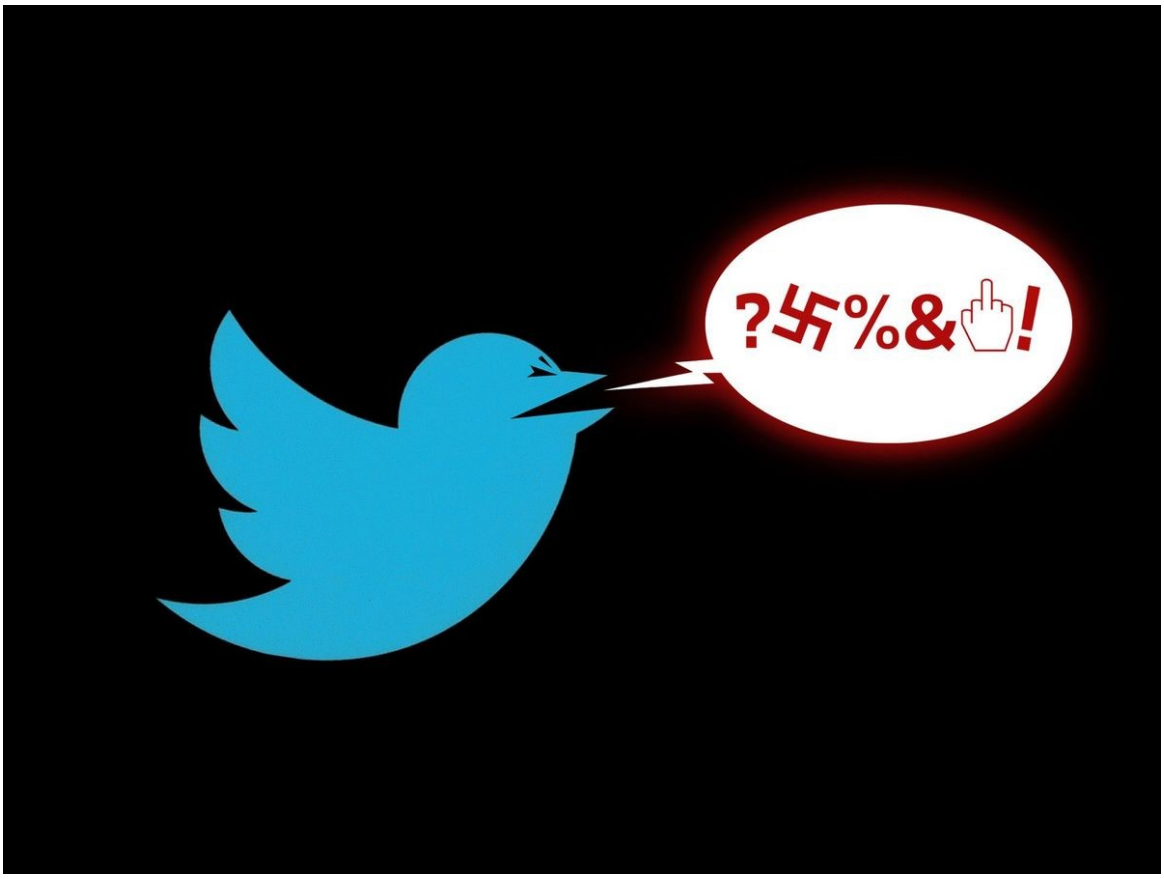
Open in Colab

Combatting Hate Speech in Tweeets



Introduction

As social media becomes more and more ingrained in our daily lives, it's easy to relate each platform to our physical environments. LinkedIn, for example, is the office: a professional landscape we use for networking, sharing news, and connecting with coworkers. Facebook is your living room, where you catch up with friends (but still guard your privacy).

And Twitter? Well, Twitter is the bar scene, where people let loose and talk to strangers, drop one-liners (or pick-up lines), and engage with personalities from all walks of life.

It is this bar-like atmosphere that makes Twitter the ultimate platform for customer engagement, and for the same reason why Twitter is the ideal social network for marketers:

Twitter is the only social network where brands and consumers have an even playing field and unrestricted lines of clear, concise communication.

Twitter

Twitter is an online news and social networking site where people communicate in short messages called tweets. Tweeting is posting short messages for anyone who follows you on Twitter, with the hope that your words are useful and interesting to someone in your

audience. Another description of Twitter and tweeting might be microblogging.

Why Twitter Is So Popular

Twitter's big appeal is how scan-friendly it is. You can track hundreds of engaging Twitter users and read their content with a glance, which is ideal for our modern attention-deficit world.
Twitter employs a purposeful message size restriction to keep things scan-friendly: every microblog tweet entry is limited to 280 characters or less. This size cap promotes the focused and clever use of language, which makes tweets easy to scan, and challenging to write. This size restriction made Twitter a popular social tool.

How Twitter Works

Twitter is easy to use as either broadcaster or a receiver. You join with a free account and Twitter name. Then you send broadcasts (tweets) daily, hourly, or as frequently as you like. Go to the What's Happening box next to your profile image, type 280 or fewer characters, and click Tweet. People who follow you, and potentially others who don't, will see your tweet. Encourage people you know to follow you and receive your tweets in their Twitter feeds. Let your friends know you are on Twitter to build up a following slowly. When people follow you, Twitter etiquette calls for you to follow them back.
To receive Twitter feeds, find someone interesting (celebrities included) and press Follow to subscribe to their tweets. If their tweets aren't as interesting as you hoped, you can always unfollow them. Go to your account at Twitter.com day or night to read your Twitter feed, which is continually changing as people post. Check out Trending topics to see what's going on in the world

Why People Tweet

People send tweets for all sorts of reasons besides sharing their thoughts: vanity, attention, shameless self-promotion of their web pages, or pure boredom. The vast majority of tweeters microblog recreationally. It's a chance to shout out to the world and revel in how many people read their tweets.
However, a growing number of Twitter users send out useful content, and that's the real value of Twitter. It provides a stream of quick updates from friends, family, scholars, news journalists, and experts. It empowers people to become amateur journalists of life, describing and sharing something that they found interesting about their day.
Twitter has a lot of drivel, but at the same time, there is a base of useful news and knowledgeable content. You'll need to decide for yourself which content is worth following there.

Hate Speech

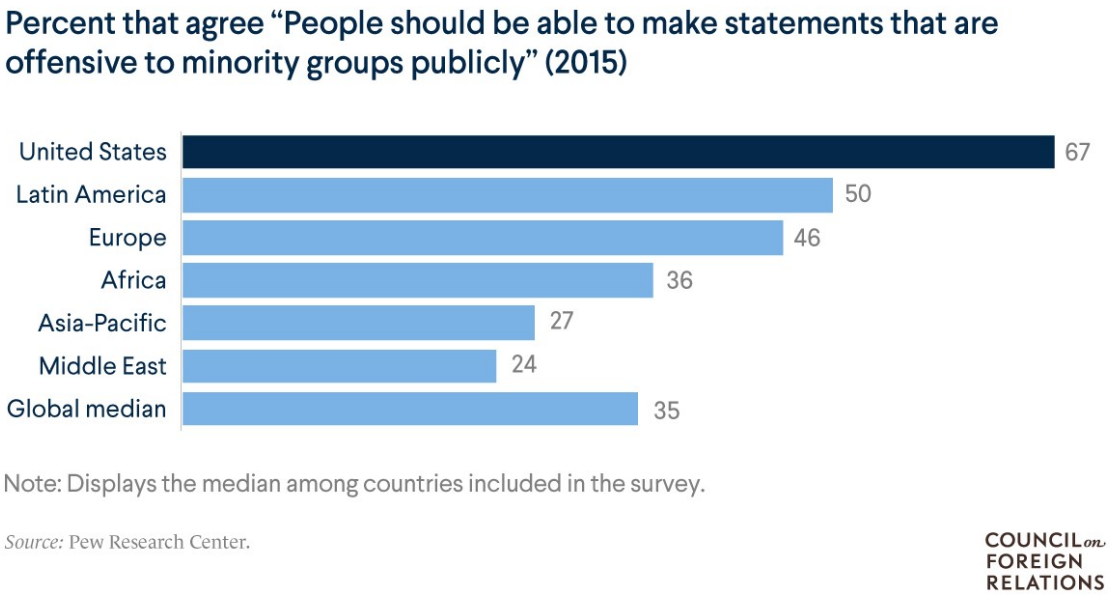
Violence attributed to online hate speech has increased worldwide. Societies confronting the trend must deal with questions of free speech and censorship on widely used tech platforms. Hate speech online has been linked to a global increase in violence toward minorities, including mass shootings, lynchings, and ethnic cleansing. This gives rise to policies used to

curb hate speech which risk limiting free speech and are inconsistently enforced. Countries such as the United States grant social media companies broad powers in managing their content and enforcing hate speech rules. Others, including Germany, can force companies to remove posts within certain time periods.

