

# **Investigating the accuracy of classifier algorithms in detecting hate speech on Twitter.**

Research Question: How accurate are classifier algorithms: Logistic Regression and Naive Bayes in classifying hate speech tweets?

**Subject: Computer Science**

Word count: 3300 words

## **Table of Contents**

### **1. Introduction**

### **2. Background Information**

- 2.1 NLP and Machine Learning
- 2.2 NLP and detection of hate speech
- 2.3 Logistic regression
- 2.4 Naive Byes

### **3. Experiment Methodology**

- 3.1 Dataset used
- 3.2 Pre-processing the database
- 3.3 Data cleaning
- 3.4 Experimental Procedure

### **4. The experimental results**

- 4.1 Confusion Matrix data presentation
- 4.2 F1 Score Tabular Analysis
- 4.3 Data analysis

### **5 Conclusion**

## **Introduction**

“Natural Language Processing (NLP) is a subset of Artificial intelligence and Machine learning linked to the ability of the computer to comprehend the human language as it is spoken or written” (Lutkevich). NLP uses AI to analyze and interpret human language. This field has a lot of potential and various real-world applications such as business intelligence, sentiment analysis, predictive text, and voice assistance (Lutkevich).

Some main tasks of NLP include text classification (categorizing texts using keywords), text extraction (summarising text), and language translation. Text classification, in particular, has an important role in sentimental analysis which is helpful in determining the emotion or sentiment behind any text. A variety of statistical techniques are used in machine learning for NLP and text analytics to recognize entities, sentiments, portions of speech, and other properties of text. Supervised machine learning, which is often used for sentiment analysis, often requires large datasets of labeled text that are tagged or annotated including examples of what the computer should check for and how it should interpret that component. Using these datasets, a statistical model is "trained," and untagged text is then provided for the model to analyze. Through training on a large amount of data to identify relevant correlations, ML algorithms are used in prediction, extraction of various text features, and categorization of texts (Lutkevich).

In today's society, the rise of hate speech and offensive content online has become a growing problem, especially on online platforms such as Twitter wherein people from a variety of cultures and educational backgrounds use the platform. A significant obstacle to the automatic detection of offensive language is distinguishing between hate speech and offensive language. In this report, we look at how the Machine Learning classifier algorithm can be used to automatically classify tweets on Twitter into two classes: hate speech and non-hate speech. Introduce the word sentiment analysis

Thus, this paper seeks to investigate and compare the accuracy of two particular popular classifier algorithms, Logistic Regression and Naive Bayes in detecting hate speech on Twitter.

This research has massive potential in automated hate speech detection in social media. Over the past few years, the importance of AI and ML algorithms in sentiment

analysis and automatic flagging has grown. ML algorithms have massive potential in flagging offensive tweets and sending them to authorities for approval. This would save a lot of time, resources and funds since it can lower the workload of removing offensive speech on social media.

To compare the accuracy of the two algorithms at detecting hate speech, a comparative analysis of both algorithms was performed on a Twitter database after using NLP pre-processing techniques. We evaluated the performance of both these algorithms in classifying tweets into hate speech or not based on a previously labeled publically available dataset. The results were later analyzed in terms of their logical and mathematical grounds.

Classifier algorithms, Logistic regression, and Naive Byes have been chosen for the purpose of the study since the paper aims to label the tweets as hate speech/not, and classifier algorithms are known to be highly accurate when a binary classification problem like this (since the tweet has to be classified either into hate speech or non-hate speech\_and the two classes are linearly separable.

## **2. Background Information:**

Machine learning, the study of giving computers the ability to perform tasks they haven't been explicitly programmed to do, is the foundation of Natural Language processing. It entails giving computers sample data to analyze and learn from, and then putting that acquired knowledge to use by performing complex tasks (Stanford University).

While there are numerous forms of machine learning algorithms, classification is the one that sentiment analysis typically uses because wherein computers get to learn how to map data into labels or categories.

