**CS6003**

**BIG DATA ANALYTICS**

**Topic: TOXIC COMMENTS CLASSIFICATION**

* ***DONE BY***

***R A NITHISH KUMAR - 2021103548  
SANGAMITHRA M – 2021103572***

***GUIDE – DR.AROCKIA XAVIER ANNIE***

**PROBLEM STATEMENT**

The project is focused on developing an automated system capable of accurately identifying and flagging toxic comments within online platforms, ultimately contributing to improved user experiences and safer online interactions.

**ABSTRACT**

* This project addresses the issue of toxic comments on online platforms using big data analytics techniques.
* Techniques such as word embedding models, recurrent neural networks, and convolutional neural networks are explored to capture semantic information.
* Additionally, the integration of user metadata enhances predictive performance.
* Leveraging big data platforms like Apache Spark and Hadoop enables efficient processing and analysis of massive comment data.
* The outcomes include improved online content moderation, fostering healthier discussions, and contributing to the advancement of big data analytics for addressing social and ethical challenges online.

**INTRODUCTION**

Our project addresses the concerning rise of toxic comments in online spaces, particularly on social media platforms where increased interactions have amplified the issue.

We aim to develop an *advanced comment toxicity detection system* to effectively tackle this problem head-on.

Leveraging cutting-edge *natural language processing (NLP) techniques like WordEmbedding and machine learning algorithms – Logistic Regression ,Bidirectional LSTM* our system will be capable of accurately identifying and classifying toxic comments with high precision.

By doing so, we aspire to mitigate the negative impacts of toxic discourse online, ultimately fostering a more positive and inclusive digital environment.

**OBJECTIVES**

* To Develop an advanced comment toxicity detection system.
* To Accurately identify and classify toxic comments.
* To Utilize extensive training data for system improvement.
* To Enable online platforms to take proactive measures against toxicity.
* To Promote safer and more respectful online environments.
* To Enhance user experience by reducing toxic content.
* To Raise awareness about the importance of addressing comment toxicity.

**CHALLENGES:**

* **Data Quality:** Ensuring the dataset is clean and labeled accurately to train the model effectively.
* **Model Complexity:** Developing a model architecture that balances complexity and performance, especially with large datasets.
* **Computational Resources:** Training deep learning models may require significant computational resources, posing challenges for resource-constrained environments.
* **Overfitting:** Preventing the model from overfitting to the training data and ensuring generalization to unseen data.
* **Evaluation Metrics:** Selecting appropriate evaluation metrics to assess the model's performance accurately.
* **User Interface Design:** Designing an intuitive and user-friendly interface for toxicity prediction that accommodates various user inputs and preferences.

Addressing these challenges is crucial for the successful implementation of the comment toxicity detection system.

**SYSTEM ARCHITECTURE**

**A diagram of a process

Description automatically generated**

**MODULE DESCREPTION**

* **Dataset Pre-processing**: Prepare dataset for analysis.
* **Text Vectorization**: Convert text data into numerical representations.
* **Model Creation**: Construct a sequential model using Keras and TensorFlow.
* **Training**: Train the model to classify toxic comments accurately.
* **Model Evaluation**: Assess model performance using metrics like precision and recall.
* **GUI Interface**: Create an interactive interface using Gradio for toxicity predictions.

The proposed system combines NLP techniques with Keras, TensorFlow, and Gradio for efficient comment toxicity detection.

**BLOCK DIAGRAM**

A diagram of a diagram

Description automatically generated

The block diagram outlines a streamlined deep learning architecture tailored for toxic comments classification. It begins with text input from online platforms, which undergoes embedding to transform it into dense vector representations capturing word semantics.

Bidirectional LSTM layers then process the embedded text bidirectionally, retaining long-term dependencies and contextual nuances.

Subsequent dense layers enhance the model's capacity to discern complex patterns before the classification layer predicts the toxicity level of the input comments, assigning them to predefined toxicity classes like toxic or non-toxic based on probability distributions.

This architecture enables efficient and accurate classification, crucial for mitigating harmful online interactions and fostering safer digital environments.

