Hate Speech Detection

Project By:S.P.V Karthik

Abstract:

The spread of hate speech has been accelerated in the digital age due to the widespread use of internet communication, which presents serious threats to public safety and harmony. This project seeks to create an advanced hate speech detection system that uses natural language processing (NLP) and machine learning techniques to automatically recognize and report abusive language in real time .Our method entails collecting and annotating a diversified dataset from several social media sites in order to train and assess a strong classification model.

Introduction:

The fast expansion of digital communication channels has changed the way people interact, share ideas, and participate in global discussions. While digital interconnection has many advantages, it has also boosted the propagation of hate speech, which poses serious threats to individual well-being and societal harmony. Hate speech, defined as rhetoric that encourages discrimination, animosity, or violence against individuals or groups based on characteristics such as race, religion, ethnicity, gender, or sexual orientation, is a major problem that requires effective solutions .Automated systems that can quickly and reliably identify hate speech in real time are becoming more and more necessary.

This project focuses on developing a comprehensive hate speech detection system utilizing advanced natural language processing (NLP) and machine learning techniques. By leveraging large datasets from diverse social media platforms, the system aims to recognize and categorize hate speech across multiple contexts and languages.

This project aims to create a comprehensive hate speech detection system using modern natural language processing (NLP) and machine learning techniques. The system tries to recognize and categorize hate speech in many situations and languages by exploiting massive datasets from various social media platforms.

Social media platforms need to detect hate speech and prevent it from going viral or ban it at the right time. So in this project, I will walk you through the task of hate speech detection with machine learning using the Python programming language.

Aims and Objective:

This aims to classify the textual content into hate speech , offensive speech and neither hate nor offensive speech.

The proposed solutions classify information as hate speech by using various feature engineering techniques and machine learning algorithms.

- The essence of this model is to develop an automated system approach for detecting hate speech and offensive language.
- According to the model ,Texts are classified into three groups based on their sentiment and other features.

Data collection:

The models used for machine learning must ingest enormous volumes of structured training data in order to create intelligent, understandable applications. Gathering enough training data is the first step in resolving any AI-based machine learning issue. Data collecting is gathering information from a variety of sources, both online and offline, and then combining it.Every dataset contains errors. This is why data preparation is crucial during the machine learning process.

In this project dataset is collected from twitter dataset which is downloaded from Kaggle .

Twitter Dataset:

count	hate_spee	offensive_	neither_cc	class	tweet
3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you should always take the trash out...
3	0	3	0	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
3	0	3	0	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
3	0	2	1	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
6	0	6	0	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya 
3	1	2	0	1	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! 😂😂😂"
3	0	3	0	1	!!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch .. I got too much shit going on!"
3	0	3	0	1	!!!!“@selfiequeenbri: cause I'm tired of you big bitches coming for us skinny girls!!”
3	0	3	0	1	" & you might not get ya bitch back & thats that "
3	1	2	0	1	"
3	0	3	0	1	" Keeks is a bitch she curves everyone " lol I walked into a conversation like this. Smh
3	0	3	0	1	" Murda Gang bitch its Gang Land "

Data cleaning:

Data cleaning is a key component of machine learning. It is an important aspect of the model-building process. It is certainly not the fanciest part of machine learning, but

there are no hidden tricks or secrets to discover. However, the success or failure of a project is dependent on adequate data cleaning. If we have a well-cleaned dataset, we may be able to produce good results using simple techniques, which can be quite advantageous at times, particularly in terms of computing when the dataset is enormous. Obviously, different types of data require different kinds of cleansing. However, this systematic approach is always a good place to start.

Data Analysis and Exploration:

Labelling the class hate ,offensive and neither hate nor offensive speech.

Index	tweet	label
0	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you should always take the trash out...	neither hate nor offensive
1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!	offensive language
2	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit	offensive language
3	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny	offensive language
4	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya 	offensive language
5	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! 😂😂😂"	offensive language
6	!!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch .. I got too much shit going on!"	offensive language

Cleaning the text.

```
def data_cleaning(text):
    text=re.sub(r'@\w+', ' ', text)
    text= re.sub(r'\bRT\b', ' ', text)
    text=html.unescape(text)
    text= re.sub(r'http[s]?://\S+', '', text)
    text=re.sub('[^a-zA-Z]', ' ', text)
    text=text.lower()
    text=text.split()
    text=[wordnet.lemmatize(word) for word in text if not word in stop_words]
    text=' '.join(text)
    return text
```

Dataset after cleaning the text.