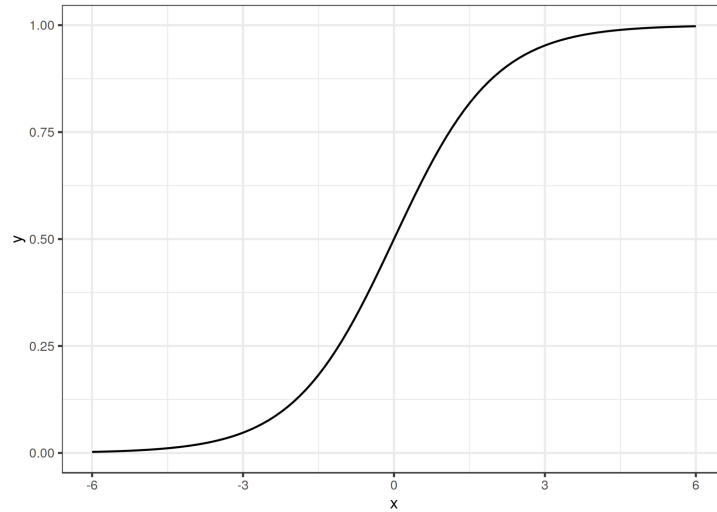
For instance, if a computer were given a collection of tweets arranged according to topics (like sports, music, and politics), the classification algorithm would analyze the properties of the tweets and make an effort to associate particular textual features/words with specific topics. This is referred to as training. If successful, the computers will eventually be able to successfully identify and recognize different tweets it has never seen before owing to correctly identifying the correct relationship between a tweet's characteristics and its type.

Classification of tweets can be achieved using two main types of machine learning: supervised and unsupervised. Unsupervised learning is used to analyze and examine unlabelled datasets. In this, ML algorithms identify occult patterns or data clusters without the assistance of a human. In supervised machine learning, since the dataset is labelled, it is possible to assess the algorithm's accuracy. A supervised machine learning algorithm, put simply, aims to use the mapping function of the labeled data with its inner workings. The dataset is divided into two segments: the training set and the testing set. Until a suitable level of performance is attained, the algorithm repeatedly tries to make predictions on the training data.

In the testing phase, the algorithm generates predictions on the testing dataset and compares the obtained classifications with the actual labels. There are numerous supervised machine learning algorithms available, but this study will concentrate on Logistic Regression, Support Vector Machine, and Naive Bayes classifier, a particular family of classification algorithms. These classifiers will be discussed in greater detail in the sections that follow.

### **Logistic Regression:**

Logistic regression is a machine learning algorithm that can be used to predict a binary outcome, such as a yes or no, like the one in this study. A logistic regression model predicts a dependent data variable by examining the correlation between one or more already present independent variables. For example, based on different features of weather like humidity, and temperature (dependent variables), the probability of rain can be calculated (independent variable). The logistic regression model uses the logistic/sigmoid function to predict the output of a linear equation between 0 and 1 rather than fitting a straight line or hyperplane. A sigmoid function gives a probability between 0 and 1.



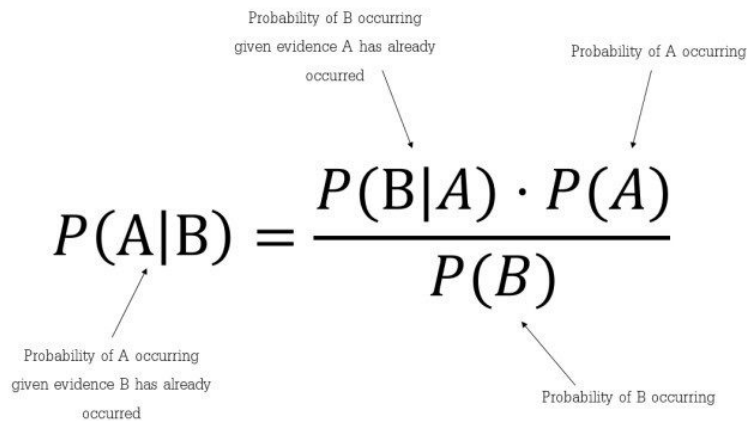
$$S(x) = \frac{1}{1 + e^{-x}}$$

(Molnar)

If ‘the function  $S(x)$ ’ goes to infinity, the output will become 1 (yes) and if  $S(x)$  goes to negative infinity, the output (predicted) will become 0 which corresponds to a no. “Logistic regression makes use of the sigmoid function which outputs a probability between 0 and 1.”(Edgar) For situations involving binary and linear classification, logistic regression is a straightforward and more effective approach. It is a classification model that is relatively simple to implement and performs well with linearly separable classes like the one in this study.