**DATASET DESCRIPTION**

* Source : Kaggle (Jigsaw Toxic Comment Classification Challenge)
* Purpose : Training machine learning models to classify toxic comments accurately.
* Format : CSV format, with each row representing a comment and its toxicity label.
* Labels : Includes toxic, severe toxic, obscene, threat, insult, and identity hate.
* Features : Mainly comment text, A possibly supplemented with metadata like comment ID and timestamp.
* Size : Typically tens to hundreds of thousands of comments.
* Pre-processing : May involve text cleaning, tokenization, and feature engineering.
* Split : Commonly divided into training, testing, and validation sets for model development and evaluation.
* Importance : Provides a valuable resource for developing effective comment toxicity detection systems.

**ALGORITHMS USED**

***1.LOGISTIC REGRESSION***

Logistic regression is a statistical algorithm used for binary classification problems, which involves predicting one of two possible outcomes. It is a supervised learning algorithm that is commonly used for predicting the probability of an event occurring, based on one or more input variables.

The sigmoid function is referred to as an activation function for logistic regression and is defined as:

where, A equation of a mathematical equation

Description automatically generated

e = base of natural logarithms

value = numerical value one wishes to transform

The following equation represents logistic regression:

here, A close-up of a math problem

Description automatically generated

x = input value , y = predicted output

b0 = bias or intercept term

b1 = coefficient for input (x)

This equation is similar to linear regression, where the input values are combined linearly to predict an output value using weights or coefficient values. However, unlike linear regression, the output value modelled here is a binary value (0 or 1) rather than a numeric value

**Working of Logistic Regression:**

* Logistic Regression calculates a weighted sum of the input features and applies the logistic function to obtain a probability score between 0 and 1.
* The logistic function ensures that the output is bounded, making it suitable for binary classification.
* A threshold (usually 0.5) is set, and if the probability is above this threshold, the instance is classified as 1; otherwise, it is classified as 0

**Training Logistic Regression:**

* The model is trained using a method called Maximum Likelihood Estimation (MLE) to find the values of coefficients (b0,b1,b2A,.....bn) that maximize the likelihood of the observed      data.
* Gradient Descent or other optimization methods are commonly used to iteratively update      the coefficients to minimize the error between predicted and actual outcomes.

A diagram of a logistic curve

Description automatically generated

**2. Bidirectional LSTM**

Bidirectional Long Short-Term Memory (LSTM) is a variant of the LSTM architecture that processes input sequences in both forward and backward directions. It combines the advantages of LSTMs in capturing long-term dependencies with the ability to consider future context as well.

Application:

- Natural Language Processing (NLP): Bidirectional LSTMs are commonly used in tasks such as sentiment analysis, named entity recognition, and text classification, where understanding the context of words in a sentence is crucial.

- Speech Recognition: In speech recognition tasks, bidirectional LSTMs can effectively capture temporal dependencies in audio data, enhancing accuracy in transcribing spoken language.

- Time Series Prediction: Bidirectional LSTMs excel in time series prediction tasks, where historical and future context is essential for accurate forecasting.

Advantages:

1. Capturing Context: By processing input sequences in both directions, bidirectional LSTMs can capture contextual information more effectively than unidirectional models.

2. Long-Term Dependencies: Like traditional LSTMs, bidirectional LSTMs can retain information over long sequences, making them suitable for tasks with dependencies spanning across multiple time steps.

