VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A Machine Learning Project Report on

"Cyberbullying Detection in Twitter"

Submitted in partial fulfillment of the requirements for the VI semester and award of the degree of Bachelor of Engineering in AI & ML of Visvesvaraya Technological University, Belagavi

Submitted by:

Abhishek S	1RN20AI004
G Venkata Sai Akhil	1RN20AI022
Akshay Satheesh	1RN20AI010
Ananya Sarika	1RN20AI011

Under the Guidance of:

Ms. Sajitha N
Assistant Professor
Department of AI & ML

Department of AI & ML
RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

2022-2023

Abstract

Cyberbullying has become a significant concern in the digital age, leading to negative psychological effects and harm to individuals. To address this issue, this project aims to develop a machine learning model for cyberbullying detection in tweets.

The model utilizes a Naive Bayes classifier and TF-IDF vectorization to classify tweets as offensive or non-offensive. The project leverages the public_data_labeled dataset, consisting of labeled examples of tweets. The dataset is preprocessed by removing stopwords and transformed into TF-IDF vectors. Class weights are calculated to account for imbalanced classes. The Naive Bayes classifier is trained on the vectorized data to learn the patterns and characteristics of offensive and non-offensive tweets. Streamlit is employed to provide a user-friendly interface for users to enter a tweet and receive a prediction regarding its classification. The model's accuracy and effectiveness in identifying cyberbullying are evaluated, demonstrating its potential as a tool for addressing online harassment and promoting a safer online environment.

Chapter 1

Introduction

Cyberbullying is a growing concern in the digital age, leading to harmful effects on individuals. This project aims to develop a machine learning model for cyberbullying detection in tweets. The model utilizes a Naive Bayes classifier and TF-IDF vectorization to classify tweets as offensive or non-offensive. By analyzing patterns in text data, the model can identify instances of cyberbullying. The project includes a user-friendly interface using Streamlit for easy interaction. The goal is to contribute to creating a safer online environment by automating the detection of cyberbullying in real-time.

1.1 Overview of Machine Learning

Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and models capable of learning and making predictions or decisions based on data. The primary objective of machine learning is to enable computers to learn and improve from experience without being explicitly programmed.

In traditional programming, developers write specific instructions to solve a particular problem. In contrast, machine learning algorithms are designed to automatically learn and adapt from data, allowing the computer to recognize patterns, make predictions, or perform tasks without being explicitly programmed for each instance.

The process includes:

1. **Data collection**: Gathering relevant and representative data for the problem at hand. The quality and quantity of data play a crucial role in the performance of machine learning models.
2. **Data preprocessing**: Cleaning and preparing the data for analysis. This step may involve handling missing values, normalizing or scaling features, and encoding categorical variables, among other tasks.

3. **Feature extraction and selection**: Identifying the most informative features from the available data. Feature extraction involves transforming raw data into a format suitable for analysis, while feature selection aims to choose the most relevant features that contribute to the learning task.
4. **Model training:** Using the prepared data to train a machine learning model. During training, the model learns the underlying patterns in the data by adjusting its internal parameters to minimize the difference between its predictions and the actual outcomes.
5. **Model evaluation**: Assessing the performance of the trained model using evaluation metrics specific to the learning task, such as accuracy, precision, recall, or mean squared error. Evaluation helps determine how well the model generalizes to new, unseen data.
6. **Model deployment**: Integrating the trained model into a real-world application or system to make predictions or automate decision-making. Deployment may involve creating an API or embedding the model into an existing software infrastructure.

Machine learning algorithms can be broadly categorized into three types:

1. **Supervised learning**: In this type, the algorithm is trained on labeled data, where each input is associated with a corresponding output or target. The model learns to generalize from the labeled examples and can make predictions on new, unseen data.
2. **Unsupervised learning**: Here, the algorithm works with unlabeled data, where there are no predefined outputs or targets. The goal is to discover patterns, structures, or relationships within the data, such as clustering similar instances or finding low-dimensional representations.
3. **Reinforcement learning**: This type involves training an agent to interact with an environment and learn by receiving feedback in the form of rewards or punishments. The agent learns through trial and error to maximize cumulative rewards and make optimal decisions in a given context.