A mounting number of attacks on immigrants and other minorities has raised new concerns about the connection between inflammatory speech online and violent acts, as well as the role of corporations and the state in policing speech. Analysts say trends in hate crimes around the world echo changes in the political climate, and that social media can magnify discord. At their most extreme, rumors and invective disseminated online have contributed to violence ranging from lynchings to ethnic cleansing.
The response has been uneven, and the task of deciding what to censor, and how, has largely fallen to the handful of corporations that control the platforms on which much of the world now communicates. But these companies are constrained by domestic laws. In liberal democracies, these laws can serve to defuse discrimination and head off violence against minorities. But such laws can also be used to suppress minorities and dissidents.

How widespread is the problem?

Incidents have been reported on nearly every continent. Much of the world now communicates on social media, with nearly a third of the world's population active on Facebook alone. As more and more people have moved online, experts say, individuals inclined toward racism, misogyny, or homophobia have found niches that can reinforce their views and goad them to violence. Social media platforms also offer violent actors the opportunity to publicize their acts.



Social scientists and others have observed how social media posts, and other online speech, can inspire acts of violence:

- In Germany a correlation was found between anti-refugee Facebook posts by the far-right Alternative for Germany party and attacks on refugees. Scholars Karsten Muller and

Carlo Schwarz observed that upticks in attacks, such as arson and assault, followed spikes in hate-mongering posts.

- In the United States, perpetrators of recent white supremacist attacks have circulated among racist communities online, and also embraced social media to publicize their acts. Prosecutors said the Charleston church shooter, who killed nine black clergy and worshippers in June 2015, engaged in a "self-learning process" online that led him to believe that the goal of white supremacy required violent action.
- The 2018 Pittsburgh synagogue shooter was a participant in the social media network Gab, whose lax rules have attracted extremists banned by larger platforms. There, he espoused the conspiracy that Jews sought to bring immigrants into the United States, and render whites a minority, before killing eleven worshippers at a refugee-themed Shabbat service. This "great replacement" trope, which was heard at the white supremacist rally in Charlottesville, Virginia, a year prior and originates with the French far right, expresses demographic anxieties about nonwhite immigration and birth rates.
- The great replacement trope was in turn espoused by the perpetrator of the 2019 New Zealand mosque shootings, who killed forty-nine Muslims at prayer and sought to broadcast the attack on YouTube.
- In Myanmar, military leaders and Buddhist nationalists used social media to slur and demonize the Rohingya Muslim minority ahead of and during a campaign of ethnic cleansing. Though Rohingya comprised perhaps 2 percent of the population, ethnonationalists claimed that Rohingya would soon supplant the Buddhist majority. The UN fact-finding mission said, "Facebook has been a useful instrument for those seeking to spread hate, in a context where, for most users, Facebook is the Internet."
- In India, lynch mobs and other types of communal violence, in many cases originating with rumors on WhatsApp groups, have been on the rise since the Hindu-nationalist Bharatiya Janata Party (BJP) came to power in 2014.
- Sri Lanka has similarly seen vigilantism inspired by rumors spread online, targeting the Tamil Muslim minority. During a spate of violence in March 2018, the government blocked access to Facebook and WhatsApp, as well as the messaging app Viber, for a

Target of the Project work

It is no doubt that social media platforms are a big forefront of todays globlized world, and also we cannot deny the fact that racial discrimination and hate attacks is as unfortunate as is disastrous for the peace of society and for a platform, thus their mitigation is a very important and pressing issue for these social media platforms like Twitter, FaceBook, Whatsapp, WeChat, etc.

In this project we will be trying to detect and remove hate-speech in the twitter front using NLP(Natural Language Processing). Here we will use a set of Machine Learning Algorithms along with NLP algorithms to detect a tweet and understand if it a **hate tweet** or not and thereby try to remove if it involves or likely to spread any kind of hatred

Besides this on the surface level we need to follow through the following points of work flow for the project as given below

1. Load the dataset
2. Clear the dataset
3. Explore the dataset
4. Create the visualization
5. Train on all the Models

Natural Language Processing

What Is Natural Language Processing (NLP)?

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI). It helps machines process and understand the human language so that they can automatically perform repetitive tasks. Examples include machine translation, summarization, ticket classification, and spell check.

Why Is Natural Language Processing Important?

One of the main reasons natural language processing is so critical to businesses is that it can be used to analyze large volumes of text data, like social media comments, customer support tickets, online reviews, news reports, and more.

All this business data contains a wealth of valuable insights, and NLP can quickly help businesses discover what those insights are. It does this by helping machines make sense of human language in a faster, more accurate, and more consistent way than human agents. NLP tools process data in real time, 24/7, and apply the same criteria to all your data, so you can ensure the results you receive are accurate – and not riddled with inconsistencies.

Once NLP tools can understand what a piece of text is about, and even measure things like sentiment, businesses can start to prioritize and organize their data in a way that suits their needs.

Challenges of NLP

While there are many challenges in natural language processing, the benefits of NLP for businesses are huge making NLP a worthwhile investment. However, it's important to know what those challenges are before getting started with NLP. Human language is complex, ambiguous, disorganized, and diverse. There are more than 6,500 languages in the world, all of them with their own syntactic and semantic rules.Even humans struggle to make sense of language.

So for machines to understand natural language, it first needs to be transformed into something that they can interpret. In NLP, syntax and semantic analysis are key to understanding the grammatical structure of a text and identifying how words relate to each other in a given context. But, transforming text into something machines can process is complicated.

Data scientists need to teach NLP tools to look beyond definitions and word order, to understand context, word ambiguities, and other complex concepts connected to human language.

How Does Natural Language Processing Work?

In natural language processing, human language is separated into fragments so that the grammatical structure of sentences and the meaning of words can be analyzed and understood in context. This helps computers read and understand spoken or written text in the same way as humans.
Here are a few fundamental NLP pre-processing tasks data scientists need to perform before NLP tools can make sense of human language:

- Tokenization: breaks down text into smaller semantic units or single clauses Part-of-speech-tagging: marking up words as nouns, verbs, adjectives, adverbs, pronouns, etc
- Stemming and lemmatization: standardizing words by reducing them to their root forms
- Stop word removal: filtering out common words that add little or no unique information, for example, prepositions and articles (at, to, a, the).

Only then can NLP tools transform text into something a machine can understand.

Library Imports

We will now import the libraries required for carrying out the filtering of Hatred.

In [ ]:

```
# imports
import nltk
import numpy as np
import pandas as pd
import  matplotlib.pyplot as plt
import seaborn as sns
import warnings as warns
import time
from nltk.corpus import stopwords
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
import re
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
from wordcloud import WordCloud, STOPWORDS
from sklearn.metrics import accuracy_score, classification_report
```