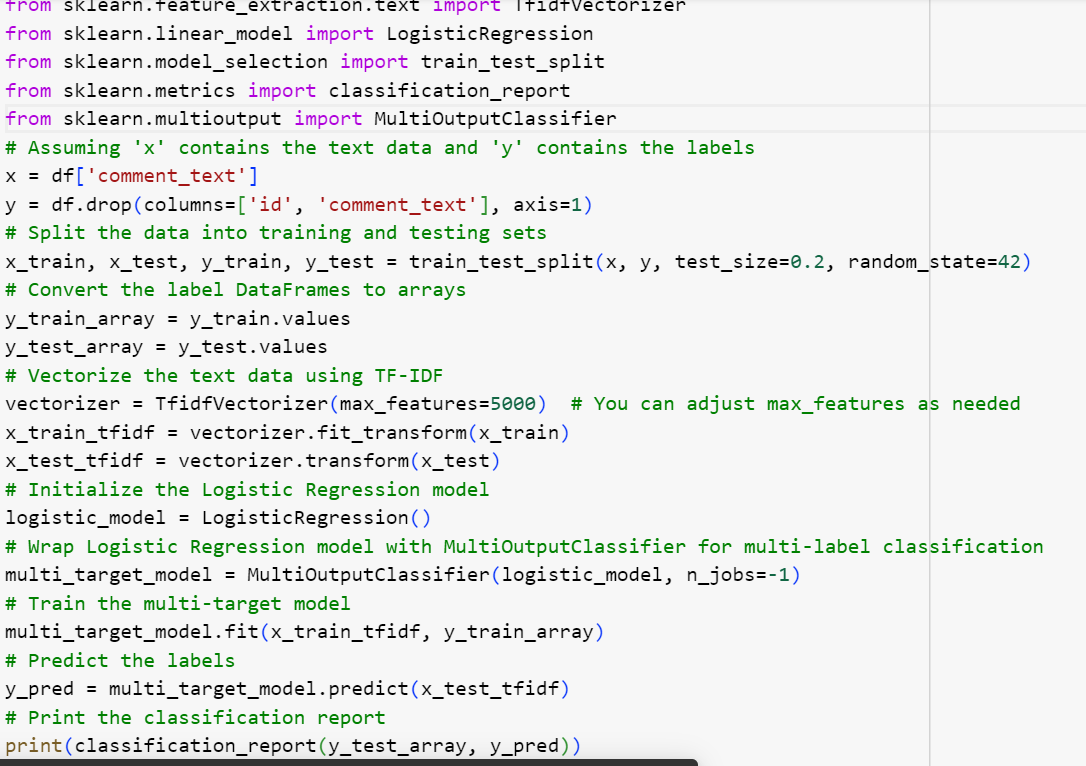
3.Reduced Information Loss: The bidirectional nature helps mitigate the vanishing gradient problem, allowing the model to retain more information from both past and future contexts during training.

4. Enhanced Performance: Bidirectional LSTMs often outperform unidirectional models in tasks requiring a comprehensive understanding of input sequences, leading to improved accuracy and robustness in various applications.

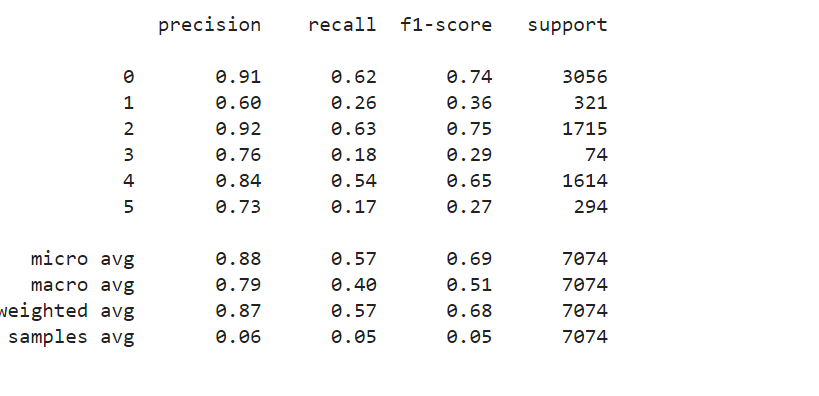
Overall, bidirectional LSTMs offer a powerful solution for sequential data processing tasks by leveraging both past and future context, thereby enhancing model performance and enabling more accurate predictions.

