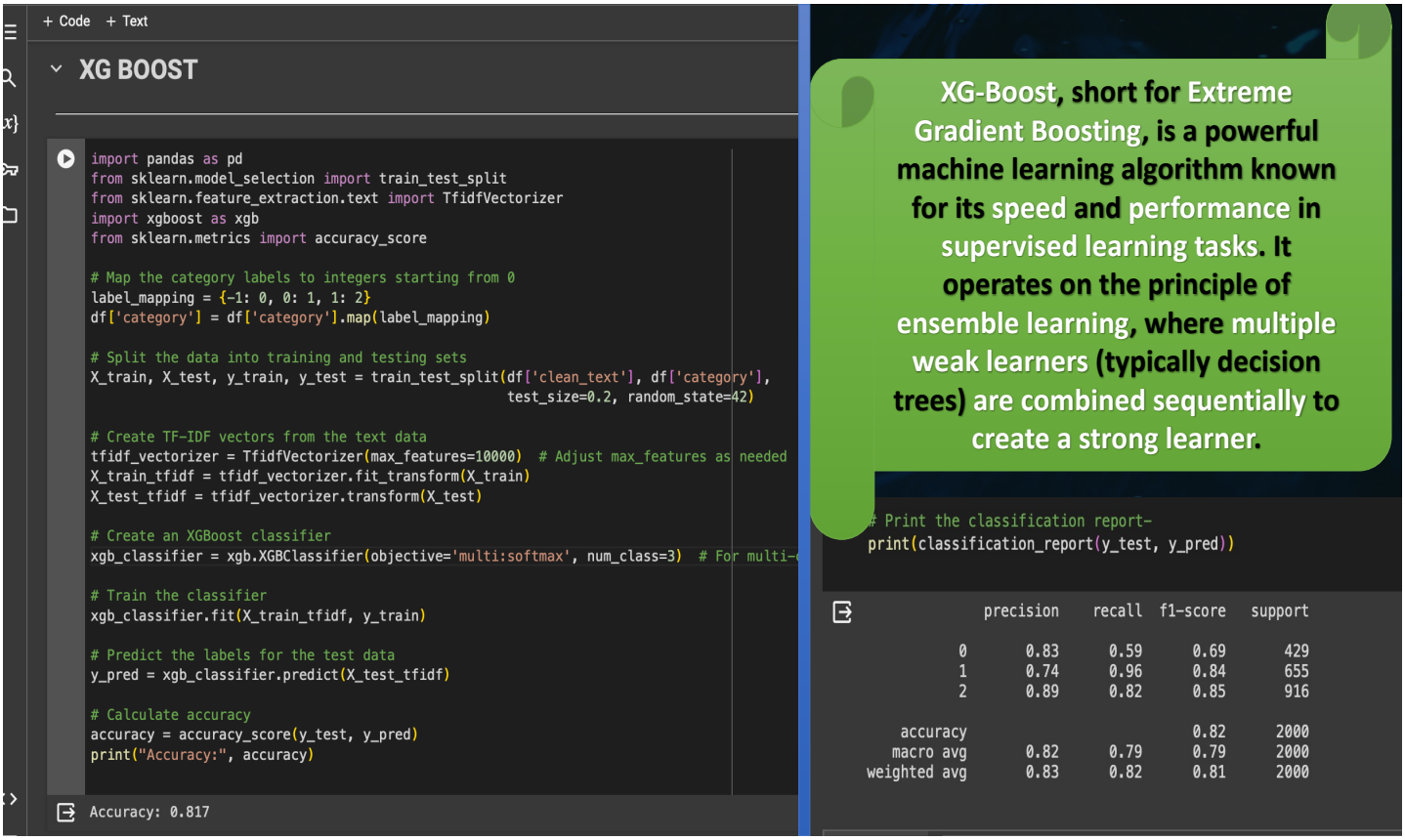
```
from sklearn.model_selection import train_test_split
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size = 0.2)
```

CODING AND SCREENSHOTS

Naïve Bayes Algorithm & Logistic Regression Model



XG BOOST Classifier



CATBOOST CLASSIFIER

```
[ ] !pip install catboost
```

```
from catboost import CatBoostClassifier
cat_model = CatBoostClassifier(verbose = 500)
# training
cat_model.fit(x_train, y_train)
```

```
Learning rate set to 0.087979
0:      learn: 1.0799479        total: 853ms    remaining: 14m 12s
500:    learn: 0.5442492        total: 2m 44s   remaining: 2m 43s
999:    learn: 0.4680361        total: 5m 20s   remaining: 0us
<catboost.core.CatBoostClassifier at 0x7ab9c1c7f310>
```

```
[92] print(classification_report(y_test, y_pred2))
```

	precision	recall	f1-score	support
-1	0.91	0.61	0.73	409
0	0.78	0.97	0.87	704
1	0.89	0.86	0.87	887
accuracy			0.85	2000
macro avg	0.86	0.81	0.82	2000
weighted avg	0.86	0.85	0.84	2000

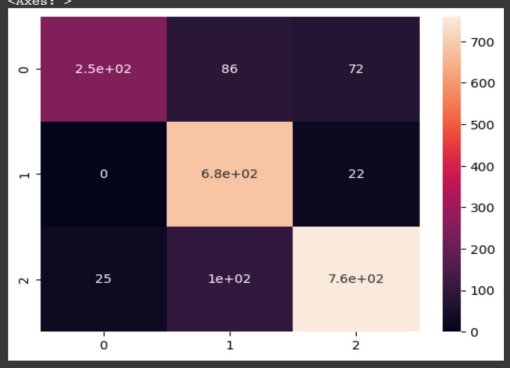
```
y_pred2 = cat_model.predict(x_test)
cm = confusion_matrix(y_test , y_pred2)
sns.heatmap(cm, annot = True)
```

```
<Axes: >
```