### **Naive Bayes:**

Naive Bayes is a machine learning classifier model based on probability. “The Bayes theorem serves as the foundation of the classifier.” Bayes theorem can be used to calculate the likelihood of happening on the condition of B occurring earlier. Here, A is the hypothesis and B is the proof or evidence (Chahaun).

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$


Probability of B occurring  
given evidence A has already  
occurred

Probability of A occurring

Probability of A occurring  
given evidence B has already  
occurred

Probability of B occurring

Considering the problem of classifying tweets, the variable A ( kind of tweet) is the main variable, which represents whether the tweet is offensive or not based on the features. Variable B, on the other hand, represents the parameters/features. Variable A only has two possible values- a yes or no. Here, it is assumed that the predictors and features are independent or unrelated. Hence, it can't learn the relationship between features and the main variable, making it less suitable for classification problems that emphasize studying the relationship between variables (Chauhan).

For example, if a problem is trying to measure the relationship between playing gold depending on the weather and the parameters (A=a1,a2,a3,a4): humidity, temperatures, rainy/sunny, overcast, etc. The Naive Byes algorithm would consider that there is no relationship between different parameters and that rain and humidity are independent of each other, which in real life is untrue. Moreover, it would place equal emphasis on all the parameters although, in reality, one or two parameters might play a bigger role/have a bigger weight in this classification problem (Chauhan).

### 3. Experiment Methodology

The major source of data in this research is primary experimental data. Two classifier algorithms (Logistic Regression and Naive Byes) were programmed (code in the appendix, heavily adapted from (Kothari, 95)) and fed data from a publically available dataset of tweets, and the resultant accuracy was recorded at labeling tweets containing

hate speech. Because there were few secondary data sources available to address the research question in this paper, an experimental methodology was used because it provides a great degree of freedom in studying the results in detail. However, using primary data had some limitations like time constraints that prevented the use of more datasets to train the models and the use of further different classifier models.

