Project Name: Hate Speech Detection

Artificial Intelligence (CSE422)

Moinul Hossain Bhuiyan (20301002)

Submitted To :Mostofa Kamal Sagor
Section 08, CSE422

INDEX

INTRODUCTION

The use of social media platforms has exploded in recent years, and with it, the amount of harmful content being shared online. Hate speech and offensive language are prevalent on social media platforms like X, and detecting such content has become an important issue for social media companies and society as a whole. In this project, we will test our model on a set of test data (X's) to see how well it generalizes to new data. If our model performs well, it can be used to flag and remove hateful or offensive content from X, making it a safer and more inclusive platform for everyone.

Motivation

- Promoting Online Safety and Well-being
- Protecting Vulnerable Communities
- Supporting Content Moderation Efforts
- Raising Awareness and Education
- Ethical AI Development.
- Skill Development and Learning
- Contributing to Research and Academia
- Inspiring Change and Advocacy
- Real-world Application of AI

Overall, our motivation for creating a hate speech and offensive language detection model should stem from a desire to contribute positively to online discourse, promote inclusivity, and leverage technology for the betterment of society.

DATASET DESCRIPTION

Our dataset is from a paper on Automated hate Speech Detection by Davidson et al. Here's the link to the actual paper and dataset.

https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset

Reference of Dataset: Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language.https://arxiv.org/abs/1703.04009

The dataset contains X's labeled as hate speech, offensive language, or neither.

It includes a total of 24,802 X's, with 3,972 labeled as hate speech, 20,298 as offensive language, and 432 as neither. The dataset includes the following features

Thus, this is a classification problem where the goal is to predict the class of a given X's as one of the three categories mentioned above. 0,1,2 refer to hate speech, offensive, or neither in the dataset in the class column respectively.

5000 rows from the dataset are used and we used the class column and X's column for our model.

The visualization of the frequency of data was done using the Matplotlib library to create a bar chart. This chart shows the frequency of each label category, which can give us an idea of how the X's are distributed among the different categories.

PRE-PROCESSING TECHNIQUES APPLIED

To start with, as seen in Figure 1, our dataset is imbalanced, and the results are unevenly distributed among classes. This mismatch can have a negative impact on how our models perform in terms of precision, recall, and f1 scores overall. To address this, we used data oversampling to evenly distribute the dataset among all three groups.

There were some unnecessary and non-informative features in the dataset. So we applied several preprocessing approaches that are more suited to NLP tasks. These methods were used to remove unimportant and non-informative elements from the dataset. The methods are listed below.

1. Removal of Punctuations
2. Lower casing
3. Tokenization
4. Removal of Stop Words
5. Lemmatization

DATASET SPLITTING

Dataset splitting is a common practice in machine learning to evaluate the performance of a model on unseen data. The most common way to split a dataset is to divide it into a training set and a test set. The training set is used to train the model, while the test set is used to evaluate its performance. A commonly used split is to allocate 80% of the data to the training set and 20% to 3 the test set. This split can be done randomly, ensuring that both the training and test sets have a representative distribution of the data.

MODEL TRAINING & TESTING

With the completion of count vectorization using bag-of-words, our dataset is now prepared for training. In this project, we have employed three distinct classification algorithms to train our dataset. These algorithms have been chosen based on their suitability to our specific problem and the expected results we hope to achieve. We will use these algorithms to classify our data and derive valuable insights from it. The three algorithms we have used are identified below, and each of them offers unique advantages that make them ideal for our dataset.

1. Logistic Regression
2. Naive Bayes Classifier
3. Kth Nearest Neighbor(KNN)

Logistic Regression :

Logistic regression is a statistical method that can be used to predict the likelihood of a binary outcome, such as whether an individual is using hate speech or not. In this study, logistic regression could be applied to predict the probability of the speech being Hate speech. The technique is commonly used for classification tasks and is particularly effective when predicting categorical dependent variables from a set of independent variables. The logistic function is used to transform the output of a linear equation into a value between zero and one, representing the probability of the binary outcome. This transformation enables the model to make predictions based on the relationships between the independent variables and the outcome variable. In 4 summary, logistic regression is a powerful statistical tool that can be used to predict the likelihood of a binary outcome based on a set of input variables.

