Project 2: Predicting Youtube Spam Comments

A Kaggle Inspired Project

By
Eddie Dew

October 8th, 2024

Introduction

Youtube is one of the largest social media platforms in the world with over 100 million users interacting with the platform on a daily basis. It is a safe haven for individuals to post videos expressing their views on certain topics, free music, and much more. It also allows public feedback where users can comment on the video expressing their viewpoints, critiques, or overall enjoyment. Unfortunately, with this popularity comes an infection of programs and people who try to exploit the enjoyment for everyone. One of the most prevalent infections are spam comments. Spam comments, as defined by the Merriam-Webster dictionary, refer to unsolicited messages sent to a large audience without consent. These comments create several issues, including wasting users' time, consuming memory, and increasing bandwidth usage, which can negatively impact content load times. More critically, spam is often used for malicious purposes, such as distributing computer viruses and executing scams designed to steal personal information. Gandra (2014) highlights that for every video on YouTube, there is an approximate ratio of 100 spam messages to 1 legitimate message (Othman, N. F., & Din, 2019, p. 1509). Therefore, detecting and filtering spam comments is crucial, both to safeguard users from scams and to optimize internet resource usage. This paper proposes a classification model to predict and detect spam comments, alongside a thorough evaluation of the model's performance.

Dataset Description

The dataset used for this algorithm is sourced from Kaggle and titled "YouTube Comments Spam Detection." It includes features such as the comment author, date, content, and the video associated with each comment. The dataset comprises 1,956 comment observations across six columns. Notably, the only missing values are found in the 'date' column, with 245 missing entries, accounting for 13% of the dataset. The target variable classifies comments as either spam (denoted by 1) or non-spam (denoted by 0). The dataset is relatively balanced, with a near 1:1 ratio of spam to non-spam comments (51% spam to 49% non-spam). The videos in this dataset were all music videos made from various authors such as Eminem, Katy Perry, and PSY (5 total music videos to be exact).

Exploratory Data Analysis

The first feature of interest lies within the ‘content’ column, what the message entails. Figure 1 shows what words appeared most frequently in spam and non-spam comments.

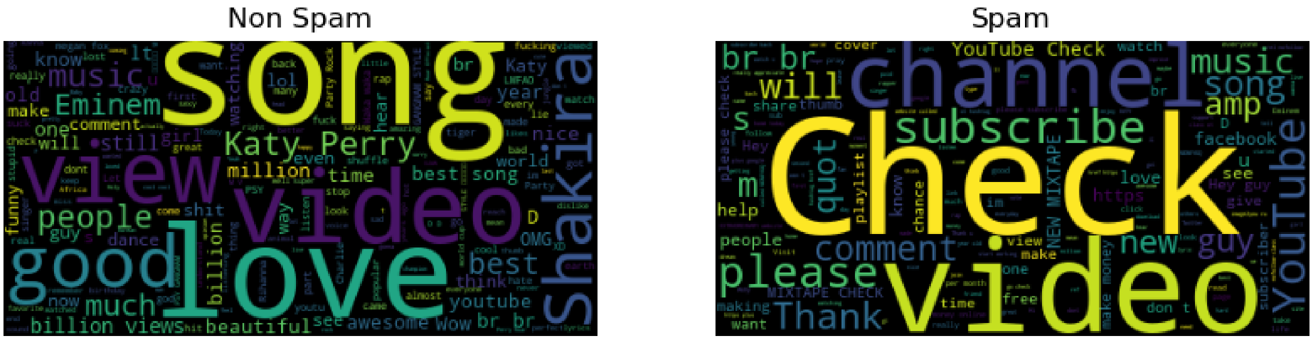

Figure 1: Word Cloud for Non-Spam and Spam comments

Words such as check and video makes sense as spam comments are a form of social engineering trying to steer you to something next. Words such as love and song go interchangeably so most of the comments that are non-spam are most likely users who enjoyed the video.

The next feature explored was the time of day the comment was sent to see if there was enough of a trend to detect spam. Figure 2 displays this trend.