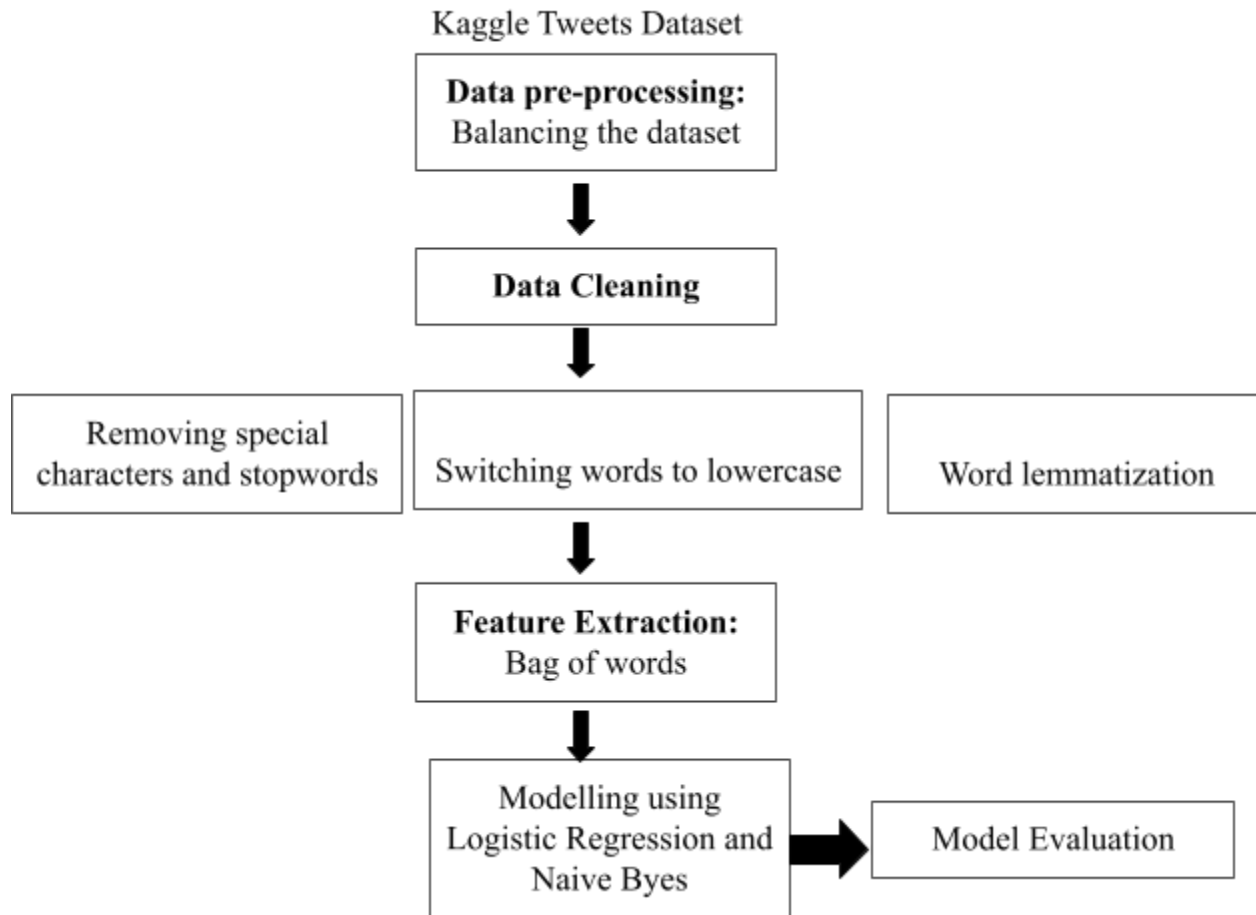
The dataset used for the purpose of the research was “Twitter Hate speech”, a public dataset on Kaggle in the form of a CSV file. The dataset contains 29,695 tweets containing the non-hate labels and 2,2340 tweets containing the hate speech label.

### **Experimental Procedure:**

Our experimental procedure is as presented below:

Firstly, the kaggle dataset will undergo data pre-processing in order to balance the skew of having non-hate labelled tweets as the majority class. Before conducting data analysis, the kaggle dataset must be cleaned of any inaccurate, corrupt, or useless data. Then, “bag of words”, a Natural Processing Language (NLP) technique will be applied in order to extract features from tweets to understand what features of tweets cause them to be offensive or not. NLP techniques like a bag of words are applied in order to extract features from the data that can be used in modeling the ML algorithms (*Towards Data Science*). Next, ML algorithms, Logistic regression and Naive Bayes are trained and modeled on the dataset after which their training accuracy will be recorded.