Naive Bayes Classifier :

Naive Bayes Classifier is a probabilistic algorithm used for classification in machine learning. It is based on Bayes' theorem, which states that the probability of a hypothesis (class) is directly proportional to the probability of the evidence (features) given that hypothesis. The "naive" assumption in Naive Bayes is that all the features are independent of each other, given the class. This means that the probability of each feature can be calculated separately and then multiplied together to get the overall probability. Naive Bayes Classifier is commonly used for text classification, spam filtering, sentiment analysis, and recommendation systems. It is fast and efficient, and can handle large amounts of data with a relatively small amount of memory. There are different variants of Naive Bayes Classifier, such as Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes, which differ in their assumptions about the distribution of the features.

Kth Nearest Neighbor(KNN):

K-nearest neighbors (KNN) is a classification technique that involves finding the nearest known data points in pattern space to classify unknown examples. KNN operates by calculating the Euclidean distance between an unknown example and all known examples in the dataset and then selecting the k closest points. The class of the unknown example is predicted based on the most common class among its k-nearest neighbors. Thus, KNN relies on the assumption that instances with similar attributes tend to belong to the same class, making it a useful and simple method for classification tasks.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

K-nearest neighbors (KNN) is a classification technique that relies on the Euclidean distance to determine the similarity between an unknown data point and existing data points. The k closest points are selected and their class labels are used to predict the class label of the unknown point through majority voting. This approach assumes that instances with similar attributes tend to belong to the same class, making KNN useful for classification tasks. In summary, KNN identifies the most similar known examples to a new example and uses their class labels to predict the class label of the new example.

MODEL SELECTION/COMPARISON ANALYSIS

Confusion Matrix:

The confusion matrices for each of our models are shown below. The confusion matrix provides a summary of how well a classification method performed. The numbers in the principal diagonal reflect the frequency with which models made accurate predictions of labels. The alternative values are inaccurate forecasts.