Index	tweet	label
0	woman complain cleaning house man always take trash	neither hate nor offensive
1	boy dat cold tyga dwn bad cuffin dat hoe st place	offensive language
2	dawg ever fuck bitch start cry confused shit	offensive language
3	look like tranny	offensive language
4	shit hear might true might faker bitch told ya	offensive language
5	shit blow claim faithful somebody still fucking hoe	offensive language
6	sit hate another bitch got much shit going	offensive language

Data Modeling:

In machine learning, data modeling involves creating a mathematical representation of a dataset to capture its underlying patterns, relationships, and properties.

1.Decision tree classifiers:

Decision tree modeling is a prominent machine learning technique for classification and regression problems. It is a form of supervised learning method that use a tree-like model of decisions and their possible consequences.

2.KNeighbours classifiers:

The k-Nearest Neighbors (k-NN) classifier is a simple and effective machine learning algorithm used for both classification and regression tasks. It is based on the concept of similarity, where the algorithm predicts the output for a given input by looking at the 'k' closest examples in the feature space.

3.Logistic Regression:

Logistic regression is a statistical method used for binary classification tasks, where the goal is to predict the probability of an instance belonging to one of two classes. It is a type of generalized linear model that extends linear regression to classification problems by using a logistic function to model the probability of the outcome.

4.Naive bayes:

Naive Bayes is a family of simple yet effective probabilistic algorithms based on applying Bayes' theorem with the assumption of independence among predictors. It is

primarily used for classification tasks and is known for its efficiency and ease of implementation.

Accuracy and precision of above classification techniques

Name of the model	confusion matrix	accuracy(%)	bias(%)	variance(%)
DecisionTreeClassifier	[[173 43 259] [42 1169 182] [324 175 5812]]	87.4679056	99.5844375	87.467906
LogisticRegression	[[115 54 306] [17 1238 138] [104 166 6041]]	90.4022497	95.3143821	90.40225
KNeighborsClassifier	[[158 84 233] [11 1183 199] [188 321 5802]]	87.3334148	90.580583	87.333415
Bernoulli Naive Bayes	[[78 58 339] [21 1095 277] [124 143 6044]]	88.2381709	91.2189834	88.238171
Gaussian Naive Bayes	[[198 95 182] [276 847 270] [2666 1494 2151]]	39.0756816	46.1394845	39.075682
Multinomial Naive Bayes	[[109 52 314] [31 1093 269] [139 140 6032]]	88.4460203	91.6947723	88.44602

Optimisation and deployment:

For our model, from the above data set we can use either Logistic regression or decision tree classifier

so for this let us consider Logistic regression. Decision tree classifier is present because of its high interpretability.

Logistic Regression:

Pros: High accuracy, low variance, good for binary classification problems like hate speech detection.

Cons: May not capture complex patterns as well as some other models.

Use Case: Suitable if you need a balance between performance and simplicity, and if you can afford to trade off a bit of interpretability for higher accuracy.

