Abstract

This project aims to develop a model that can accurately identify hate speech in text data. By leveraging machine learning algorithms and natural language processing techniques, the objective is to differentiate between hateful and non-hateful content to mitigate the negative impact of hate speech on social platforms. The developed system is trained and tested using a labeled dataset to achieve optimal performance.

Objective

The primary objective of this project is to develop a machine learning model capable of detecting hate speech in textual data. This involves preprocessing text, training an appropriate model, evaluating its performance, and ensuring that it can be effectively used for hate speech moderation.

Introduction

Hate speech is a form of expression that can promote discrimination and violence, especially on social media platforms. As the online community grows, the need for automated tools to detect and mitigate hate speech becomes increasingly vital. This project addresses this need by developing a hate speech detection model using machine learning and natural language processing.

Methodology

- **Data Collection**: A labeled dataset containing examples of hate speech and non-hate speech was used for training and testing.
- **Data Preprocessing**: Text data was cleaned by removing punctuation, converting to lowercase, and removing stop words.
- **Feature Extraction**: Techniques like TF-IDF or word embeddings were employed to transform the text into a numerical form suitable for model training.
- **Model Training**: Various machine learning models (e.g., logistic regression, SVM, or deep learning) were trained using the preprocessed data.
- **Evaluation**: The models were evaluated using metrics like accuracy, precision, recall, and F1-score.

Importing the dataset

```
In [131]: import pandas as pd
          import numpy as np

In [132]: df = pd.read_csv("Hatespeech_data.csv")

In [133]: df
```

Out[133]:

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	tweet
0	0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't...
1	1	3	0	0	3	0	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2	2	3	0	0	3	0	!!!!!! RT @UKKindOfBrand Dawg!!!! RT @80sbaby...
3	3	3	0	0	2	1	!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4	4	6	0	0	6	0	1	!!!!!!!!! RT @ShenikaRoberts: The shit you...
...	...	...	...	...	...	...	...	...
24778	25291	3	0	0	2	1	1	you's a muthaf***in lie “@LifeAsKing: @2...
24779	25292	3	0	0	1	2	2	you've gone and broke the wrong heart baby, an...
24780	25294	3	0	0	3	0	1	young buck wanna eat!.. dat nigguh like i ain...
24781	25295	6	0	0	6	0	1	you got wild bitches tellin you lies
24782	25296	3	0	0	0	3	2	---Ruffled	Ntac Eileen Dahlia - Beautiful col...

24783 rows × 7 columns

```
In [134]: df.isnull()
```

Out[134]:

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	tweet
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
24778	False	False	False	False	False	False	False
24779	False	False	False	False	False	False	False
24780	False	False	False	False	False	False	False
24781	False	False	False	False	False	False	False
24782	False	False	False	False	False	False	False

24783 rows × 7 columns

```
In [135]: df.isnull().sum()
```

Out[135]:

```
Unnamed: 0           0
count                0
hate_speech          0
offensive_language   0
neither              0
class                0
tweet                0
dtype: int64
```

```
In [136]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24783 entries, 0 to 24782
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Unnamed: 0          24783 non-null  int64 
 1   count               24783 non-null  int64 
 2   hate_speech         24783 non-null  int64 
 3   offensive_language  24783 non-null  int64 
 4   neither             24783 non-null  int64 
 5   class               24783 non-null  int64 
 6   tweet               24783 non-null  object
dtypes: int64(6), object(1)
memory usage: 1.3+ MB
```

```
In [137]: df.describe()
```

Out[137]:

	Unnamed: 0	count	hate_speech	offensive_language	neither	class
count	24783.000000	24783.000000	24783.000000	24783.000000	24783.000000	24783.000000
mean	12681.192027	3.243473	0.280515	2.413711	0.549347	1.110277
std	7299.553863	0.883060	0.631851	1.399459	1.113299	0.462089
min	0.000000	3.000000	0.000000	0.000000	0.000000	0.000000
25%	6372.500000	3.000000	0.000000	2.000000	0.000000	1.000000
50%	12703.000000	3.000000	0.000000	3.000000	0.000000	1.000000
75%	18995.500000	3.000000	0.000000	3.000000	0.000000	1.000000
max	25296.000000	9.000000	7.000000	9.000000	9.000000	2.000000

```
In [138]: df["Labels"] = df["class"].map({0: "Hate Speech",
                                          1: "Offensive Language",
                                          2: "No hate or Offensive Language"})
```

```
In [139]: df
```