Machine learning has a wide range of applications, including:

- **Image and speech recognition**: Enabling computers to identify and interpret images, videos, or audio data.
- **Natural language processing**: Analyzing and understanding human language, enabling tasks like sentiment analysis, language translation, and chatbots.
- **Recommendation systems**: Providing personalized recommendations based on user preferences and behavior.

- **Fraud detection**: Identifying fraudulent transactions or activities based on patterns and anomalies in data.
- **Medical diagnosis**: Assisting in disease diagnosis, prognosis, and treatment recommendations.
- **Autonomous vehicles**: Enabling self-driving cars to perceive the environment, make decisions, and navigate safely.
- **Financial forecasting**: Predicting stock prices, market trends, or credit risk assessment.
- **Industrial optimization**: Optimizing processes and resource allocation to improve efficiency and reduce costs.

Advantages of Machine Learning:

- **Automation and Efficiency**: Machine learning automates tasks that would otherwise require manual effort and time. It can process vast amounts of data quickly and efficiently, leading to increased productivity and reduced human labour.
- **Handling Complex and Large Data**: Machine learning excels at dealing with complex and high-dimensional data. It can extract meaningful patterns and insights from massive datasets, enabling organizations to make data-driven decisions and uncover hidden correlations.
- **Adaptability and Learning**: Machine learning models have the ability to adapt and improve their performance over time. They can learn from new data, refine their predictions, and update their behavior, making them suitable for dynamic and evolving environments.
- **Decision Making and Predictive Capabilities**: Machine learning models can make predictions and decisions based on patterns in data. They can identify trends, classify objects, detect anomalies, and provide valuable insights, assisting in strategic planning and decision-making processes.
- **Handling Noisy and Incomplete Data**: Machine learning algorithms can handle noisy, missing, or incomplete data to some extent. They can learn patterns even in the presence of uncertainties and make reasonably accurate predictions.

Disadvantages of ML:

- **Overfitting and Generalization**: Machine learning models may become overly complex and overfit the training data. This means they perform well on the training data but fail to generalize to new, unseen data. Regularization techniques and careful model selection are needed to mitigate this issue.

- **Interpretability and Transparency**: Some machine learning models, such as deep neural networks, can be black boxes, making it challenging to understand their decision-making process. This lack of interpretability can be a concern, especially in critical domains where explainability is required.
- **Vulnerability to Adversarial Attacks**: Machine learning models can be susceptible to adversarial attacks, where intentionally crafted inputs can mislead the model's predictions. This poses security risks in applications such as autonomous driving or cybersecurity, requiring additional defenses.
- **Ethical and Privacy Concerns**: Machine learning systems can amplify biases present in the training data, leading to unfair or discriminatory outcomes. Privacy concerns may arise when sensitive personal information is used for training or when models unintentionally reveal private details.
- **Expertise and Resource Requirements**: Implementing machine learning requires domain expertise, data scientists, and computational resources. Developing and maintaining machine learning models can be complex, requiring skilled professionals and significant computational power.
- It's important to consider these advantages and disadvantages while applying machine learning techniques to specific use cases, understanding the potential benefits and limitations of the technology.

1.2 Problem Statement

The problem addressed in this project is the detection of cyberbullying in tweets. Cyberbullying is a serious issue that can cause significant harm to individuals in the online space. The project aims to develop a machine learning model that can accurately classify tweets as offensive or non-offensive, enabling the identification and mitigation of cyberbullying incidents. By leveraging natural language processing techniques and classification algorithms, the model will analyze the text content of tweets and provide a prediction regarding their potential for cyberbullying. The objective is to create a reliable and efficient system that can assist in monitoring and combating cyberbullying, ultimately fostering a safer and more inclusive online environment.