**IMPLEMENTATION:**

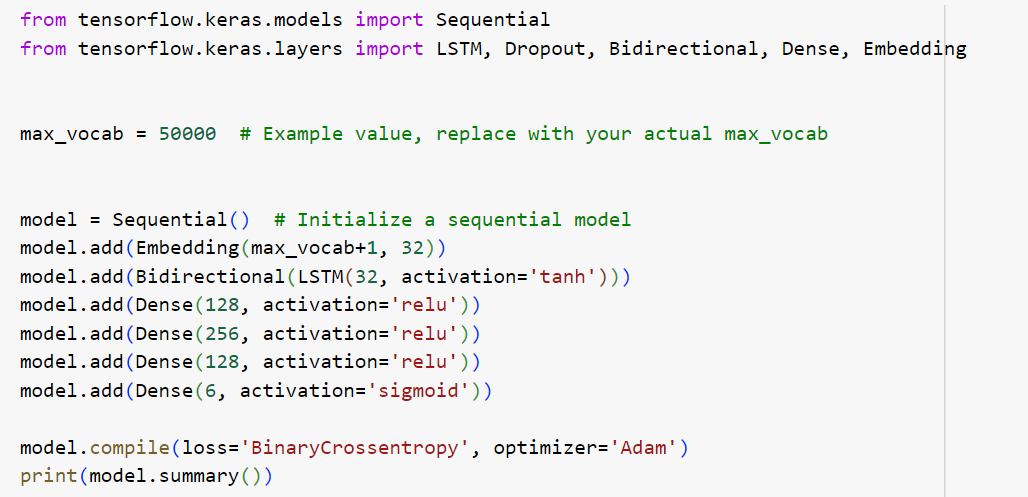
**1.LOGISTIC REGRESSION:**



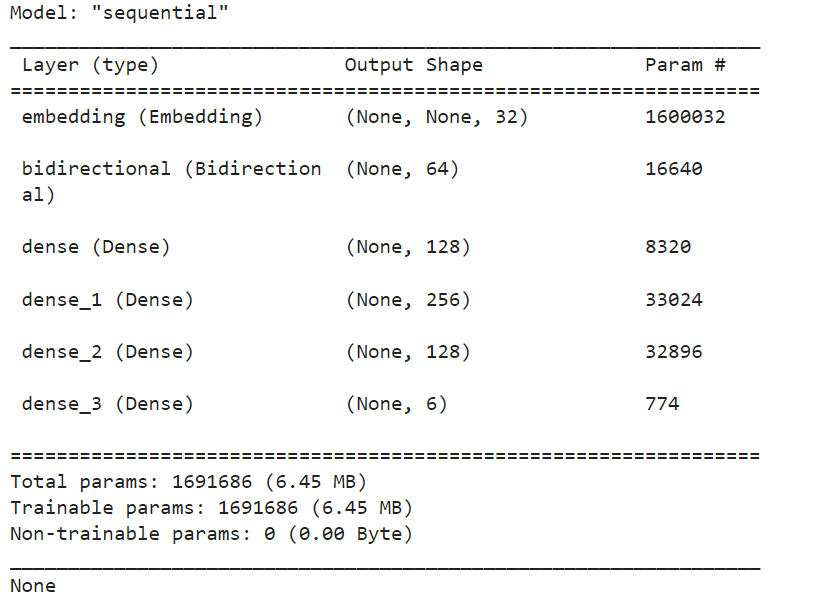
**OUTPUT:**



**2.BIDIRECTIONAL LSTM:**



**OUTPUT:**



**3.GRADIO (GUI):**



**OUTPUT:**

A screenshot of a computer

Description automatically generated

The model *successfully classified comments* into their correct toxicity labels with high accuracy and precision.

Leveraging a sophisticated architecture comprising *bidirectional Long Short-Term Memory (LSTM) layers and dense classification layers*, it effectively captured the nuanced patterns and contextual cues indicative of toxicity within the input text.

By considering both past and future context bidirectionally, the model demonstrated a *robust ability to discern between toxic and non-toxic comments,* contributing to a safer and more inclusive online environment.

**PERFORMANCE METRIC:**

* **PRECISION**: Proportion of correctly identified toxic comments among flagged comments.
* **RECALL** : Proportion of correctly identified toxic comments among all truly toxic comments.
* **ACCURACY** : Overall correctness of toxicity predictions.
* **F1-SCORE** : Balanced measure of precision and recall.
* **PROCESSING SPEED** : Time taken for comment processing and toxicity prediction.
* **USER SATISFACTION** : Feedback on perceived effectiveness of toxicity detection.
* **BIAS AND FAIRNESS** : Assessment of performance across demographic groups.
* **SCALABILITY :** Ability to handle increasing comment volumes.
* **ROBUSTNESS** : Resilience to adversarial attempts to manipulate detection.
* **COMPLIANCE** : Adherence to regulatory requirements and platform guidelines.

**REFERENCES:**

<https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge>.

Alsharef, A., Aggarwal, K., Koundal, D., Alyami, H., & Ameyed, D. (2022). An automated toxicity classification on social media using LSTM and word embedding. Computational Intelligence and Neuroscience, 2022.

https://medium.com/@jeremyarancio/a-data-science-project-toxic-comments-classification-using-na%C3%AFve-bayes-logistic-regression-d745304fcd.