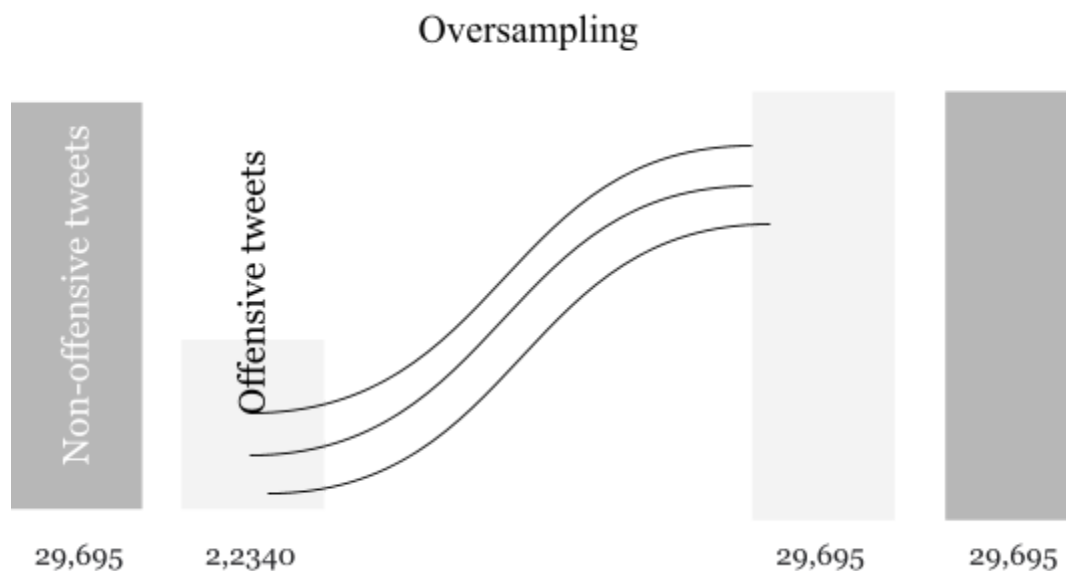


Before the experiment, data pre-processing had to be conducted on the dataset for the analysis to take place.

### **Data pre-processing:**

To increase the model's accuracy and ability to correctly classify hate speech labeled tweets, it was necessary to address the huge skew i.e the imbalance between the number of non-hate tweets and the number of hate speech labeled tweets. This is because imbalanced datasets prevent the classification algorithms from being better able to extract features or gain a better sense of what causes a tweet to be hate labeled due to the low number. Hence, because of an imbalanced dataset, the algorithm may not accurately predict a hate tweet and may favor non-hate tweets. (Towards Data Science)

This problem was solved using oversampling. Oversampling is a method that is used to balance datasets by increasing the size of the overrepresented class by creating artificial data points. SMOTE (Synthetic Minority Over-Sampling Technique) was used in this experiment to generate new tweets through code (Towards Data Science) and increase the size of the training data.



Data Cleaning:

After balancing the dataset, the dataset underwent data cleaning.

- Firstly, data cleaning took place to remove punctuation (“,” “.” “!”) and replace them with space. Then special characters and stop words (Stop words are typically articles or prepositions, distort the context or real meaning of a phrase) that didn't contribute to the meaning of the tweet were also removed.

```
review = re.sub("[^a-zA-z | ^\w+'t]",'', data['Review'][i])
```

```
all_stopwords=stopwords.words('english')
```

```
all_stopwords.remove(words)
```

- Data cleaning took place to make data analysis easier and less complex by eliminating meaningless elements that only bloat the number of words the algorithm needs to learn and extract features from.
- Lemmatization was conducted on the dataset which is connecting different forms of a word to one form in order to analyze all their meaning as one, for example, making “happier”= happy.

(Superior University)

### **NLP Techniques applied:**

#### **Bag of words:**

Finally, to train the models for the classification algorithm and extract features from the tweets to help with prediction, a bag of words, an NLP technique was applied. Bag of words allows the model to study words associated with a particular tweet. A bag of words converts text into vectors to identify the presence of particular words that make a tweet offensive. The technique is known as “bag of words” because the models ignores the structure/style of the data, it just considers words that occur in the document. It works on the principle that if two tweets have similar content/similar words that occur, they would have a similar meaning (Towards Data Science).

It is used to turn each piece of data into a free vector that can be used as an input value to train a Machine Learning model. It creates a list of words that occur throughout the dataset and notes the occurrence of each word in a tweet. If a word is present in a tweet, it attaches a boolean value of 1 to it and 0 if it is not available. Hence, through training our model after applying a bag of words, we can ascertain whether a tweet is offensive or not based on the presence of certain words and their frequency (Towards Data Science).

**Tf-id:**

Term frequency-inverse document frequency is abbreviated as tf-idf. It is a metric that quantifies the significance of a word in a corpus or collection of documents. Tfid is used as a weighting factor in feature generation for ML models.. The more times a word appears in a document, the higher its tf-idf value will be. It is countered by the word's frequency in the corpus, which helps to account for some words that are used more frequently overall.

Term frequencies measure how frequently a word appears in a tweet whereas inverse document frequency is responsible for measuring the log of total number of documents divided by the number of documents containing that word.

$$w_{i,j} = tf_{i,j} \times \log \left( \frac{N}{df_i} \right)$$

(Towards Data Science)

**Results:**

Two classifier algorithms: Logistic regression and Naive Byes were programmed using two separate NLP techniques, 'bag of words and 'TFID' after which their accuracy in clarifying hate tweets was recorded.