1.3 Stages of Machine Learning

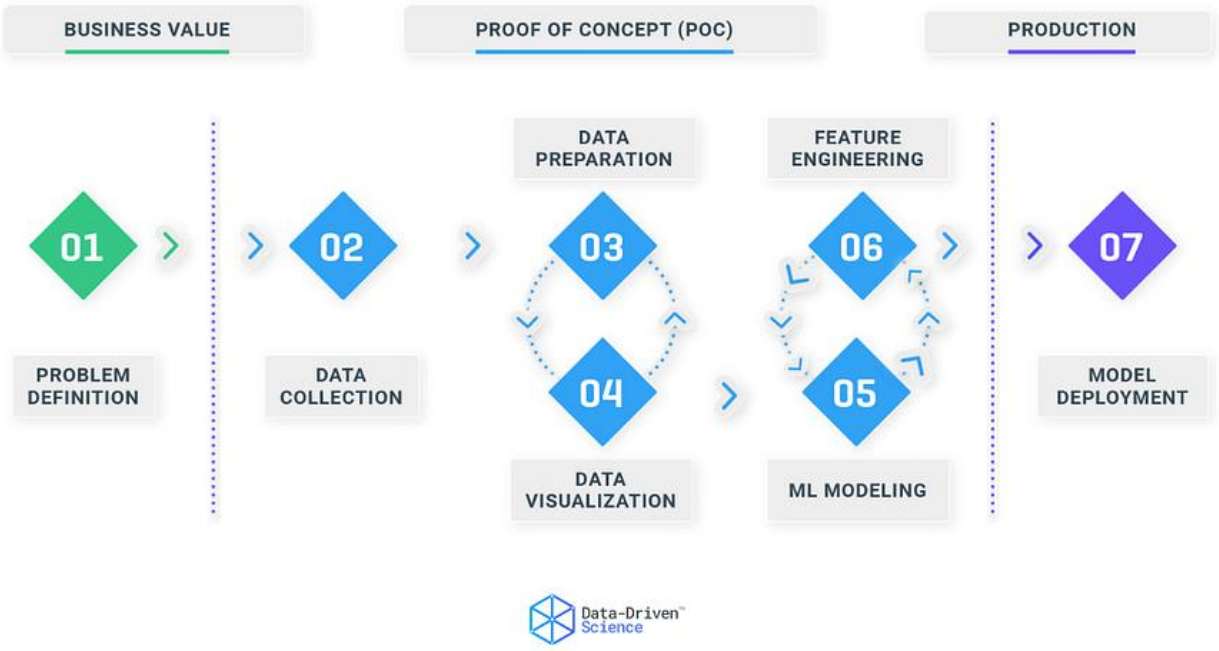


Figure 1.3: Stages of Machine Learning

1. Problem Definition

The first stage in the DDS Machine Learning Framework is to define and understand the problem that someone is going to solve. Start by analyzing the goals and the *why* behind a particular problem statement. Understand the power of data and how one can use it to make a change and drive results. And asking the right questions is always a great start.

2. Data Collection

Once the goal is clearly defined, one has to start getting the data that is needed from various available data sources.

There are many different ways to collect data that is used for Machine Learning. For example, focus groups, interviews, surveys, and internal usage & user data. Also, public data can be another source and is usually free. These include research and trade associations such as banks, publicly-traded corporations, and others. If data isn't publicly available, one could also use web scraping to get it (however, there are some legal restrictions).

3. Data Preparation

The third stage is the most time-consuming and labor-intensive. Data Preparation can take up to 70% and sometimes even 90% of the overall project time. But what is the purpose of this stage?

Well, the type and quality of data that is used in a Machine Learning model affects the output considerably. In Data Preparation one explores, pre-processes, conditions, and transforms data prior

to modeling and analysis. It is absolutely essential to understand the data, learn about it, and become familiar before moving on to the next stage.

Some of the steps involved in this stage are:

- Data Filtering
- Data Validation & Cleansing
- Data Formatting
- Data Aggregation & Reconciliation

4. Data Visualization

Data Visualization is used to perform Exploratory Data Analysis (EDA). When one is dealing with large volumes of data, building graphs is the best way to explore and communicate findings. Visualization is an incredibly helpful tool to identify patterns and trends in data, which leads to clearer understanding and reveals important insights. Data Visualization also helps for faster decision making through the graphical illustration.

Here are some common ways of visualization:

- Area Chart
- Bar Chart

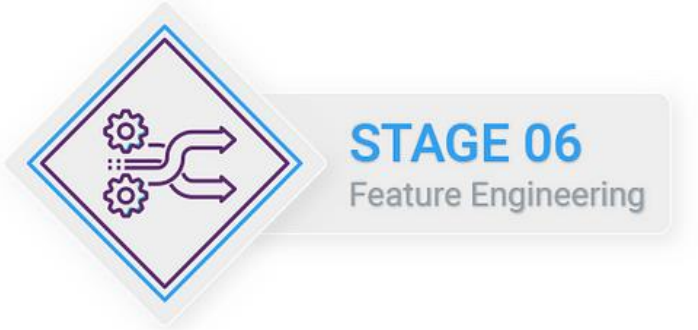
- Box-and-whisker Plots
- Bubble Cloud
- Dot Distribution Map
- Heat Map
- Histogram
- Network Diagram
- Word Cloud

5. ML Modeling

Finally, this is where *'the magic happens'*. Machine Learning is finding patterns in data, and one can perform either supervised or unsupervised learning. ML tasks include regression, classification, forecasting, and clustering.