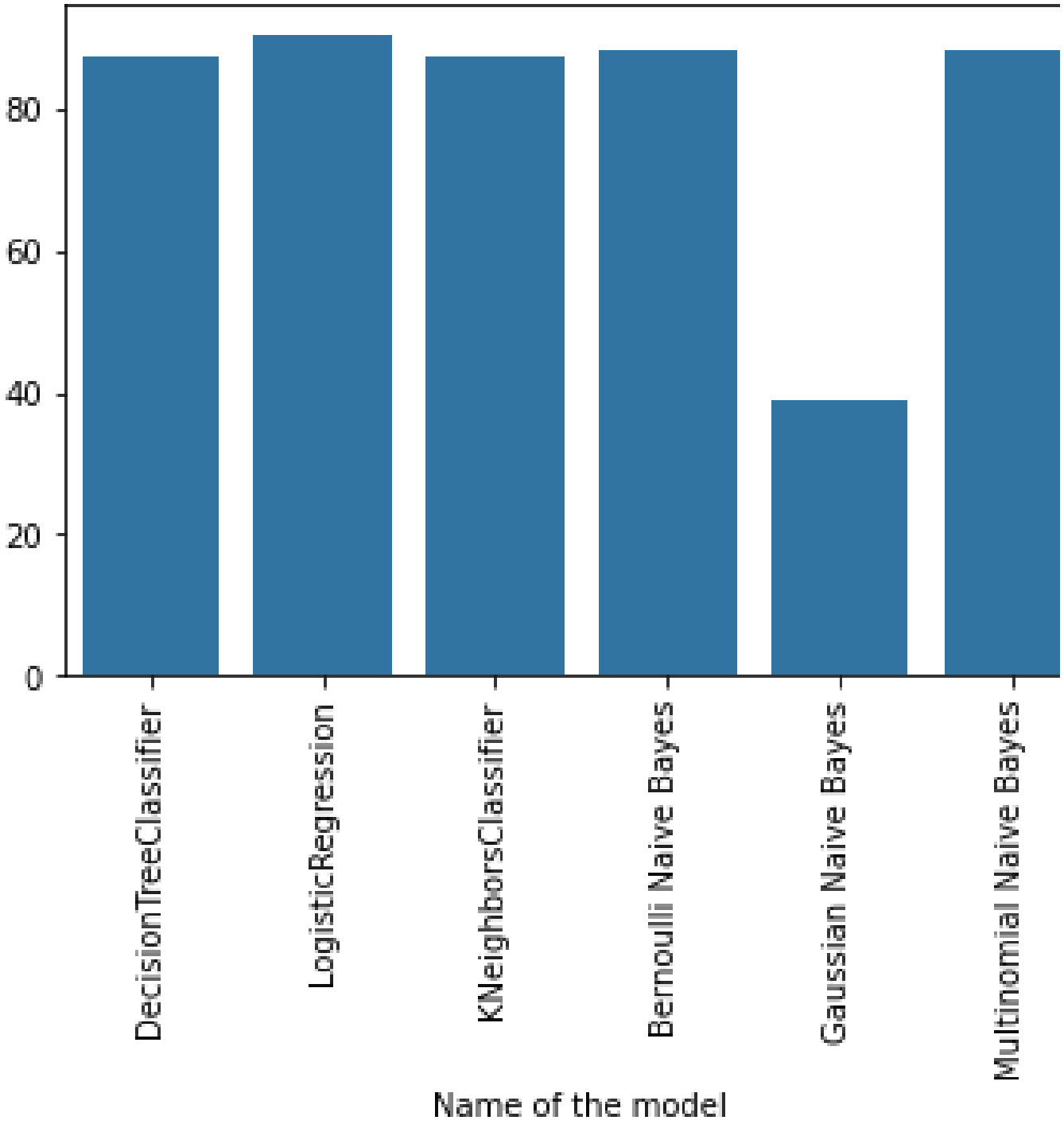
Decision Tree Classifier:

Pros: Highly interpretable, can capture complex patterns.

Cons: High bias, which can be problematic for generalization.

Use Case: Good if interpretability is crucial and you need to understand the decision process for each classification.

However, you might need to prune the tree to avoid overfitting.'''