Out[139]:

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	tweet	Labels
0	0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't...	No hate or Offensive Language
1	1	3	0	0	3	0	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...	Offensive Language
2	2	3	0	0	3	0	!!!!!! RT @UKKindOfBrand Dawg!!!! RT @80sbaby...	Offensive Language
3	3	3	0	0	2	1	!!!!!!! RT @C_G_Anderson: @viva_based she lo...	Offensive Language
4	4	6	0	0	6	0	!!!!!!!!! RT @ShenikaRoberts: The shit you...	Offensive Language
...	...	...	...	...	...	...	...	...
24778	25291	3	0	0	2	1	you's a muthaf***in lie “@LifeAsKing: @2...	Offensive Language
24779	25292	3	0	0	1	2	you've gone and broke the wrong heart baby, an...	No hate or Offensive Language
24780	25294	3	0	0	3	0	young buck wanna eat!.. dat nigguh like i ain...	Offensive Language
24781	25295	6	0	0	6	0	you got wild bitches tellin you lies	Offensive Language
24782	25296	3	0	0	0	3	---Ruffled	Ntac Eileen Dahlia - Beautiful col...	No hate or Offensive Language

24783 rows × 8 columns

```
In [140]: data = df[["tweet", "Labels"]]
```

```
In [141]: data
```

Out[141]:

	tweet	Labels
0	!!! RT @mayasolovely: As a woman you shouldn't...	No hate or Offensive Language
1	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...	Offensive Language
2	!!!!!! RT @UKKindOfBrand Dawg!!!! RT @80sbaby...	Offensive Language
3	!!!!!!! RT @C_G_Anderson: @viva_based she lo...	Offensive Language
4	!!!!!!!!! RT @ShenikaRoberts: The shit you...	Offensive Language
...	...	...
24778	you's a muthaf***in lie “@LifeAsKing: @2...	Offensive Language
24779	you've gone and broke the wrong heart baby, an...	No hate or Offensive Language
24780	young buck wanna eat!.. dat nigguh like i ain...	Offensive Language
24781	you got wild bitches tellin you lies	Offensive Language
24782	---Ruffled	Ntac Eileen Dahlia - Beautiful col...	No hate or Offensive Language

24783 rows × 2 columns

Data preprocessing

```
In [142]: import re
          import nltk
          nltk.download('stopwords')
          import string

          [nltk_data] Downloading package stopwords to
          [nltk_data]     C:\Users\ANIlAppData\Roaming\nltk_data...
          [nltk_data]   Package stopwords is already up-to-date!

In [143]: from nltk.corpus import stopwords
          stopwords = set(stopwords.words("english"))

In [144]: # import stemming
          stemmer = nltk.SnowballStemmer("english")
```

Data Cleaning

```
In [179]: def clean_data(text):
              text = str(text).lower()
              text = re.sub('\[.*?\]', '', text)
              text = re.sub('https?://\S+|www\.\S+', '', text)
              text = re.sub('<.*?>+', '', text)
              text = re.sub('\w*\d\w*','', text)
              text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
              text = re.sub('\n', '', text)
              text = re.sub('\w*\d\w*','', text)
              text = [word for word in text.split(' ') if word not in stopwords] #removing stopwords
              text = " ".join(text)
              text = [stemmer.stem(word) for word in text.split(' ')]
              text = " ".join(text)
              return text
```

```
In [180]: data["tweet"] = data["tweet"].apply(clean_data)
```

```
C:\Users\ANIlAppData\Local\Temp\ipykernel_9900\1823165696.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data["tweet"] = data["tweet"].apply(clean_data)
```

```
In [181]: data
```