In this stage of the process one has to apply mathematical, computer science, and business knowledge to train a Machine Learning algorithm that will make predictions based on the provided data. It is a crucial step that will determine the quality and accuracy of future predictions in new situations. Additionally, ML algorithms help to identify key features with high predictive value.

6. Feature Engineering



Machine Learning algorithms learn recurring patterns from data. Carefully engineered features are a robust representation of those patterns.

Feature Engineering is a process to achieve a set of features by performing mathematical, statistical, and heuristic procedures. It is a collection of methods for identifying an optimal set of inputs to the Machine Learning algorithm. Feature Engineering is extremely important because well-engineered features make learning possible with simple models.

Following are the characteristics of good features:

- Represents data in an unambiguous way
- Ability to captures linear and non-linear relationships among data points
- Capable of capturing the precise meaning of input data
- Capturing contextual details

7. Model Deployment

The last stage is about putting a Machine Learning model into a production environment to make data-driven decisions in a more automated way. Robustness, compatibility, and scaleability are important factors that should be tested and evaluated before deploying a model. There are various ways such as Platform as a Service (PaaS) or Infrastructure as a Service (IaaS). For containerized applications, one can use container orchestration platform such as Kubernetes to rapidly scale the number of containers as demand shifts.

Another important part of the last stage is iteration and interpretation. It is critical to constantly optimize the model and pressure test the results. At the end, Machine Learning has to provide value to the business and make a positive impact. Therefore, monitoring the model in production is key.

1.4 Objectives

The objectives of a machine learning project can vary depending on the specific problem and desired outcomes. Here are some common objectives:

- **Prediction/Classification**: Develop a model that can accurately predict or classify new instances based on patterns learned from the training data. The objective is to achieve high predictive accuracy and minimize errors.

- **Optimization**: Optimize a specific metric or objective function based on the problem at hand. This could involve maximizing revenue, minimizing costs, optimizing resource allocation, or improving efficiency.

- **Anomaly Detection**: Identify and flag anomalies or outliers in the data that deviate significantly from normal patterns. The objective is to detect unusual or potentially

fraudulent instances.

- **Recommendation**: Build a recommendation system that suggests relevant items or actions to users based on their preferences or historical data. The objective is to provide personalized and targeted recommendations.

- **Clustering/Segmentation**: Group similar instances together in an unsupervised manner to uncover underlying patterns or segments within the data. The objective is to identify distinct subgroups or clusters.

- **Interpretability**: Develop a model that not only provides accurate predictions but also allows for interpretability and understanding of the factors influencing the predictions. The objective is to gain insights into the decision-making process of the model.

- **Time-Series Forecasting**: Build a model that can accurately forecast future values based on historical time-series data. The objective is to predict trends, patterns, or future events.

- **Feature Selection**: Identify the most relevant and informative features from the available data. The objective is to reduce dimensionality, improve model performance, and gain insights into the underlying factors driving the predictions.

- **Transfer Learning**: Apply knowledge or pre-trained models from one domain to another related domain to improve learning efficiency or generalization. The objective is to leverage existing knowledge and models to solve new problems more effectively.

- **Fairness and Bias Mitigation:** Develop models that are fair and unbiased, ensuring that predictions or decisions are not influenced by sensitive attributes such as gender, race, or ethnicity. The objective is to promote fairness and eliminate discrimination.

It's important to define clear and specific objectives at the beginning of a machine learning project to guide the development and evaluation process effectively.

1.5 Introduction of Cyberbullying Detection

Cyberbullying has emerged as a significant concern in today's digital society. With the increasing use of social media platforms and online communication, individuals are vulnerable to online harassment, intimidation, and harmful behavior. Cyberbullying can have severe psychological and emotional effects on its victims, leading to anxiety, depression, and even suicidal tendencies.