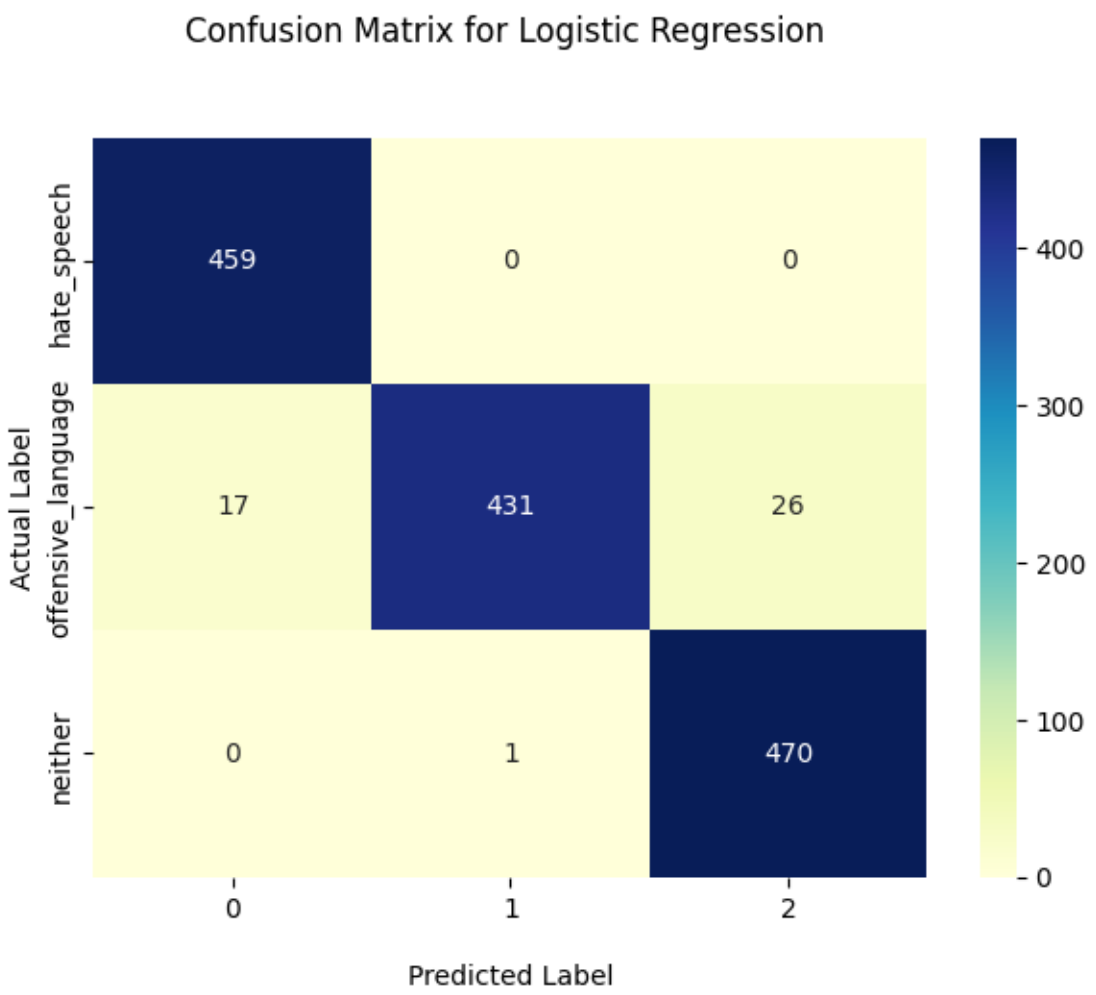

Figure 1: Confusion matrix for logistic regression

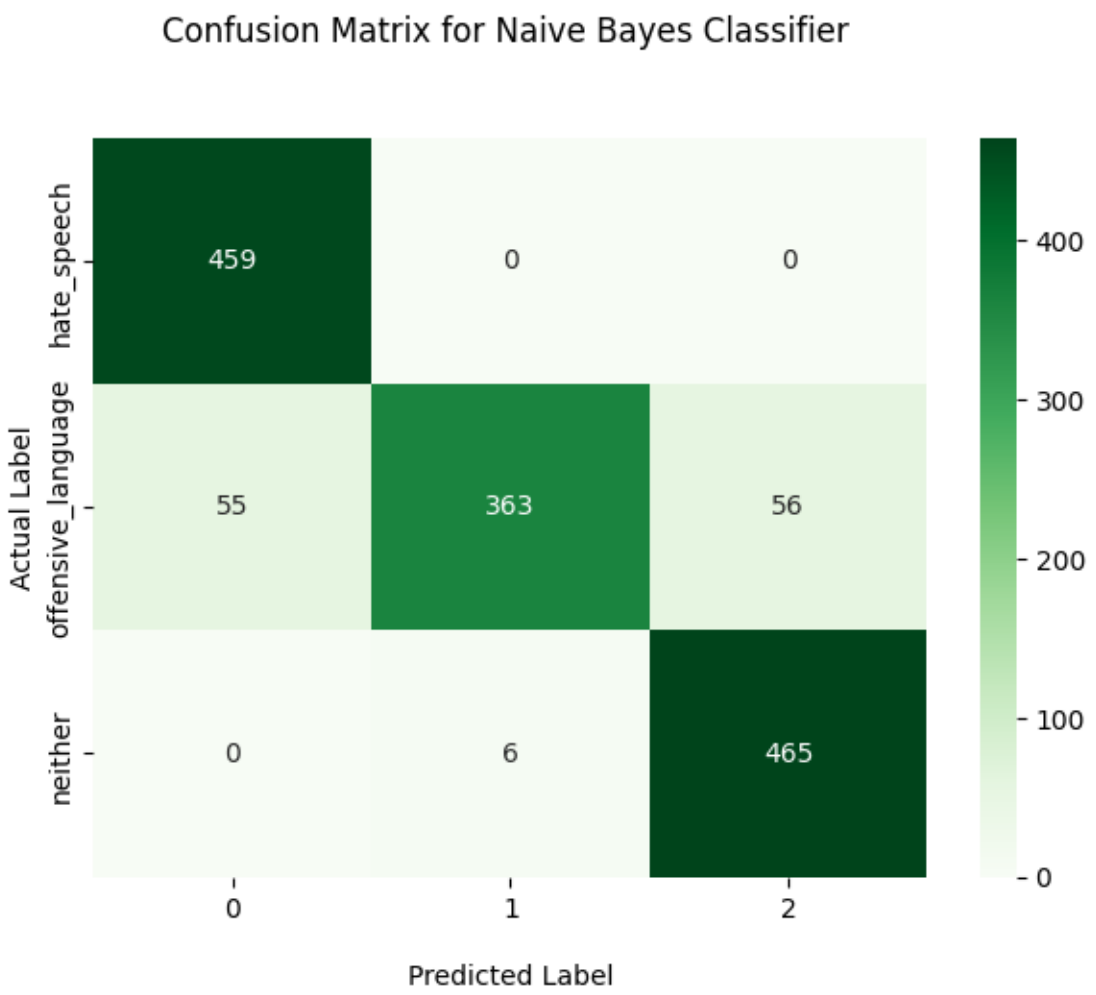

Figure 2: Confusion matrix for Naive Bayes Classifier

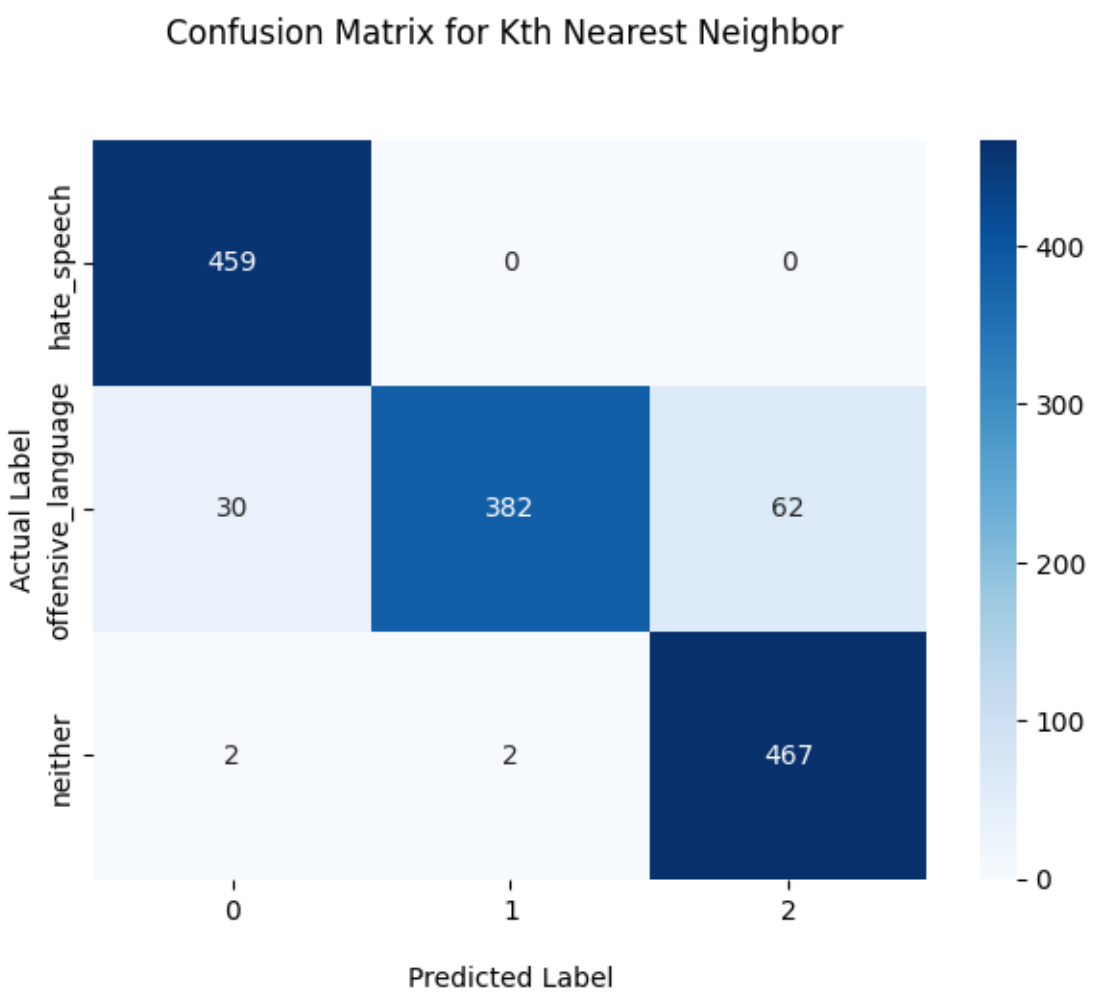

Figure 3: Confusion matrix for Kth Nearest Neighbour (KNN)

Logistic Regression:

Accuracy: 0.9686609686609686

	precision	recall	f1_score
0	0.96	1.0	0.98
1	1.0	0.91	0.95
2	0.95	1.0	0.97

Naive Bayes Classifier:

Accuracy: 0.9166666666666666

	precision	recall	f1_score
0	0.89	1.0	0.94
1	0.98	0.77	0.86
2	0.89	0.99	0.94

KNN:

Accuracy: 0.9316239316239316

	precision	recall	f1_score
0	0.93	1.0	0.97
1	0.99	0.81	0.89
2	0.88	0.99	0.93

Result Analysis

We can see that here; Logistic regression has Higher number of accuracies compared to naïve bayes classifiers and KNN. By using Logistic regression, we got around 96% of accuracy whereas Knn and naïve bayes had around 89~90% of accuracy. These things can also be noticed in the heatmap provided above. Thus we can say that, The Logistic regression has the best accuracy.