Primaries

In [ ]:

```
nltk.download('stopwords')
nltk.download('punkt')
pd.set_option('display.max_colwidth', -1)
warns.filterwarnings("ignore")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarni
ng: Passing a negative integer is deprecated in version 1.0 and will not be
supported in future version. Instead, use None to not limit the column widt
h.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

Data Acquisition

The data for this project is obtained from the twitter hate speech curated exmples dataset as is provided by Rahul Agarwal(user id: vkrahul) in Kaggle.com the link of data is as given below.

> https://www.kaggle.com/vkrahul/twitter-hate-speech

The set of data provided has two datasets

1. train_E6oV3lV.csv : for training the machine learning models link
2. test_tweets_anuFYb8.csv : for testing the trained models link

For easy access as per this project we will be accessing the same data as reuploaded in Github exclusively for use in this project. https://github.com/WolfDev8675/RepoSJX7/tree/Assign5_2/Data

In [ ]:
```
# import data
trainSet=pd.read_csv("https://github.com/WolfDev8675/RepoSJX7/raw/Assign5_2
testSet=pd.read_csv("https://raw.githubusercontent.com/WolfDev8675/RepoSJX7
```

Data Display

Training Set

In [ ]:
```
trainSet.head()
```

Out[ ]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
1	2	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in urð±!!! ðððð¦ð¦ð¦
4	5	0	factsguide: society now #motivation

Testing Set

In [ ]:
```
testSet.head()
```

Out[ ]:

	id	tweet
0	31963	#studiolife #aislife #requires #passion #dedication #willpower to find #newmaterialsâ¦
1	31964	@user #white #supremacists want everyone to see the new â #birdsâ #movie â and hereâs why
2	31965	safe ways to heal your #acne!! #altwaystoheal #healthy #healing!!
3	31966	is the hp and the cursed child book up for reservations already? if yes, where? if no, when? ðð #harrypotter #pottermore #favorite
4	31967	3rd #bihday to my amazing, hilarious #nephew eli ahmir! uncle dave loves you and missesâ¦

Data Exploration

Datatype Information

Training Set

In [ ]:
```
trainSet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31962 entries, 0 to 31961
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      31962 non-null  int64 
 1   label   31962 non-null  int64 
 2   tweet   31962 non-null  object
```

```
dtypes: int64(2), object(1)
memory usage: 749.2+ KB
```

Testing Set

In [ ]:
```
testSet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17197 entries, 0 to 17196
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      17197 non-null  int64 
 1   tweet   17197 non-null  object
dtypes: int64(1), object(1)
memory usage: 268.8+ KB
```

Size of datasets

Training Set

In [ ]:
```
#Train size
print("Size of the training dataset : ", trainSet.shape)
```

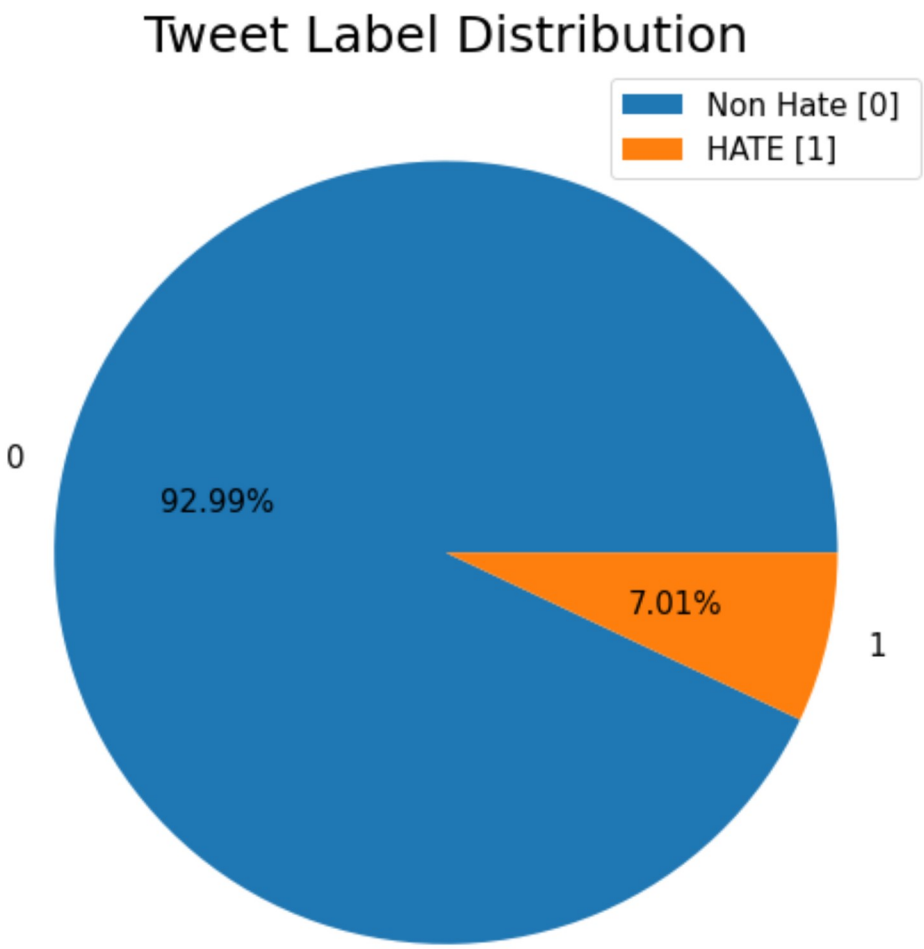
```
Size of the training dataset :  (31962, 3)
```

Testing Set

In [ ]:
```
# unlabeled test size
print("Size of the testing dataset : ", testSet.shape)
```

```
Size of the testing dataset :  (17197, 2)
```

In [ ]:
```
# Tweet label distribution
percs=trainSet.label.value_counts()*100/len(trainSet.label)
idfs=trainSet.label.value_counts().index.values
fig1=plt.figure(figsize=(9,9));ax1=fig1.add_subplot(111)
ax1.pie(percs,labels=idfs,autopct='%1.2f%%', textprops={'fontsize': 15});
ax1.legend(['Non Hate [0] ','HATE [1] '],fontsize=15);
plt.title("Tweet Label Distribution",fontdict = {'fontsize' : 25})
plt.show()
```



So we see that a very low quantity of examples are there (approx: 7%) of the 31962 tweets that can be termed as Hate Tweets or approximately 2237 tweets convey hatred. Thus we have fewer examples for our learning algorithms to learn.

Word Clouds

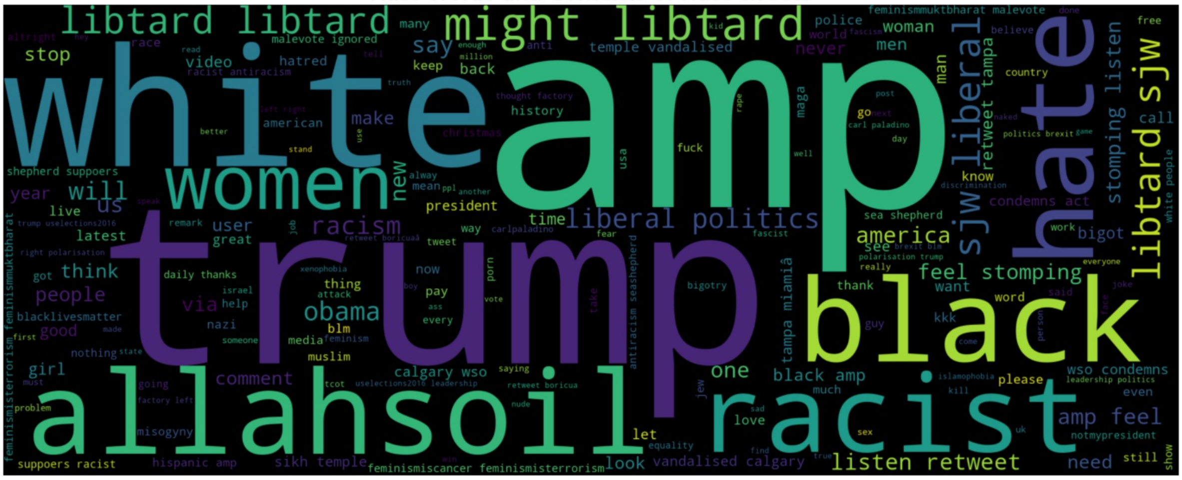
Hatred Set

In [ ]:

```
# Hate speech specific word cloud
temp_df = trainSet[trainSet.label==1]
words = " ".join(temp_df.tweet)
tweeted_words = " ".join([w for w in words.split()
                                if 'http' not in w
                                  and not w.startswith('@')
                                  and w!='RT'])

wrdcld = WordCloud(stopwords=STOPWORDS,
                   background_color='black',
                   width=2500,
                   height=1000).generate(tweeted_words)
fig2=plt.figure(figsize=(25,10));ax2=fig2.add_subplot(111);
plt.imshow(wrdcld)
plt.axis('off')
plt.title(" Hate Words and Contents of Hate tweets ",fontdict = {'fontsize'
plt.show()
```

Hate Words and Contents of Hate tweets



Non Hatred Set

In [ ]:

```
# Non-Hate speech specific word cloud
temp_df = trainSet[trainSet.label==0]
words = " ".join(temp_df.tweet)
tweeted_words = " ".join([w for w in words.split()
                              if 'http' not in w
                                and not w.startswith('@')
                                and w!='RT'])

wrdcld = WordCloud(stopwords=STOPWORDS,
                   background_color='black',
                   width=2500,
                   height=1000).generate(tweeted_words)
fig3=plt.figure(figsize=(25,10));ax3=fig3.add_subplot(111);
plt.imshow(wrdcld)
plt.axis('off')
plt.title(" Contents of Non-Hate tweets ",fontdict = {'fontsize' : 25})
plt.show()
```

Contents of Non-Hate tweets

Data Preparation

Preset

In [ ]:

```
stops_EN=set(stopwords.words('english'))
```

Training Set

In [ ]:
```
trainSet.insert(2,"Cleaned",range(trainSet.shape[0]))
for idx in trainSet.index:
  tweet=trainSet.tweet[idx]
  alph_0=re.sub("[^a-zA-Z]", " ",tweet)
  words=alph_0.lower().split();
  cleaned=[word for word in words if word not in stops_EN]
  trainSet['Cleaned'][idx]=' '.join(cleaned)
```

Result of Cleansing

In [ ]:
```
trainSet.head()
```

Out[ ]:

	id	label	Cleaned	tweet
0	1	0	user father dysfunctional selfish drags kids dysfunction run	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
1	2	0	user user thanks lyft credit use cause offer wheelchair vans pdx disapointed getthanked	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
2	3	0	bihday majesty	bihday your majesty
3	4	0	model love u take u time ur	#model i love u take with u all the time in urð±!!! ððððð¦ð¦ð¦
4	5	0	factsguide society motivation	factsguide: society now #motivation

Testing Set

In [ ]:
```
testSet.insert(2,"Cleaned",range(testSet.shape[0]))
for idx in testSet.index:
  tweet=testSet.tweet[idx]
  alph_0=re.sub("[^a-zA-Z]", " ",tweet)
  words=alph_0.lower().split();
  cleaned=[word for word in words if word not in stops_EN]
  testSet['Cleaned'][idx]=' '.join(cleaned)
```

Result of Cleansing

In [ ]:
```
testSet.head()
```

Out[ ]:

	id	tweet	Cleaned
0	31963	#studiolife #aislife #requires #passion #dedication #willpower to find #newmaterialsâ¦	studiolife aislife requires passion dedication willpower find newmaterials
1	31964	@user #white #supremacists want everyone to see the new â #birdsâ #movie â and hereâs why	user white supremacists want everyone see new birds movie
2	31965	safe ways to heal your #acne!! #altwaystoheal #healthy #healing!!	safe ways heal acne altwaystoheal healthy healing

	id	tweet	Cleaned
3	31966	is the hp and the cursed child book up for reservations already? if yes, where? if no, when? ðððð #harrypotter #pottermore #favorite	hp cursed child book reservations already yes harrypotter pottermore favorite

Model Creation

Trainer & Tester Datasets

Splitter

In [ ]:
```
# dataset splitting for "trainSet"
trainer_X,tester_X,trainer_Y,tester_Y=train_test_split(trainSet.Cleaned, tr
```

Vectorization

In [ ]:
```
# Vectorizer object
vector=CountVectorizer(analyzer = "word",tokenizer = None,preprocessor = Nc
```

Trainers

In [ ]:
```
# vectorizing trainers
trainerFeatures=vector.fit_transform(trainer_X)
trainerFeatures=trainerFeatures.toarray()
```

Testers

In [ ]:
```
# vectorizing testers
testerFeatures=vector.fit_transform(tester_X)
testerFeatures=testerFeatures.toarray()
```

Classification Models

Our task in this project is to identify ***A Hate Tweet*** from ***A Non-Hate Tweet***, hence we need to classify one tweet from another or in other words we have a problem of classifying tweets into either of the two groups. Classification can be performed on structured or unstructured data. Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under. As per the terms of Classification there are many algorithms but for logicality and feasibility of this project work we will employ ‘‘7 ” of the most common Classification Algorithms.

1. *Logistic Regression*
   Definition: Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.
   Advantages: Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.

Disadvantages: Works only when the predicted variable is binary, assumes all predictors are independent of each other and assumes data is free of missing values.

2. *Naïve Bayes*
   Definition: Naive Bayes algorithm based on Bayes' theorem with the assumption of independence between every pair of features. Naive Bayes classifiers work well in many real-world situations such as document classification and spam filtering.
   Advantages: This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.
   Disadvantages: Naive Bayes is is known to be a bad estimator.

3. *Stochastic Gradient Descent*
   Definition: Stochastic gradient descent is a simple and very efficient approach to fit linear models. It is particularly useful when the number of samples is very large. It supports different loss functions and penalties for classification.
   Advantages: Efficiency and ease of implementation.
   Disadvantages: Requires a number of hyper-parameters and it is sensitive to feature scaling.

4. *K-Nearest Neighbours*
   Definition: Neighbours based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbours of each point.
   Advantages: This algorithm is simple to implement, robust to noisy training data, and effective if training data is large.
   Disadvantages: Need to determine the value of K and the computation cost is high as it needs to compute the distance of each instance to all the training samples.

5. *Decision Tree*
   Definition: Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.
   Advantages: Decision Tree is simple to understand and visualise, requires little data preparation, and can handle both numerical and categorical data.
   Disadvantages: Decision tree can create complex trees that do not generalise well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

6. *Random Forest*
   Definition: Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.
   Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.
   Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

7. *Support Vector Machine*

   Definition: Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

   Advantages: Effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient.

   Disadvantages: The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

Model Class Imports (Multiple Models)

In [ ]:

```
from sklearn.linear_model import LogisticRegression as LGR
from sklearn.naive_bayes import MultinomialNB as NBS
from sklearn.linear_model import SGDClassifier as SGD
from sklearn.neighbors import KNeighborsClassifier as KNN
from sklearn.tree import DecisionTreeClassifier as DTC
from sklearn.ensemble import RandomForestClassifier as RFC
from sklearn.svm import SVC
```

Model Creation (Multiple Models)

In [ ]:

```
#Model Creation (Multiple models)
model_list={'LGR':'Logistic Regression',
            'NBS':'Naïve Bayes',
            'SGD':'Stochastic Gradient Descent',
            'KNN':'K-Nearest Neighbours',
            'DTC':'Decision Tree',
            'RFC':'Random Forest',
            'SVM':'Support Vector Machine'}

print('Creating models for : \n')
for key in model_list: print('\t',model_list[key].ljust(30,' '),' : ',key,s
print('_'.center(80,'_'));
models={'LGR':LGR(solver='lbfgs', max_iter=10000),
        'NBS':NBS(),
        'SGD':SGD(),
        'KNN':KNN(n_neighbors = 5),
        'DTC':DTC(),
        'RFC':RFC(n_estimators=200),
        'SVM':SVC(kernel='linear',C=1.0)};
```

```
Creating models for :

         Logistic Regression              :   LGR
         Naïve Bayes                      :   NBS
         Stochastic Gradient Descent      :   SGD
         K-Nearest Neighbours             :   KNN
         Decision Tree                    :   DTC
         Random Forest                    :   RFC
         Support Vector Machine           :   SVM
________________________________________________________________________________
_____
```

Model Training (Multiple Models)

In [ ]:
```
#Training Data on Models (Multiple models)
print(' Model Training Status : ')
for key in models:
    tic=time.perf_counter();
    models[key].fit(trainerFeatures,trainer_Y)
    toc=time.perf_counter()
    print('\tTrained Model : ',key,f"\t time taken: {toc - tic:0.4f} seconc
print('_'.center(80,'_'));
```

```
Model Training Status :
        Trained Model :  LGR     time taken: 25.5523 seconds
        Trained Model :  NBS     time taken: 3.8908 seconds
        Trained Model :  SGD     time taken: 5.0862 seconds
        Trained Model :  KNN     time taken: 31.2326 seconds
        Trained Model :  DTC     time taken: 576.5010 seconds
        Trained Model :  RFC     time taken: 738.8697 seconds
        Trained Model :  SVM     time taken: 2112.6999 seconds
```

Model Testing (Multiple Models)

Generating Predictions for Tests

In [ ]:
```
#Prediction tests from the Trained Models
predicts=dict.fromkeys(models.keys())
print(' Prediction Generation Status : ')
for key in models:
    tic=time.perf_counter()
    predicts[key]=models[key].predict(testerFeatures)
    toc=time.perf_counter()
    print('Prediction generated for : ',key,f"\t time taken: {toc - tic:0.4
print('_'.center(80,'_'));
```