Results were studied in the form of confusion matrices which represent the predicted and actual values. When classifying objects, a confusion matrix is frequently used to gauge how accurate the classifiers are. This experiment will use confusion matrices to measure the accuracy rate of classifier algorithms. Confusion matrices produce a table with predicted values and actual values to measure the accuracy. It uses data separate from the training process (Brownlee).

## Confusion matrix

The diagram shows a 2x2 confusion matrix. The columns are labeled 'ACTUAL VALUES' with 'Positive' and 'Negative'. The rows are labeled 'PREDICTED VALUES' with 'Positive' and 'Negative'. The cells contain 'TP', 'FP', 'FN', and 'TN' respectively. Four text boxes with arrows point to each cell: 'TP: Total Positive-values that are predicted and actual positive' points to the TP cell; 'FP: False Positive-values that are predicted positive but are not actually positive' points to the FP cell; 'FN: False: Negative- values that are predicted positive but are not actually negative' points to the FN cell; and 'TN: Total Negative- values that are predicted and actual negative' points to the TN cell.

		ACTUAL VALUES	
		Positive	Negative
PREDICTED VALUES	Positive	TP	FP
	Negative	FN	TN

(Deep AI)

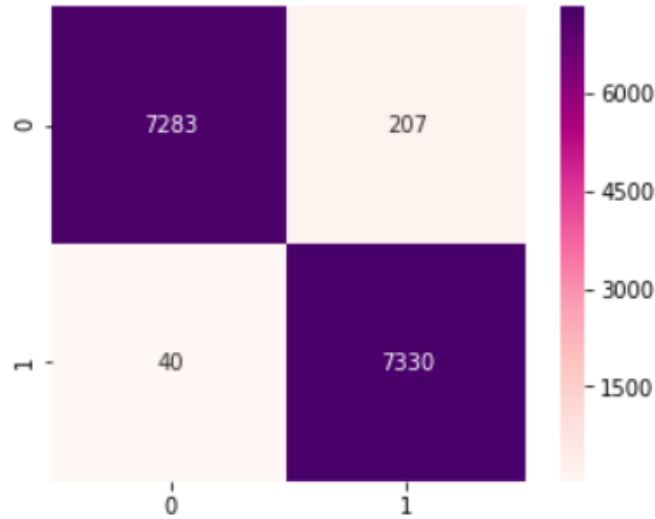
Accuracy is determined by adding tweets that were predicted as hate tweets and were actual hate tweets with tweets that were predicted as nonoffensive and were actually non-offensive tweets and diving that with the total number of tweets.

Accuracy is calculated by:  $\frac{TP+TN}{Total}$

Finally, Machine Learning models of Logistic Regression in Naive Byes were trained and their accuracies at classifying tweets into hate [label=1] or non-hate [label=0] was recorded.

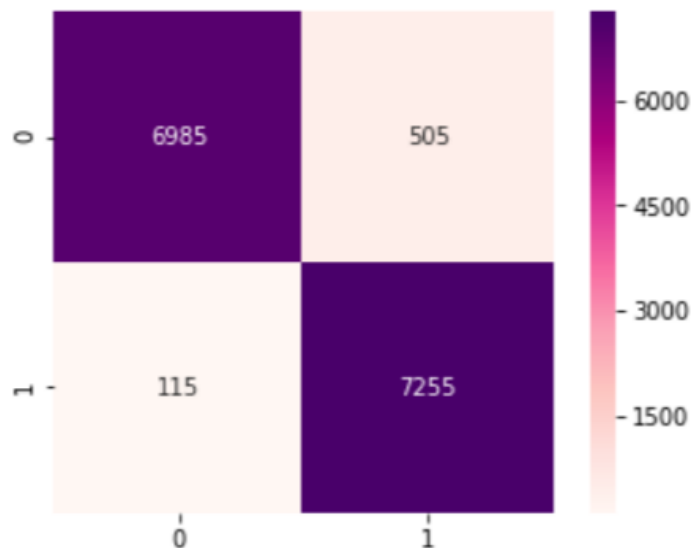
The confusion matrix created through modeling the two algorithms and testing them on the dataset can be seen below:

### Logistic Regression:



The confusion matrix above represents how accurate the Logistic regression model is by showing the total number of predicted and hate speech labeled tweets as 7283 with a minor number of 207 tweets falsely detected offensive.