CONCLUSION

In order to classify hate speech and offensive X's throughout this study, some efficient classifier algorithms and mathematical models were used. Our dataset was trained using three classification algorithms: Logistic Regression, Naive Bayes Classifier, and Kth Nearest Neighbour (KNN). By looking at the overall result of these algorithms, the most accurate of them is Logistic Regression, while the least effective is the Naive Bayes Classifier.

Future Work/Plan

- Develop techniques to identify sarcastic or ironic statements, as these can sometimes be misinterpreted by traditional models.
- Build a system that monitors Twitter feeds in real time and provides feedback to users when their posts contain potentially offensive content, allowing them to reconsider before posting.
- Design techniques to make your model's decisions more understandable. This is especially important in applications involving moderation and potential content removal.
- Collaborate with social media platforms to integrate your model into their content moderation systems to enhance user safety.
- Ensure that your model's deployment respects user privacy and ethical considerations. Strive to minimize false positives and negatives to avoid undue censorship or allowing harmful content.

Reference:

1. Deepansi, “Text preprocessing NLP: Text preprocessing in NLP with python codes,” Analytics Vidhya,19-Jul-2022.[Online].
Available: Text Preprocessing in NLP with Python codes (analyticsvidhya.com) [Accessed: 31-Aug-2022].
2. Pynative.com. [Online]. Available:Python Regex Split String using re.split() – PYnative. [Accessed: 31-Aug-2022].[Accessed: 31-Aug-2022].
3. “How to remove any URL within a string in Python,” Stack Overflow. [Online].
Available: regex - How to remove any URL within a string in Python - Stack Overflow
4. V. Aruchamy, “How to plot confusion matrix in python and why you need to?,”
Stack Vidhya. Available at: How To Plot Confusion Matrix In Python And Why You Need To? - Stack Vidhya [Access: 29-sept-2021]