440 Spam Detection Project Write-up

By: Skyllar Estill, Molly Iverson, Anne Tansengco, and Emma Fletcher

Background and Problem Statement

With the transition into the digital age, a large amount of our communication is done through digital means. These include but are not limited to, phone and video calls, texts, emails, and social media sites. With communication styles moving digital, additional problems such as unsolicited messages are popping up. Many of these messages can carry malicious text, links, and images. It is these unsolicited messages that make spam and phishing ploys. While spam and phishing can be found in calls, texts, and social media instant messages, it is most prevalent in email. One survey mentions that around “45% to 50%” of email communication today is spam, with others saying “as much as 85% of all email traffic today is spam” [1]. With the mass amount of spam in existence today, it is essential to have a filter on email inboxes. This spam filter is exactly what we are looking into for this project, more specifically, we will be looking at what algorithms can be used and which ones work the best for detecting and filtering spam with accuracy.

Solution Approach

As spam is such a prevalent issue in today’s society, it is a problem that gets looked at frequently. One of the questions that pops up is, what exactly is the best way to filter spam? Most people agree AI and Machine Learning Algorithms are the way to go, as they “can accurately identify and filter spam emails” [2]. However, this does not mean that all AI and Machine Learning algorithms are suitable for spam, and among the ones that are, there may be clear benefits to using a certain algorithm over another. Common spam detection and filtering algorithms include, “Naïve Bayesian classification, Support Vector Machine, K Nearest Neighbor, and Neural Networks” [3]. Based on what we have found are the most commonly used algorithms, we decided to implement the K Nearest Neighbor, Naive Bayes, and C4.5 decision tree algorithms for this project to compare and determine which one is better suited for spam detection based on accuracy and computing time and space.

First, we preprocessed the data to start with clean, identical datasets and split it into training and testing datasets at a 70/30 ratio. We decided to use a 70/30 ratio for training and testing because studies show that when 20-30% of data points are allocated for testing and the rest of the 70-80% of data points are used for training, the accuracy estimates are the most valid and the most accurate [5]. We also shuffled the dataset before running it through each algorithm because the email dataset we tested with comes sorted with all spam listed first. We have tested a series of different K values ranging from 0 to 17 to determine which value works best for the K Nearest

Neighbor algorithm implementation. For the Naive Bayes classifier, we used the top 100 most informative features from the transformed data features for the spam classification for better efficiency of evaluating based on features. Finally, we implemented the C4.5 decision tree algorithm using recursion to build the trees and determine spam predictions from the tree. Then, we were able to accurately compare the results between the three algorithms to determine which is better at detecting spam with maximum accuracy, and minimum computing speed and resources.

Results and Discussions

We used a standard accuracy percentage and a confusion matrix to analyze each algorithm's results. A confusion matrix is a table consisting of the number of accurate and inaccurate instances predicted by a model through true positives, true negatives, false positives, and false negatives [4]. The true positives indicate the times when the algorithm accurately predicts a spam email, the true negatives indicate the times when the algorithm accurately predicts a non-spam email, the false positives are for inaccurately predicting a spam email, and the false negatives are for inaccurately predicting a non-spam email. From these points, we could better understand each algorithm's accuracy.

K Nearest Neighbor

The K Nearest Neighbor algorithm works by splitting the shuffled email data into 70% training and 30% testing data, creating a dictionary of the counts of each word within each email from the training data, creating the same type of dictionary for the testing data emails, and finding the distance from each testing email to each training email. We tested both the Manhattan distance and the Euclidean distance to find the distance with the most accuracy and found that using the Euclidean distance had better accuracy. So, the Euclidean distance was found from each testing email to each training email. If the K-closest emails were the majority spam, the testing email was labeled spam.

Upon testing our range of K values to determine the optimal one to use in the algorithm, we found that K=11 had the best detection accuracy of 81%. With this K value, the confusion matrix showed 2 false positive results and 323 false negatives.

Naïve Bayes

Similarly to the KNN algorithm, the shuffled email data was split into training and testing data for the Naive Bayes Classifier (NBC). This time, the training and testing data was also split into X and y subsets. The X subset held the text of the emails while the y subset held the spam classification values. We used the Python library sklearn’s CountVectorizer tool to gather all features from the X training data, and then we further filtered the features with SelectKBest to

train the data with the top 100 most informative features for the best accuracy and efficiency. We trained the classifier to learn the probability of these features given the class labels from the y subset so the NBC could predict the classification of spam or not spam for the test data. With this algorithm, we found that the Naive Bayes had a spam detection accuracy of 95%. The confusion matrix for the NBC had 26 false positive predictions and 53 false negative predictions.