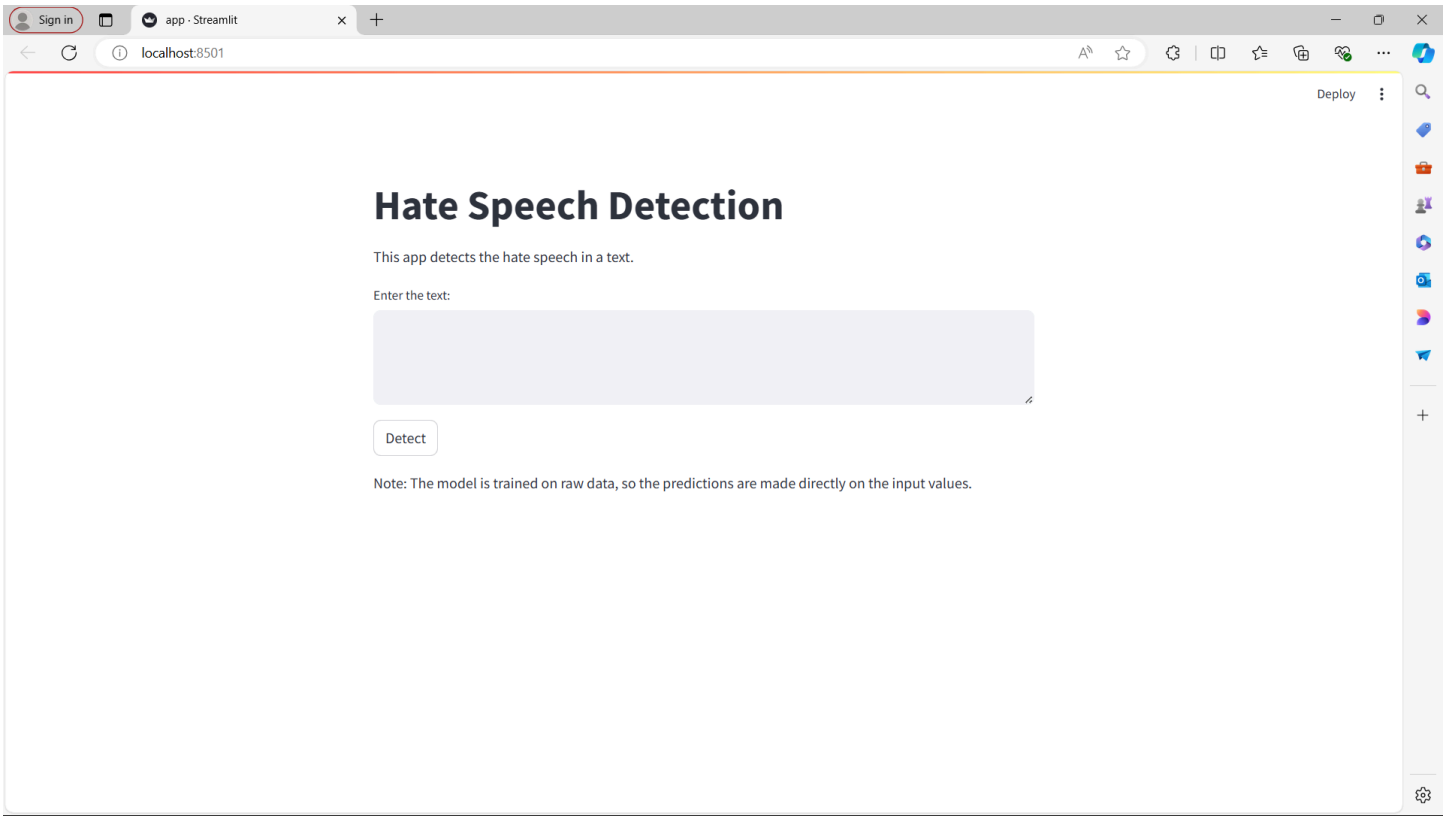
Deployment:

The Python pickle package is used to serialize and deserialize a Python object structure. Any Python object can be pickled and stored to disk. Pickle "serializes" an object before writing it to a file. Pickling is a method for converting a Python object (list, dictionary, etc.) into a character stream.

```
import streamlit as st
import pickle
```

```
# Load pre-trained model
with open('final_model1.pkl', 'rb') as file:
    model = pickle.load(file)

# Load the fitted CountVectorizer
with open('count_vectorizer.pkl', 'rb') as file:
    cv = pickle.load(file)

st.title('Hate Speech Detection')

st.write("""
This app detects the hate speech in a text.
""")

text=st.text_area("Enter the text:")
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import html
stop_words=set(stopwords.words('english'))
wordnet=WordNetLemmatizer()
def data_cleaning(text):
    text=re.sub(r'@\w+', ' ', text)
    text= re.sub(r'\bRT\b', ' ', text)
    text=html.unescape(text)
    text= re.sub(r'http[s]?://\S+', '', text)
    text=re.sub('[^a-zA-Z]', ' ', text)
    text=text.lower()
    text=text.split()
    text=[wordnet.lemmatize(word) for word in text if not word in stop_words]
    text=' '.join(text)
    return text
```

```
if st.button("Detect"):
    cleaned_text = data_cleaning(text)
    transformed_text = cv.transform([cleaned_text]).toarray()  # Transform expects a list of texts
    # Predict using the model
    prediction = model.predict(transformed_text)
    # Since the prediction is an array like array(['hate speech'], dtype=object)
    st.write(f"Detection: {prediction[0]}")

st.write("""Note: The model is trained on raw data, so the predictions are made directly on the input values.""")
```



Conclusion:

In conclusion, hate speech identification is an important task in machine learning. Hate speech detection seeks to recognize and categorize objectionable or damaging language in a variety of text formats, including social media postings, online comments, and public forums. The goal is to create models and algorithms that can automatically recognize and flag such information, so contributing to safer online environments and encouraging inclusive and polite conversation.

Future scope of the model:

Hyper parameter tuning and more diverse datasets are required to improve the accuracy of the model.