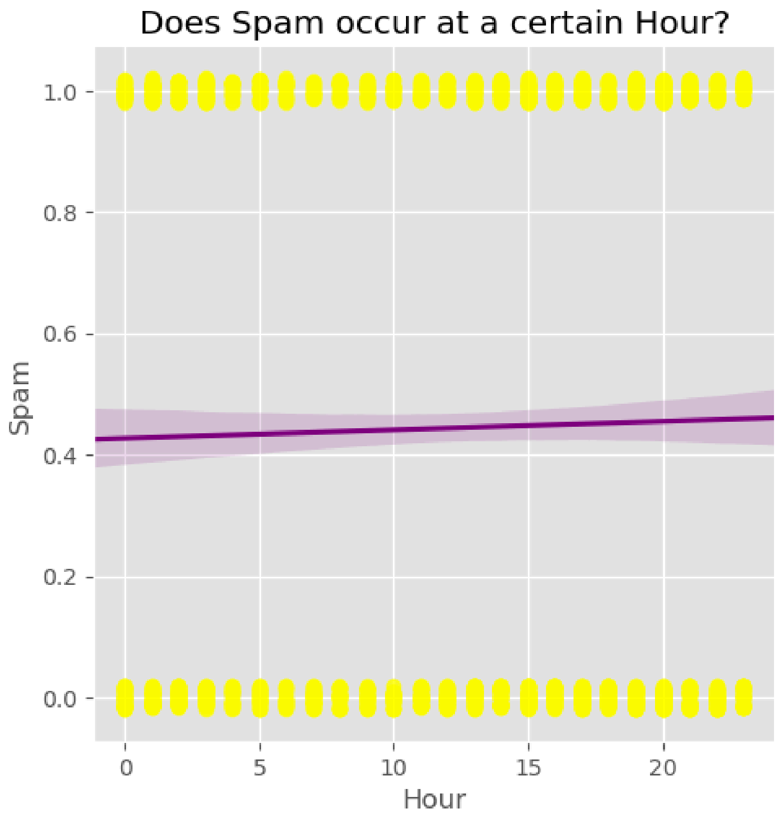

Figure 2: Youtube comment influx throughout the day

While a slightly positive trend, denoting spam messages occur later in the day, it is not a reliable trend to use in a model. When taking a look at the mode for each comment however, the most non-spam comments occurred around the 17th-20th hour and spam was at the 19th hour.

The last feature explored was the content length, the number of words each comment contained. Figure 3 helps explain the distribution of the content lengths for non-spam and spam.

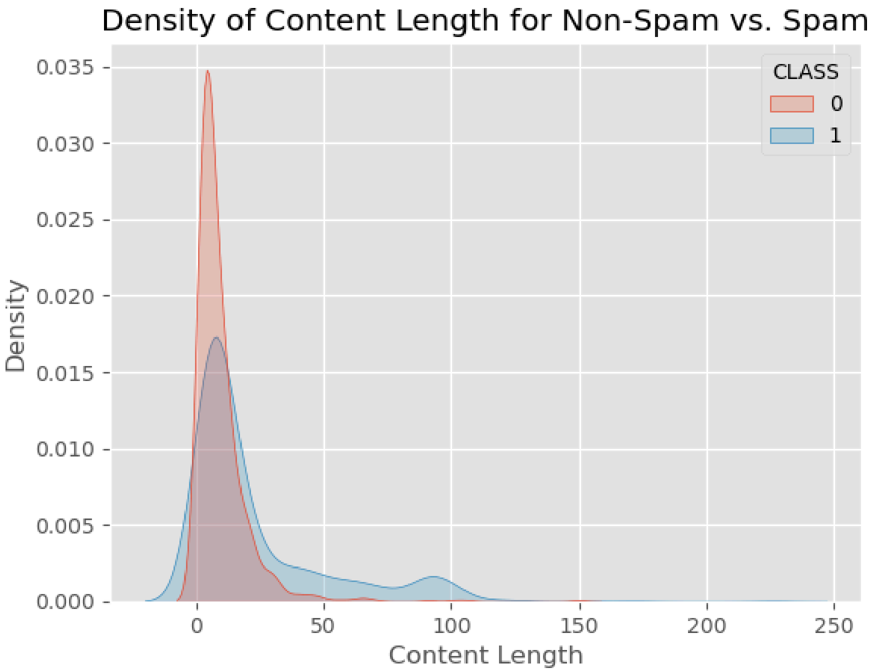

Figure 3: Distribution of content length for spam and non-spam

Both distributions appear to have a right skew where spam comments generally have longer content length. This feature would be helpful for a model, unfortunately it was the only one this paper found useful which means a new method will need to be implemented to make reliable predictions.

Tf idf Vectorizer

The purpose of TF-IDF is to highlight words that are important within a specific document but less common across all documents in a corpus. The process begins by calculating the term frequency (TF), which measures how frequently a word appears in a single document. It then adjusts this value using inverse document frequency (IDF), which decreases the weight of words that appear frequently across many documents. In the context of this dataset, TF-IDF will examine the words in the 'content' column, identifying commonly used terms such as 'song,' 'check,' 'video,' and 'love' (as shown in Figure 1). However, words that are less frequently used across the dataset, such as 'amp,' will be assigned a higher TF-IDF score, indicating their relative importance within individual comments. When this is fit to the ‘content’ column it returns a sparse matrix where each word from each row is scored and a total of 3,594 words are used. Looping through the iterations and summing the scores, we can find the terms most frequently used in this vectorizer with their score. Figure 4 shows the top 5 words with their summed score.

Word	**TF-IDF Score**
Check	95.769
Video	78.932
Youtube	78.059
Song	68.187
Love	65.798

Figure 4: Top 5 words based on summed TF-IDF Score

Depending on how their scores were for each row will be how they are valued. For example, if the first row is a spam comment and ‘check’ is valued the highest, the model will learn to associate ‘check’ with spam.