Out[181]:

	tweet	Labels
0	rt mayasolov woman shouldnt complain clean ho...	No hate or Offensive Language
1	rt boy dat coldtyga dwn bad cuffin dat hoe ...	Offensive Language
2	rt ukkindofbrand dawg rt ever fuck bitch sta...	Offensive Language
3	rt cganderson vivabas look like tranni	Offensive Language
4	rt shenikarobert shit hear might true might f...	Offensive Language
...	...	...
24778	yous muthafin lie coreyemanuel right tl tras...	Offensive Language
24779	you gone broke wrong heart baabi drove redneck...	No hate or Offensive Language
24780	young buck wanna eat dat nigguh like aint fuck...	Offensive Language
24781	you got wild bitch tellin lie	Offensive Language
24782	ruffl ntac eileen dahlia beauti color combin...	No hate or Offensive Language

24783 rows × 2 columns

```
In [182]: x = np.array(data["tweet"])
          y = np.array(data["Labels"])
```

```
In [183]: x
```

Out[183]:

```
array(['  rt mayasolov woman shouldnt complain clean hous amp man alway take trash',
       '  rt  boy dat coldtyga dwn bad cuffin dat hoe  place',
       '  rt urkindofbrand dawg rt  ever fuck bitch start cri confus shit',
       ...,
       'young buck wanna eat dat nigguh like aint fuckin dat',
       'you got wild bitch tellin lie',
       'ruffl ntac eileen dahlia  beauti color combin pink orang yellow amp white coll '],
      dtype='<U34')
```

FEATURING

```
In [184]: from sklearn.feature_extraction.text import CountVectorizer
          from sklearn.model_selection import train_test_split

In [185]: cv = CountVectorizer()
          x = cv.fit_transform(x)

In [186]: x
```

Out[186]:

```
<24783x39781 sparse matrix of type '<class 'numpy.int64'>'
        with 197851 stored elements in Compressed Sparse Row format>
```

```
In [187]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,random_state = 42)

In [188]: x_train
```

Out[188]:

```
<16604x39781 sparse matrix of type '<class 'numpy.int64'>'
        with 132628 stored elements in Compressed Sparse Row format>
```

Building ML model

```
In [189]: from sklearn.tree import DecisionTreeClassifier

In [190]: dt = DecisionTreeClassifier()

In [191]: dt.fit(x_train, y_train)

In [192]: DecisionTreeClassifier()

In [193]: y_pred = dt.predict(x_test)

In [194]: #confusion matrix
          from sklearn.metrics import confusion_matrix
          cm = confusion_matrix(y_test, y_pred)
          cm
```

Out[194]:

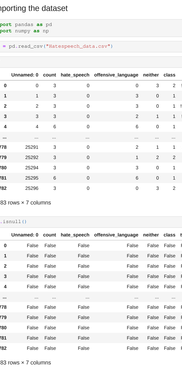
```
array([[ 153,   41,  271],
       [  33, 1156,  190],
       [ 235,  243, 5857]], dtype=int64)
```

```
In [195]: import seaborn as sns
          import matplotlib.pyplot as plt
          %matplotlib inline

In [196]: sns.heatmap(cm, annot = True, fmt = "d", cmap="YlGnBu" )
```

Out[196]:

```
<AxesSubplot:>
```




```
In [197]: from sklearn.metrics import accuracy_score
          accuracy_score(y_test, y_pred)
```

Out[197]:

```
0.8761462814525
```

Model Sample

```
In [198]: sample = "let's unite and kill all the people who are protesting against the government"
          sample = clean_data(sample)

In [199]: sample

Out[199]: 'let unit kill peopl protest govers'

In [200]: data1 = cv.transform([sample]).toarray()

In [201]: data1

Out[201]: array([[0, 0, 0, ..., 0, 0, 0]], dtype=int64)

In [202]: dt.predict(data1)

Out[202]: array(['Hate Speech'], dtype=object)
```

Conclusion

The project successfully developed a model that can classify text as hate speech or non-hate speech with reasonable accuracy. During evaluation, the model demonstrated an accuracy of around 87%, with precision and recall scores reflecting its ability to effectively differentiate between hateful and non-hateful content. The confusion matrix highlighted that while most instances were correctly classified, a small percentage of false positives and false negatives were observed. This indicates that, although the model performs well overall, it may need further tuning or a larger dataset to handle edge cases more accurately. The results suggest that the model can be a useful tool for moderating online content, but there is still room for improvement, particularly in reducing misclassifications and enhancing robustness against diverse types of hate speech.