```
 Prediction Generation Status :
Prediction generated for :  LGR          time taken: 0.3699 seconds
Prediction generated for :  NBS          time taken: 0.1623 seconds
Prediction generated for :  SGD          time taken: 0.1091 seconds
Prediction generated for :  KNN          time taken: 1275.4819 seconds
Prediction generated for :  DTC          time taken: 0.1376 seconds
Prediction generated for :  RFC          time taken: 6.1561 seconds
Prediction generated for :  SVM          time taken: 183.5973 seconds
```

Studying Accuracy and Reports of the Model

In [ ]:
```
accr=dict.fromkeys(models.keys())
for key in models:
    accr[key]=100*accuracy_score(predicts[key],tester_Y)
    print('Metrics for : ',model_list[key])
    print('Accuracy : {:10.4f}%'.format(accr[key]))
    print(classification_report(predicts[key],tester_Y))
    print('_'.center(80,'_'));print('\n\n');
print('*'.center(80,'-'));
```

```
Metrics for :  Logistic Regression
Accuracy :    91.7879%
              precision    recall  f1-score   support

           0       0.98      0.93      0.96      6295
           1       0.01      0.05      0.02        98
```

accuracy			0.92	6393
macro avg	0.50	0.49	0.49	6393
weighted avg	0.97	0.92	0.94	6393

Metrics for : Naïve Bayes
Accuracy : 72.4073%

	precision	recall	f1-score	support
0	0.75	0.94	0.84	4740
1	0.37	0.10	0.16	1653
accuracy			0.72	6393
macro avg	0.56	0.52	0.50	6393
weighted avg	0.65	0.72	0.66	6393

Metrics for : Stochastic Gradient Descent
Accuracy : 89.1287%

	precision	recall	f1-score	support
0	0.95	0.93	0.94	6083
1	0.06	0.08	0.07	310
accuracy			0.89	6393
macro avg	0.51	0.51	0.51	6393
weighted avg	0.91	0.89	0.90	6393

Metrics for : K-Nearest Neighbours
Accuracy : 93.1331%

	precision	recall	f1-score	support
0	1.00	0.93	0.96	6391
1	0.00	0.00	0.00	2
accuracy			0.93	6393
macro avg	0.50	0.47	0.48	6393
weighted avg	1.00	0.93	0.96	6393

Metrics for : Decision Tree
Accuracy : 83.1222%

	precision	recall	f1-score	support
0	0.88	0.94	0.91	5569
1	0.21	0.11	0.14	824
accuracy			0.83	6393
macro avg	0.54	0.52	0.53	6393
weighted avg	0.79	0.83	0.81	6393

```
_____


Metrics for :  Random Forest
Accuracy :    86.6260%
              precision    recall  f1-score   support

           0       0.92      0.93      0.93      5905
           1       0.08      0.07      0.08       488

    accuracy                           0.87      6393
   macro avg       0.50      0.50      0.50      6393
weighted avg       0.86      0.87      0.86      6393

_____


Metrics for :  Support Vector Machine
Accuracy :    86.6416%
              precision    recall  f1-score   support

           0       0.92      0.94      0.93      5864
           1       0.13      0.11      0.12       529

    accuracy                           0.87      6393
   macro avg       0.52      0.52      0.52      6393
weighted avg       0.86      0.87      0.86      6393