Cat-Boost is a gradient boosting algorithm designed for high-performance machine learning tasks. It excels in handling categorical features without the need for extensive preprocessing, while also providing efficient training speed and robustness to noisy data.

```
y_pred2 = cat_model.predict(x_test)
cm = confusion_matrix(y_test , y_pred2)
sns.heatmap(cm, annot = True)
```

```
<Axes: >
```




```
[93] df.info()
     df["category"].value_counts()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10000 entries, 102593 to 62391
```

Accuracy

S.No.	Model	Accuracy	Class	Precision	Recall	F1-Score
1.	Naïve Bayes Classifier	0.60	-1	0.70	0.28	0.40
			0	0.86	0.34	0.49
			1	0.54	0.95	0.69
			macro avg	0.70	0.52	0.53
			weighted avg	0.69	0.60	0.56
2.	Logistic Regression	0.84	-1	0.81	0.65	0.72
			0	0.82	0.91	0.86
			1	0.87	0.87	0.87
			macro avg	0.81	0.81	0.82
			weighted avg	0.84	0.84	0.84
3	Cat Boost Classifier	0.85	-1	0.91	0.61	0.73
			0	0.78	0.97	0.87
			1	0.89	0.86	0.87
			macro avg	0.86	0.81	0.82
			weighted avg	0.86	0.85	0.84
4.	XG Boost Model	0.82	-1	0.83	0.59	0.69
			0	0.74	0.96	0.84
			1	0.89	0.82	0.85
			macro avg	0.82	0.79	0.79
			weighted avg	0.83	0.82	0.81

We get 85% accuracy for our test dataset which is considered to be a good accuracy!

CONCLUSIONS

- Effective Sentiment Analysis: The project demonstrates the effectiveness of leveraging contextual embeddings and NLP techniques for accurate sentiment analysis on Twitter data.
- Insights and Recommendations: The findings provide valuable insights and recommendations for businesses and organizations to better understand and respond to sentiment trends on social media platforms.
- Future Research Directions: Future research could explore advanced model architectures, multi-lingual analysis, and real-time monitoring to further enhance sentiment analysis capabilities.

FUTURE SCOPE

- Advanced Model Architectures: Explore hybrid models and transformer-based architectures for improved sentiment analysis accuracy.
- Multi-Lingual Analysis: Adapt models for handling diverse languages and nuances on global social media platforms.
- Real-Time Monitoring: Develop capabilities for real-time sentiment analysis to respond promptly to evolving trends.
- Interpretability: Focus on enhancing the interpretability of deep learning models like LSTMs for transparent decision-making.
- Ethical Bias Mitigation: Address biases and ethical considerations in sentiment analysis algorithms for fair analysis.
- Integration with User Feedback Systems: Integrate sentiment analysis insights into user feedback systems to enhance product/service improvements and customer satisfaction monitoring, contributing to a more data-driven feedback loop**.**
- Sentiment Analysis in Multimedia Content: Explore sentiment analysis techniques for multimedia content such as images, videos, and audio clips shared on social media platforms to capture rich sentiment expressions.

REFERENCES

- Coletta, L. F., da Silva, N. F., Hruschka, E. R., & Hruschka, E. R. (2014, October). Combining classification and clustering for tweet sentiment analysis. In *2014 Brazilian conference on intelligent systems* (pp. 210-215). IEEE.

- [2] Hasan, M. R., Maliha, M., & Arifuzzaman, M. (2019, July). Sentiment analysis with NLP on Twitter data. In 2019 international conference on computer, communication, chemical, materials and electronic engineering (IC4ME2) (pp. 1-4). IEEE.

- [3] Pham, D. H., & Le, A. C. (2018). Learning multiple layers of knowledge representation for aspect based sentiment analysis. Data & Knowledge Engineering, 114, 26-39.

- [4] Sahayak, V., Shete, V., & Pathan, A. (2015). Sentiment analysis on twitter data. International Journal of Innovative Research in Advanced Engineering (IJIRAE), 2(1), 178-183.

- [5] Rajak, A., Panda, R. N., & Kumar, A. (2024, February). Combining Text Information and Sentiment Dictionary for Sentiment Analysis on Twitter During Covid. In 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT) (Vol. 5, pp. 298-302). IEEE.

- [6] Rashid, A., & Huang, C. Y. (2021). Sentiment Analysis on Consumer Reviews of Amazon Products. International Journal of Computer Theory and Engineering, 13(2), 7.

- [7] Haque, T. U., Saber, N. N., & Shah, F. M. (2018, May). Sentiment analysis on large scale Amazon product reviews. In 2018 IEEE international conference on innovative research and development (ICIRD) (pp. 1-6). IEEE.

- [8] Saif, H., He, Y., & Alani, H. (2012). Semantic sentiment analysis of Twitter. In The Semantic Web–ISWC 2012: 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I (pp. 508-524). Springer Berlin Heidelberg.

- [9] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. J. (2011, June). Sentiment analysis of twitter data. In Proceedings of the workshop on language in social media (LSM 2011) (pp. 30-38).

- [10] AlQahtani, A. S. (2021). Product sentiment analysis for amazon reviews. International Journal of Computer Science & Information Technology (IJCSIT) Vol, 13.