Model

Two machine learning models, Logistic Regression and Support Vector Classifier, were employed to predict spam. Both models are well-suited for balanced datasets, which implies they should yield reliable predictions. Logistic Regression, in particular, works by fitting an S-curve to the data while applying L2 regularization. This regularization helps mitigate bias by shrinking variable weights, thus improving the generalizability of the model. Support Vector Classifier (SVC) operates by fitting a linear decision boundary through the data using a linear kernel. A key strength of SVC lies in its ability to map the data into higher dimensions, allowing for more effective separation when the data is not linearly separable. In this case, the model was tuned and it was determined that polynomial regression in the first dimension offered the best predictive performance for spam detection. This approach allowed the model to fit a curved decision boundary within a one-dimensional feature space, enhancing its ability to differentiate between spam and non-spam messages. Figure 5 depicts the classification report for both models.

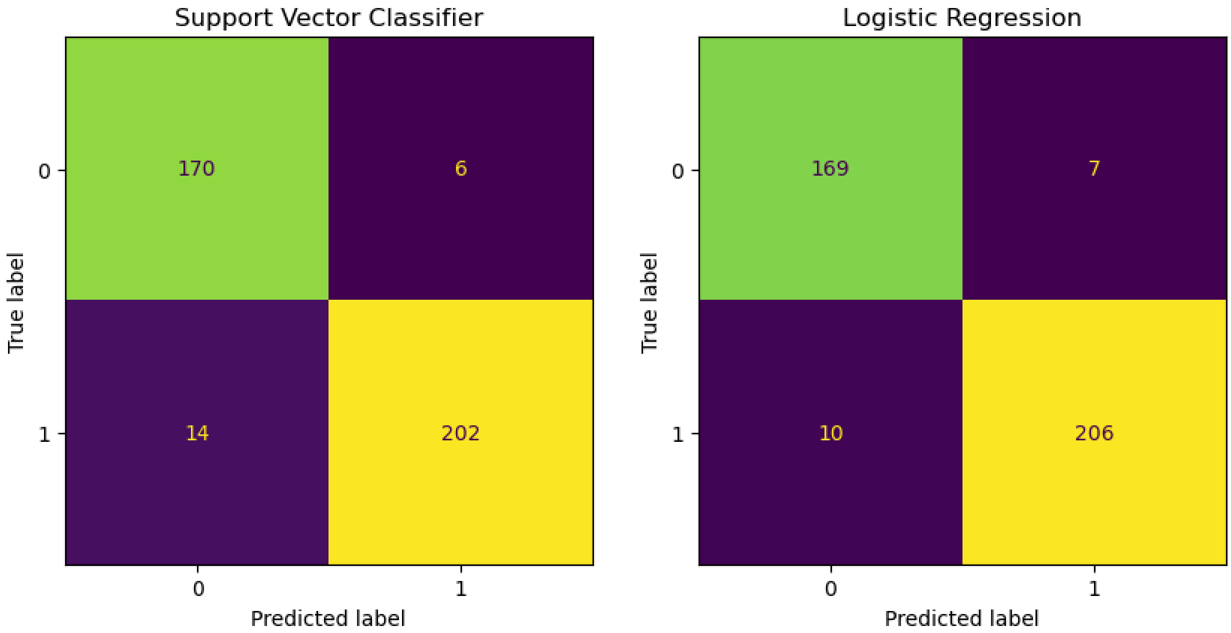

Figure 5: Classification report for both models

Both models demonstrated strong performance, achieving accuracy scores of 96% for the SVC and 95% for Logistic Regression. However, accuracy alone is insufficient for selecting the optimal model, particularly in contexts where minimizing specific types of errors is critical. This study focuses on reducing false positives, where non-spam messages are incorrectly classified as spam, in order to safeguard users' freedom of speech by preventing genuine comments from being falsely flagged. As such, precision, which measures the proportion of correctly identified spam messages among all messages classified as spam, is the primary metric of interest. SVC outperforms Logistic Regression in this regard, achieving a precision score of 96% compared to 95%. Notably, SVC's higher precision on non-spam messages, 94% versus 92% for Logistic SVC, further justifies its selection as the preferred model for this task.

Impact

This model performed better than anticipated, despite the limited use of derived features and relying primarily on weighted word frequencies, highlighting the efficacy of the TF-IDF vectorizer. However, the model faces challenges when handling imbalanced datasets, as well as spam comments that may lack indicative keywords such as 'check' or 'subscribe'. While it demonstrates strong predictive accuracy, the variability in comment structure, content and scalability can undermine the model's confidence in identifying spam.

References

Alberto, T.C., Lochter, J.V. *Youtube Comments Spam Dataset*. Kaggle, 2024, https://www.kaggle.com/datasets/ahsenwaheed/youtube-comments-spam-dataset/data.

Gandra, S. (2014). *Implementation Of Prototype To Detect Spam In YouTube Using The Application TubeKit And Naïve Bayes Algorithm*.

Othman, N. F., & Din, W. I. S. W. (2019). *Youtube spam detection framework using naïve bayes and logistic regression*. Indonesian Journal of Electrical Engineering and Computer Science, 14(3), 1508-1517.

spam. 2024. *Merriam-Webster.com*, Retrieved October 8, 2024, from www.merriam-webster.com/dictionary/spam.