_____


----------------------------------------*
```

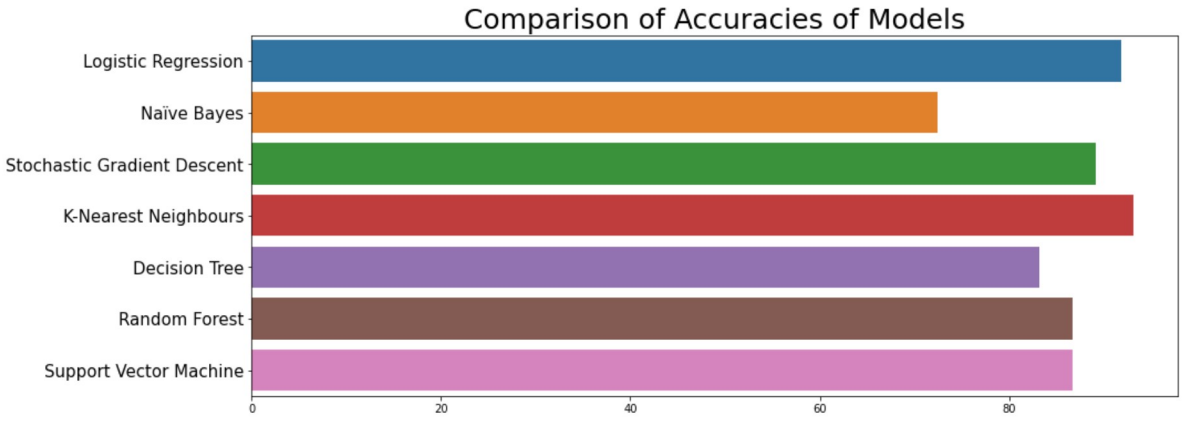
Comparative Study

As per the status we stand we have trained and obtained metics of 7 classification machine learning models for removing hate-speech. Now, we need to make a comparison on which model is more feasible to handle the problem of hate-speech

Accuracy Comparisons

In [ ]:

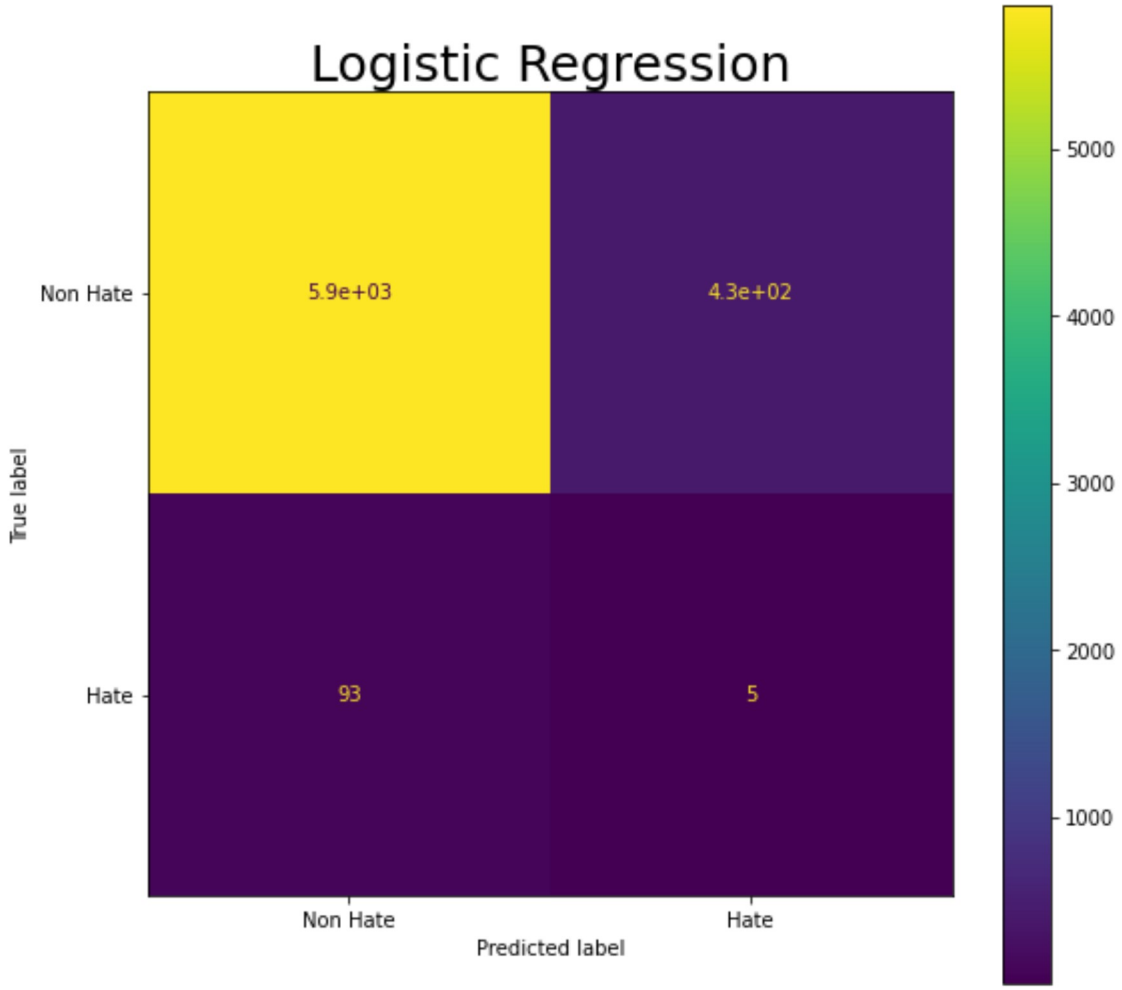
```
#Comparison of Accuracies of Models