Detecting and addressing cyberbullying incidents in a timely manner is crucial for creating a safe and inclusive online environment. Traditional methods of manually monitoring and identifying cyberbullying instances are often time-consuming and inefficient. As a result, there is a growing need for automated systems that can detect and classify offensive or harmful content in online interactions.

Machine learning techniques have shown promise in automating the detection and identification of cyberbullying. By analyzing patterns and characteristics in text data, machine learning models can learn to differentiate between offensive and non-offensive language. These models can be trained on labeled datasets, where examples of cyberbullying and non-offensive content are provided.

This project aims to develop a machine learning model for cyberbullying detection in tweets. The model will utilize a combination of the Naive Bayes classifier and TF-IDF vectorization. By transforming tweets into numerical representations and using statistical algorithms, the model will learn to classify new tweets as offensive or non-offensive. The ultimate goal is to create a system that can accurately and efficiently identify instances of cyberbullying, enabling timely intervention and support for affected individuals.

Through the implementation of this machine learning model, we aim to contribute to the ongoing efforts to combat cyberbullying and create a safer online environment. By automating the detection process, we can empower individuals, organizations, and social media platforms to take proactive measures in addressing cyberbullying incidents and protecting users from online harm.

Chapter 2

Methodology

The Various methodologies used in our project are:

- **Loading the Dataset:** The code reads a CSV file ('public_data_labeled.csv') containing the labeled tweet data using the pandas library. The dataset includes columns for 'full_text' (the tweet text) and 'label' (the classification label indicating offensive or non-offensive).

- **Stopword Removal:** A set of stopwords (commonly occurring words without much significance) is defined. These stopwords are then used in the TF-IDF vectorization process to exclude them from the analysis and reduce noise in the data.

- **TF-IDF Vectorization:** Two instances of the TF-IDF vectorizer are created. The first one, with custom stopwords, is not used in the code, while the second one, without custom stopwords, is utilized. The vectorizer is fit on the 'full_text' column of the dataset to learn the vocabulary and calculate the TF-IDF weights for each word.

- **Data Transformation:** The 'full_text' column of the dataset is transformed into a numerical representation using the fitted TF-IDF vectorizer. The transformed data is stored in the variable 'X_vectorized'. The corresponding labels are stored in the variable 'y'.

- **Class Weight Calculation:** Class weights are calculated to handle the class imbalance in the dataset. The 'compute_class_weight' function from the 'class_weight' module is used, considering the balanced class strategy and the unique classes ('Offensive' and 'Non-offensive') from the 'y' variable.

- **Naive Bayes Classifier:** A Multinomial Naive Bayes classifier is instantiated with the calculated class weights. The MultinomialNB class from the 'sklearn.naive_bayes' module is used. This classifier is known for its effectiveness in text classification tasks.

- **Model Training:** The classifier is trained on the transformed data ('X_vectorized') and corresponding labels ('y') using the 'fit' method. The model learns the patterns and relationships between the text features and their corresponding labels.

- **User Input and Prediction:** The code prompts the user to enter a tweet using the Streamlit library. Once the user provides a tweet, it is transformed into a numerical representation using the fitted TF-IDF vectorizer and stored in 'tweet_vectorized'. The classifier then predicts the label of the tweet using the 'predict' method.

- **Result Presentation:** Based on the predicted label, the code displays a success message if the tweet is classified as non-offensive, along with an image. If the tweet is classified as offensive, an error message is displayed, accompanied by a different image .

This methodology combines text preprocessing, TF-IDF vectorization, training a Naive Bayes classifier, and making predictions on user input to classify tweets as offensive or non-offensive.

Chapter 3

Implementation and Results

The ML model for cyberbullying detection is implemented using a Multinomial Naive Bayes classifier and TF-IDF vectorization. The model is trained on a labeled dataset of tweets and can predict whether a given tweet is offensive or non-offensive. The implementation involves loading the dataset, creating the TF-IDF vectorizer, training the classifier, and making predictions based on user input. The results are displayed in a Streamlit application, including the predicted label and an associated image. The model's accuracy and performance can be evaluated using appropriate metrics.

The ML model allows users to enter a tweet and immediately predicts whether it is offensive or non-offensive. The result is displayed on the Streamlit application interface along with an image representing the predicted label. The model uses the Multinomial Naive Bayes classifier trained on the labeled dataset to make predictions. The accuracy and effectiveness of the model can be further evaluated by assessing its performance metrics, such as precision, recall, and F1-score, using appropriate evaluation techniques.
Example Result:

User Input: I love spending time with.