### Naïve Bayes:



The confusion matrix above represents how accurate the Logistic regression model is by showing the total number of predicted and hate speech labeled tweets as 6985 with a minor number (but greater than the number of false positives in Logistic Regression) of 505 tweets falsely detected offensive.

### Tabular representation of F1 scores:

The accuracy of a model on a dataset can be judged by using F-score, also known as the F1-score. It's utilized to analyze binary classification algorithms, for example labeling tweets as "hate" or "non-hate". F-score combines the model's precision and recall ability. Precision is measured by dividing the number of correctly labeled hate speech tweets by a number of falsely detected hate speech tweets added with true hate speech tweets.

#### Model accuracy scores with both NLP techniques used

		Accuracy Percentage		
		F1 Score	Recall Score	Precision Score
<b>Logistic Regression</b>	<b>Bag of words</b>	98.3%	99.4%	97.2%
	<b>Tfid</b>	97.3%	99.2%	95.4%
<b>Naive Byes</b>	<b>Bag of words</b>	95.9%	98.4%	93.4%
	<b>Tfid</b>	95.7%	98.7%	92%

**Table 1:** Shows accuracy of different classifier models at detecting and classifying hate labeled tweets on the Kaggle dataset

As it can be extrapolated from the table, Logistic Regression has a higher recall score i.e it can correctly identify a higher proportion of hate tweets. The accuracy must also be considered since it gives us an indication of the number of false positives which are important to consider since a large number of them twitter analysts will have to scrutinize a large number of tweets that are not offensive which wastes time and resources. However, in this case, the logistic regression outperforms in the accuracy too by nearly 4%. Hence, logistic regression is clearly a better model to detect hate-labeled tweets as per our results as it maintains a high recall value at the same

time, keeping great precision. Hence, Logistic regression is one of the most efficient algorithms when it comes to predicting an outcome when the data is linearly separable. Perhaps, another reason it performed better than Naive Byes was how it does not make a lot of assumptions.

## **Conclusion:**

This study compares classifier algorithms in order to identify the most suitable to automatically detect and classify an offensive tweet. The dataset used in this study included 2776 hate speeches and 1226 non-hate speeches that had been carefully categorized and approved by another study. (Towards data science ) The demand for automatic hate speech detection methods persists as long as hate speech is a social issue. In this study, we proposed a method to identify hate speech and offensive language on Twitter using Bag of Words and TF IDF values in machine learning. we conducted comparative analyses of Logistic Regression, and Naive Bayes based on various features and parameters. The outcome demonstrated that with the bag of words approach, Logistic Regression outperforms Naive Byes in comparison. According to this study's experiments, the Logistic Regression had the greatest accuracy score (89.3%) when analyzing the Kaggle Dataset.

## **Further Research Opportunities:**

For expanding the study, it may be useful to use more models and compare how they fare in detecting hate speech. Moreover, more complex deep learning models could be considered to yield higher accurate results as deep learning models possess the ability to learn from the model themselves and identify features that correlate faster and atomically.

Moreover, more complex NLP techniques can be used to extract features from the dataset before training ML algorithms since the current techniques: bag of words and tfidf vectorized the tweets based on the frequency of words and not their meanings. Nevertheless, the dataset could be modeled using more classifier algorithms and a variety of datasets could be looked at, especially considering that this dataset is 4 years old and in the language, English.



**Comparison with other studies:**

In another study conducted by Superior university in Lahore, Logistic Regression attained a value of 83% in detecting offensive tweets against Naiye Byes which attend accuracy of 73%. This conforms to the broad result of this paper since even the results conducted on the Kaggle public dataset indicate that Logistic Regression has a higher performance. However, it is important to note that the accuracy of the algorithm in this study is higher than that conducted by Superior university. This may be due to a variety of factors like different training data which is more linearly separable, and overfitting (model bias caused by a modeling error where the data collection is too closely correlated with the model).