C4.5 Decision Tree

Once again, the shuffled email data was split to be used as the training and testing data to be used in the C4.5 decision tree algorithm. This algorithm also made use of X and y matrices for feature and label information to predict the spam email classification. This algorithm differed, however, as it used recursion to build a decision tree based on the training data. The nodes within the tree were determined based on the information gained from a decision, and the leaf nodes were the decisions themselves. The prediction for spam classification was then performed with the testing data by searching through the tree with each test data point until a leaf node has been reached, in which the classification of the data point is assigned. This algorithm had a very high spam detection accuracy as well, with an accuracy of 95%. The confusion matrix showed 43 false positives and 34 false negatives for the decision tree predictions.

Comparison

Based on the results from the accuracy tests above, we found that the KNN had the worst spam detection accuracy out of the three algorithms. The KNN accuracy was 14% less accurate than the Naive Bayes Classifier and the C4.5 Decision Tree algorithm. So, the KNN algorithm seems to be the least effective in determining the classification of spam or non-spam emails. The Naive Bayes Classifier and the C4.5 Decision Tree algorithm seemed to perform at the same level of accuracy since we found a 95% accuracy in both algorithms using the same dataset. To determine the best algorithm between these two, we looked into the confusion matrix values as well as the overall time and space performance of the algorithms.

The Naive Bayes Classifier had 26 false spam predictions while the C4.5 Decision Tree algorithm had 43 false spam predictions, so the C4.5 Decision Tree algorithm had 17 more false positives than the Naive Bayes Classifier. The Naive Bayes Classifier had 54 false non-spam predictions while the C4.5 Decision Tree algorithm had 34 false non-spam predictions, so the Naive Bayes Classifier had 20 more false non-spam predictions than the C4.5 Decision Tree algorithm. This data shows that the Naive Bayes Classifier is more likely to falsely predict that spam emails are non-spam emails than the C4.5 Decision Tree algorithm, but the C4.5 Decision Tree algorithm is more likely to falsely predict that a non-spam email is spam.

While the confusion matrix helps visualize the strengths and weaknesses of the C4.5 Decision Tree and Naive Bayes Classifier algorithms when it comes to spam prediction, it still doesn’t

give substantial evidence of which one could be better overall. So, we looked at the algorithm performance itself to determine which algorithm would be best to use in detecting spam emails. While the C4.5 Decision Tree algorithm was very accurate in predicting spam in the whole dataset, the process of building the decision tree was very time-consuming, taking around 6 minutes to create. The tree also takes up a lot of space, so there is a time and space drawback to using the C4.5 Decision Tree algorithm. On the other hand, Naive Bayes Classifier did not need to build any structure like a tree, allowing it to have a much quicker execution. Because the Naive Bayes Classifier has better time and space efficiency than the C4.5 Decision Tree algorithm, it seems that it is the best algorithm out of all three to use for spam email detection because while the C4.5 Decision Tree algorithm is just as accurate as the Naive Bayes Classifier, there is a sacrifice in time and space for it to be used, whereas one could have the same level of accuracy in detecting spam with better execution using the Naive Bayes algorithm.

From the comparison of the three algorithms through accuracy and performance, it is clear that the Naive Bayes algorithm is the best algorithm to use for detecting spam emails. While the KNN algorithm has a good level of accuracy at 81%, it is the lowest of the three algorithms we have tested. The C4.5 Decision Tree is an incredibly accurate algorithm but has drawbacks when considering the time and space taken up by the decision tree. Naive Bayes has a very high accuracy and is a more time and space-efficient algorithm. Through these findings, we conclude that the Naive Bayes algorithm is the best algorithm for email spam detection.

References

[1] R. Lever, “What Spam Email Is and How To Stop It,” *U.S. News*, Oct. 12, 2022. https://www.usnews.com/360-reviews/privacy/what-spam-email-is#:~:text=Some%20surveys%20place%20the%20percentage,links%20and%20attachments%20you%20click

[2] “4 Ways AI Improves Email Handling,” *Eccentex*, Feb. 23, 2023. https://www.eccentex.com/2023/02/23/4-ways-ai-improves-email-handling/

[3] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: review, approaches and open research problems,” *Heliyon*, vol. 5, no. 6, p. e01802, Jun. 2019, doi: https://doi.org/10.1016/j.heliyon.2019.e01802

[4] GeeksforGeeks, “Confusion Matrix in Machine Learning - GeeksforGeeks,” *GeeksforGeeks*, Feb. 07, 2018. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

[5] H. M. Basir, A. Javaherian, Z. H. Shomali, R. D. Firouz-Abadi, and S. A. Gholamy, "Acoustic wave propagation simulation by reduced order modelling," *Exploration Geophysics*, vol. 49, no. 3, pp. 386–397, Jun. 2018, doi: https://doi.org/10.1071/eg16144