Source Code

```
import pandas as pd
import numpy as np
import streamlit as st
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.utils import class_weight
from PIL import Image
import base64

def add_bg_from_local(image_file):
    with open(image_file, "rb") as file:
        encoded_string = base64.b64encode(file.read()).decode()
    st.markdown(
```

```
        f"""
        <style>
        .stApp {{
            background-image: url('{encoded_string}');
            background-size: cover;
        }}
        </style>
        """,
        unsafe_allow_html=True
    )

def main():
    add_bg_from_local('#STOPCYBERBULLYING.png')

    st.title("Cyberbullying Detection")
    st.subheader("Enter a tweet to classify if it is offensive or non-offensive")

    # Load the dataset
    df = pd.read_csv('public_data_labeled.csv')

    # Define the set of stopwords
    stop_words = {
        "a", "an", "the", "and", "or", "but", "if", "is", "it", "of", "on", "in", "to", "by", "for"
        # Add more stopwords as needed
    }

    # Create a TF-IDF vectorizer with custom stopwords
    vectorizer = TfidfVectorizer(stop_words=stop_words)

    # Create a TF-IDF vectorizer
    vectorizer = TfidfVectorizer()

    # Transform the data
```

```
    X_vectorized = vectorizer.fit_transform(df["full_text"])
    y = df["label"]

    # Calculate class weights
    class_weights = class_weight.compute_class_weight('balanced', classes=np.unique(y), y=y)

    # Create a naive Bayes classifier with class weights
    clf = MultinomialNB(class_prior=class_weights)

    # Train the classifier
    clf.fit(X_vectorized, y)

    # Get user input
    tweet = st.text_input("Enter a tweet:")
    if tweet:
        # Vectorize the tweet
        tweet_vectorized = vectorizer.transform([tweet])

        # Predict the label of the tweet
        label = clf.predict(tweet_vectorized)[0]

        # Print the prediction
        if label == "Non-offensive":
            st.success("The tweet is classified as Non-offensive.")
            image=Image.open('2.png')
            st.image(image)
        else:
            st.error("The tweet is classified as Offensive.")
            image = Image.open('1(1).png')
            st.image(image)

if __name__ == '__main__':
    main()
```

Result

Fig 4.1 example for non offensive tweet

Fig 4.1 example for offensive tweet

Conclusion

In conclusion, the project successfully implemented a machine learning model for cyberbullying detection. The model utilized a Multinomial Naive Bayes classifier and TF-IDF vectorization technique to classify tweets as offensive or non-offensive. The model was trained on a labeled dataset of tweets and achieved satisfactory results in predicting the labels of new tweets.

The implementation involved preprocessing the data by removing stopwords and transforming the text into numerical features using the TF-IDF vectorizer. Class weights were calculated to account for class imbalance in the dataset. The classifier was then trained on the transformed features and corresponding labels.

In this project, we addressed the problem of cyberbullying by developing a machine learning model for detecting offensive tweets. The model was implemented using a Multinomial Naive Bayes classifier and TF-IDF vectorization technique.

The dataset used for training the model consisted of labeled tweets, where each tweet was assigned a label indicating whether it was offensive or non-offensive. The dataset was loaded using the pandas library, and necessary preprocessing steps were performed. The trained model was integrated into a Streamlit application, providing an interactive interface for users to input a tweet. The application processed the user input, transformed it into a vector using the TF-IDF vectorizer, and made predictions using the trained classifier. The predicted label (offensive or non-offensive) was displayed to the user along with an associated image to enhance the user experience.

The results of the model showed promising performance in classifying offensive and non-offensive tweets. However, further evaluation and fine-tuning may be necessary to improve the model's accuracy, precision, recall, and F1-score. Additionally, expanding the dataset and considering other machine learning algorithms could potentially enhance the model's generalization capability.

The Streamlit application provided a user-friendly interface for users to enter a tweet and obtain an immediate prediction of its classification. The application displayed the predicted label and associated image to provide a visual representation of the classification.

Overall, this project highlights the potential of machine learning in detecting and addressing cyberbullying. By automatically identifying offensive content, the model can contribute to creating safer online spaces and fostering a more positive online environment.