## Bibliography:

Singh, Prasoon. "Fundamentals of Bag of Words and TF-IDF." *Medium, Analytics Vidhya*, 15 Feb. 2020, <https://medium.com/analytics-vidhya/fundamentals-of-bag-of-words-and-tf-idf-9846d301ff22>.

Dutta, Mimi. "Bag-of-Words vs TFIDF Vectorization –a Hands-on Tutorial." *Analytics Vidhya*, 14 July 2021, <https://www.analyticsvidhya.com/blog/2021/07/bag-of-words-vs-tfidf-vectorization-a-hands-on-tutorial/>.

Abro, Sindhu, et al. "Automatic Hate Speech Detection Using Machine Learning: A Comparative Study." *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, 2020, <https://doi.org/10.14569/ijacsa.2020.0110861>.

Lutkevich, Ben, and Ed Burns. "What Is Natural Language Processing? an Introduction to NLP." *SearchEnterpriseAI, TechTarget*, 2 Mar. 2021, <https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP>.

St, Indra & Wikarsa, Liza & Turang, Rinaldo. (2016). Using logistic regression method to classify tweets into the selected topics. 385-390. 10.1109/ICACSYS.2016.7872727.

"Detecting Hate Tweets — Twitter Sentiment Analysis." *Towards Data Science*, 19 Dec. 2019, <https://towardsdatascience.com/detecting-hate-tweets-twitter-sentiment-analysis-780d8a82d4f6>.

Agarwal, Rahul. "Twitter Hate Speech." *Kaggle*, 26 July 2018, <https://www.kaggle.com/datasets/vkrahul/twitter-hate-speech>.

"F-Score." *DeepAI*, 17 May 2019, <https://deepai.org/machine-learning-glossary-and-terms/f-score>.

Brownlee, Jason. "What Is a Confusion Matrix in Machine Learning." *MachineLearningMastery.com*, 14 Aug. 2020, <https://machinelearningmastery.com/confusion-matrix-machine-learning/>.

Hassan, Umair (2022): Sentiment analysis using machine learning classification models. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.19783384.v1>

Molnar, Christoph. "Interpretable Machine Learning." 5.2 *Logistic Regression*, 12 Nov. 2022, <https://christophm.github.io/interpretable-ml-book/logistic.html>.

Edgar, Thomas. "Logistic Regression." *Logistic Regression - an Overview | ScienceDirect Topics*, <https://www.sciencedirect.com/topics/computer-science/logistic-regression>.

.

Chauhan, Nagesh Singh. "Naïve Bayes Algorithm: Everything You Need to Know." *KDnuggets*, <https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html>.

Stanford University. Supervised Learning. Web. 7 July 2017. Stanford University. Unsupervised Learning. Web. 8 July 2017

## Appendix:

```
[ ] X_train, X_val, y_train, y_val = train_test_split(train_upsampled['tweet'],
                                                    train_upsampled['label'], random_state =
                                                    X_train.shape, X_val.shape
                                                    ((44580,)), (14860,))
```

```
[ ] vect = CountVectorizer().fit(X_train)
vect
CountVectorizer()
```

```
▶ print('Total features =', len(vect.get_feature_names()))
print(vect.get_feature_names()[0:5000])

Total features = 34928
['00027', 'braves', 'echobelly', 'hornets', 'mic', 'pupils', 'svpol']
```

```
▶ X_train_vectorized = vect.transform(X_train)
X_train_vectorized
```

```
↳ <44580x34928 sparse matrix of type '<class 'numpy.int64'>'
   with 507700 stored elements in Compressed Sparse Row format>
```

```
[ ] import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
model = LogisticRegression()
```

Heavily adapted from:

VISHAKHA , Agarwal, R. (2018) Twitter hate speech, Kaggle. Available at: <https://www.kaggle.com/datasets/vkrahul/twitter-hate-speech> (Accessed: November 30, 2022).