fig4=plt.figure(figsize=(15,6));ax4=fig4.add_subplot(111)
sns.barplot(list(accr.values()),list(range(len(accr))),orient='h');
plt.yticks(range(len(accr)), list(model_list.values()),rotation=0,ha='right
plt.title(" Comparison of Accuracies of Models ",fontdict = {'fontsize' : 2
plt.show()
```

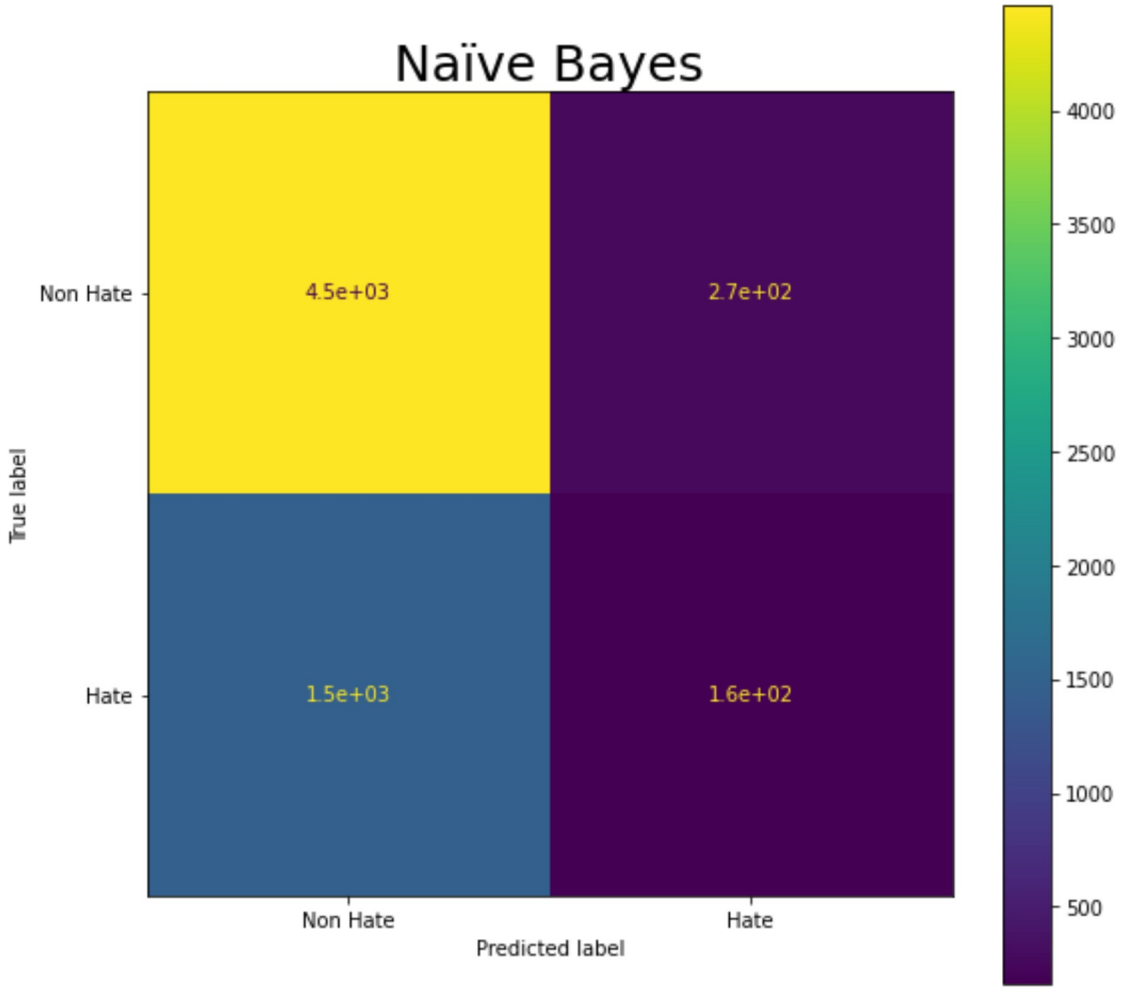


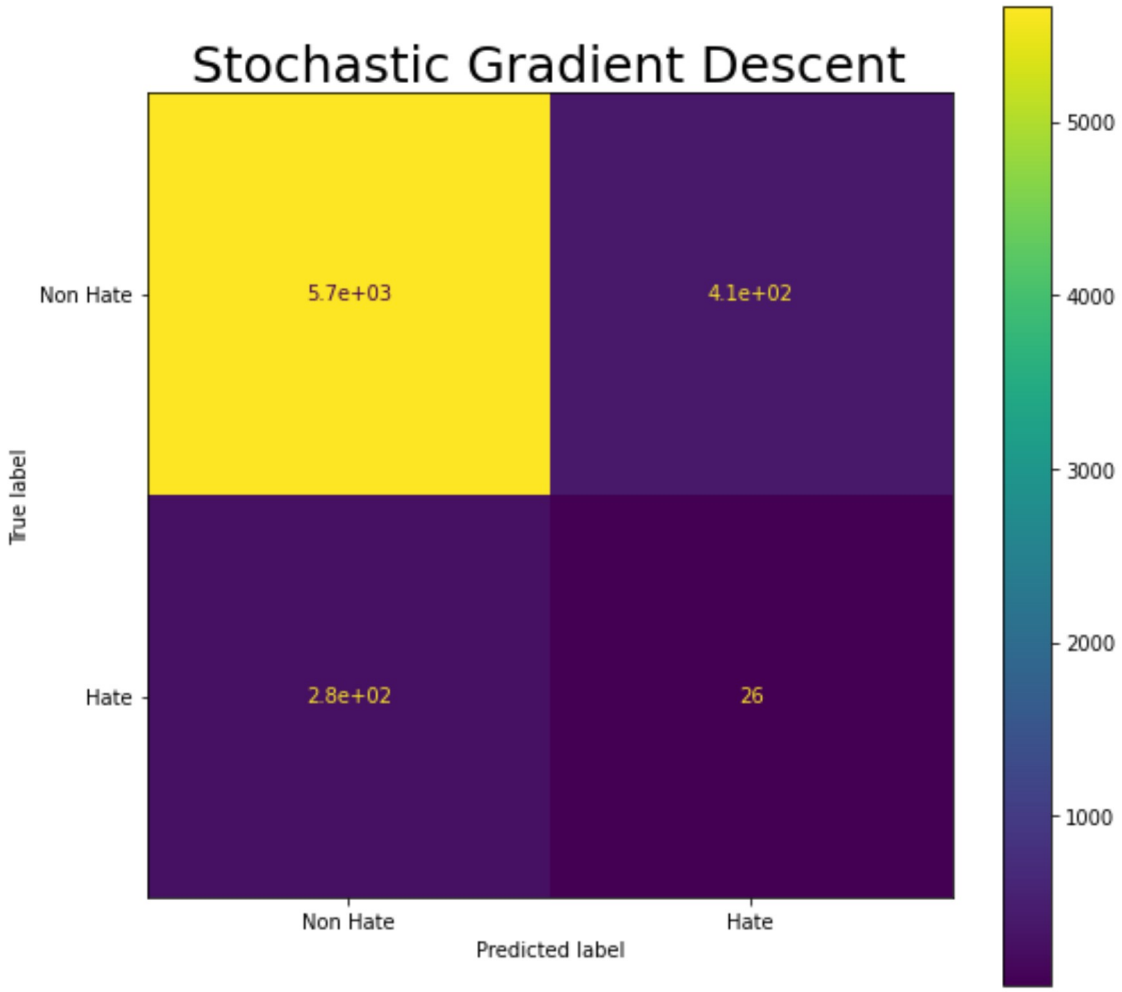
Confusion Matrices

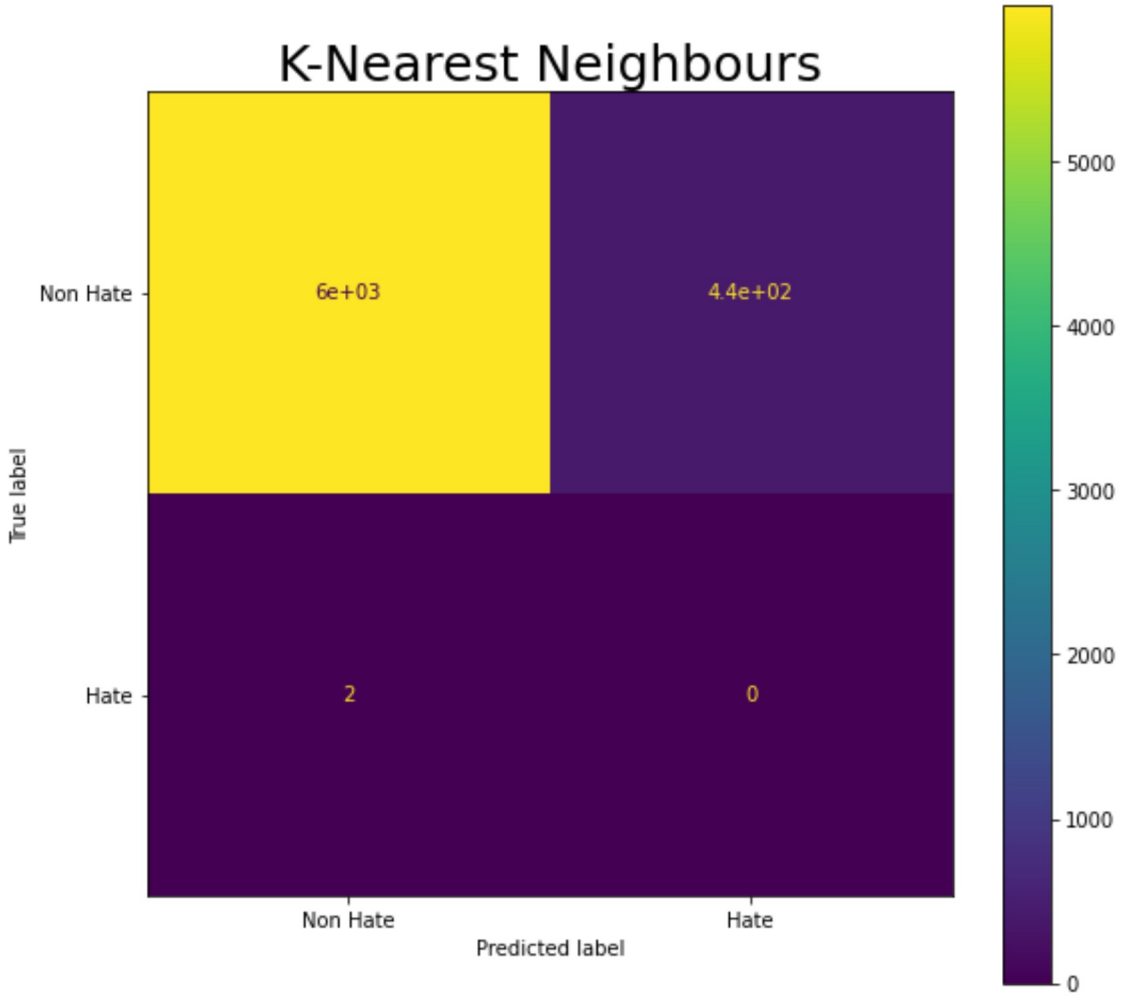
In [ ]:

```
# Confusion Matrices for the models
for key in models:
  fig5=plt.figure(figsize=(9,9));ax5=fig5.add_subplot(111);
  confMat=confusion_matrix(predicts[key],tester_Y)
  viewer=ConfusionMatrixDisplay(confMat,display_labels=['Non Hate','Hate'])
  viewer.plot(ax=ax5);
  plt.title(model_list[key],fontdict = {'fontsize' : 25});
  plt.show()
```

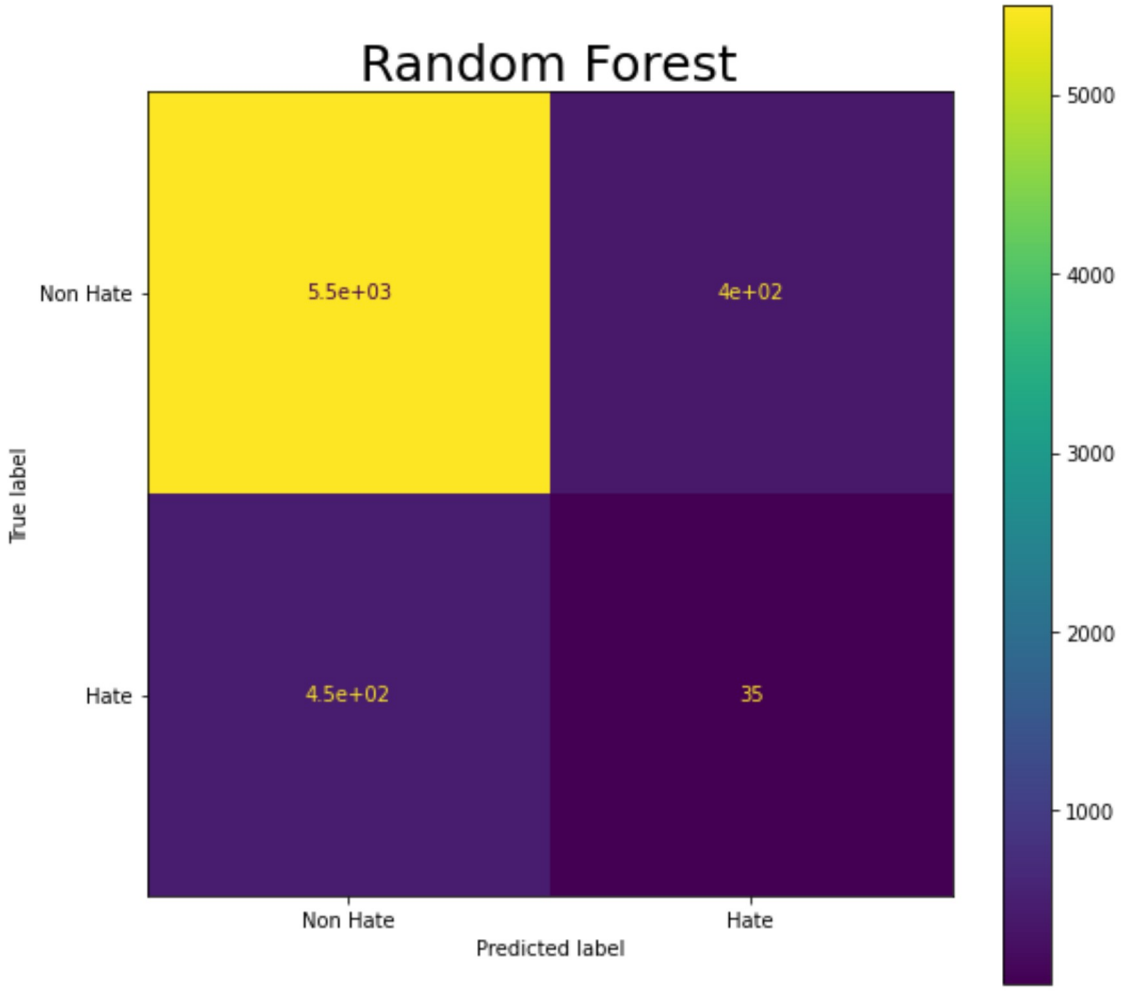


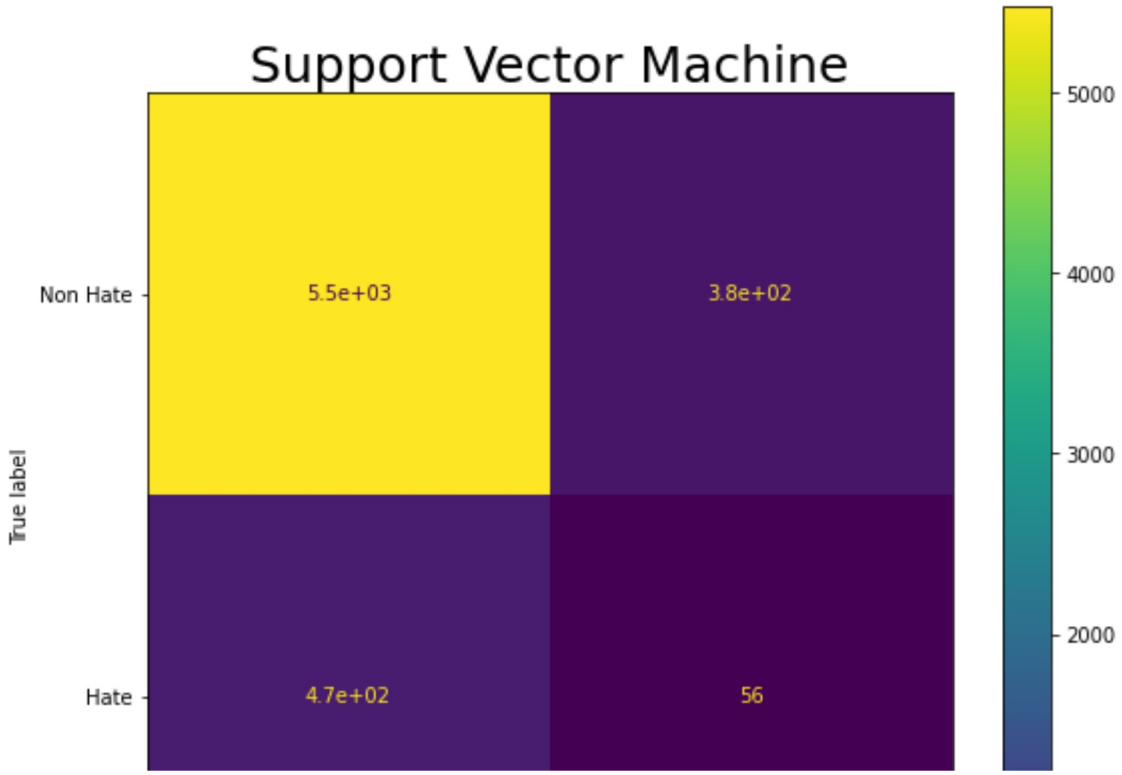

Naïve Bayes
Non Hate
Hate
True label
4.5e+03
2.7e+02
1.5e+03
1.6e+02
Non Hate
Hate
Predicted label
4000
3500
3000
2500
2000
1500
1000
500


Stochastic Gradient Descent
Non Hate
Hate
True label
5.7e+03
4.1e+02
2.8e+02
26
Non Hate
Hate
Predicted label
5000
4000
3000
2000
1000


K-Nearest Neighbours
Non Hate
Hate
True label
6e+03
4.4e+02
2
0
Non Hate
Hate
Predicted label
5000
4000
3000
2000
1000
0


Decision Tree
Non Hate
Hate
True label
5.2e+03
3.5e+02
7.3e+02
91
Non Hate
Hate
Predicted label
5000
4000
3000
2000
1000





as far as visible we have worked on, the models in descending order of their accuracies the top 3 are

1. K-Nearest Neighbour : accuracy at 93.1331%
2. Logistic Regression : accuracy at 91.7879%
3. Stocastic Gradient Descent : accuracy at 89.1287%

If we see carefully the KNN algorithm has many areas of performance better than the others at the cost of Precision and Recall metrics being quite one-sided, marking that there may be cases of Over-trained or an Under-trained model besides the time parameters are also to be noted in this situationAlso notable is the fact that the Multinominal Naïve Bayes and Support Vector Machines took majority of the time rather than others.

- Logistic Regression took 25.5523 seconds
- Naïve Bayes took 3.8908 seconds
- Stocastic Gradient descent took 5.0862 seconds
- K-Nearest Neighbour took 31.2326 seconds
- Decision Trees took 576.5010 seconds
- Random Forests took 738.8697 seconds
- Support Vector Machine took 2112.6999 seconds

Moreover if cases are researched we have a lower quantity of hate tweets for the machine to learn properly. Rather suggestive to this is that we may use Cross-Validation in later stages to refine this work.

With detailed scrutinization we might choose the Support Vector Machines as a better model inspite of the fact it provides lower accuracy than the KNN marked at a value of 86.6416%. The same also goes for the Random Forests. Inspite of this low accuracy the SVM model has precision and recall values better than the compared models. Now, for SVM models we might need hyperparameter boosting as a further measure to understand the hate-speech more than the other models. We also conclude about SVM from studies made from the Confusion matrices, showing a tendency to define hate speech more than the other models comparably. Thus showing a promise for better chances in the future if trained, but this thing also goes for all other models, viz., if given the scope, data, training to the proper limits and weightages of parameters and hyperparameters all models could be equally beter in classifying their forte in spaces the models are comfortable with or to say working with each model keeping in mind the cons of a model more than the pros. Till then our choice goes to SVM only if took a little less time to train than others.

On this note mention must be made of the data in the tweets which show that there are words which turn up in both the hate-tweets as well as non-hate ones that, as per a machine, learning that will obviously confuse it as we people with our brain an deeper understanding of language often fail to understand languages and the information they want to convey.

† $NOTE$: ***All information and metrics discussed above is subject to change on every execution. The study made here is purely on the basis of a complete run which took a time of 1 hour and 15 minutes for complete execution. The results so obtained from that execution cycle is used for study and the conclusions drawn. Any values spoken here may not be taken for face value if future executions.***

End Notes

After getting success in speech recognition and vision research, natural language processing is the most targeted research area in artificial intelligence.

Although it is started decades ago, most people lack the NLP experience. Because it's hard to teach a machine with the challenges listed below:

1. Sarcasm
2. Ambiguity (Syntactical and Lexical)
3. Syntax
4. Co-reference
5. Typos
6. Normalization
7. Puns

For a machine running on numbers it is behemoth of a task to make it understand which even normal people fail to cope up with sometimes owning to large variations languages among us, the sense in which it is spoken, tone of voice, etc., although for tweets a large amount of this problem is overcome due to the size of the tweets and scanability of tweets still a challege is a challenge and as long as people will be there, there will be communication, there will be sentiments involved in communication, there will difference of opininon which some may take for the face value others may make it a bigger issue, thus fight for a stand to express the opinion over others, and till the time there will be a rift in opinion there will be either indifference to some hatred for others.

As far as technology goes a platform will always try to supress hatred to maintain an friendly environment than going full on hostile if this hate is allowed to spread.

Thank You

Bibliography

- https://developer.twitter.com/en/community/success-stories/hatelab
- https://forward.com/news/352133/twitter-gives-online-hate-speech-its-biggest-platform-why/
- https://www.convinceandconvert.com/social-media-strategy/twitter-engagement/
- https://www.lifewire.com/what-exactly-is-twitter-2483331
- https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons
- https://monkeylearn.com/blog/what-is-natural-language-processing/
- https://analyticsindiamag.com/7-types-classification-algorithms/
- https://stackoverflow.com/questions/62658215/convergencewarning-lbfgs-failed-to-converge-status-1-stop-total-no-of-iter
- https://towardsdatascience.com/do-you-know-how-to-choose-the-right-machine-learning-algorithm-among-7-different-types-295d0b0c7f60
- https://towardsdatascience.com/nlp-with-spacy-part-1-beginner-guide-to-nlp-4b9460652994
- https://towardsdatascience.com/the-10bias-and-causality-techniques-of-that-everyone-needs-to-master-6d64dc3a8d68

©Bishal Biswas(@WolfDev8675)
(b.biswas_94587@ieee.org)