**Case Study 3: Spam Classification Using Naïve Bayes and Clustering**

# 1 Introduction

The goal of this project is to build a spam classifier. A spam classifier is a program that identifies real emails versus spam emails based on the information provided by a baseline set of standard emails. We were provided 5 files identified as Ham and Spam. Ham in this case is considered real emails. 3 of the files are ham emails and 2 of the files are spam. Within these files we were provided with everything that comes in an email. Using the *email* package in Python, we extracted the subject, body, and the sender of the email. From there, we were tasked with a binary classification problem where the email is either spam or not spam.

# 2 Methods

There were 6,941 emails labeled as “Not Spam (Ham)” and 2,379 emails labeled as “Spam”. Our dataset is not heavily imbalanced so there was no need for under-sampling or over-sampling techniques. As the team received the emails in a bulk zip, the team utilized the *HTMLParser*, *Email* and *OS* module from Python. In order to extract relevant information, each email was read from their organized zip directories using the RFC-5332 protocol. Each email was able to be pre-labeled by interpreting which subdirectory the email was stored in (e.g., if in a folder containing “ham” label as 0 for non-spam, otherwise label as 1 for “spam”).

Furthermore, the emails extracted required regular expressions to remove newlines, breaks and other symbols. From there, we joined each email’s body with the subject and sender so the model can learn the context of these features. No data imputation was required for the missing data since most of the missing data resulted from some errors thrown during the extraction of the content of the emails. For preprocessing, we used Sklearn’s CountVectorizer to vectorize the data. This resulted in a sparse matrix of 9320 x 70542, consisting of 70,542 input features, and 9,320 observations. Finally, we selected the Sklearn’s StratifiedKFold for the cross-validation method with 10 splits, because this method allows testing on balanced classes.

For modeling, we used the Multinomial Naïve Bayes Classifier. Then, we used two clustering algorithms K-Means and Agglomerative Clustering to see if there was an improvement on accuracy. We chose 2 clusters for K-Means and 3 clusters for Agglomerative Clustering.

# 3 Results

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Clustering Algorithm** | **Number of Clusters** | **Mean Accuracy over Folds** |
| Multinomial Naïve Bayes | N/A | N/A | 99.20% |
| Multinomial Naïve Bayes | K-Means | 2 | 99.20% |
| Multinomial Naïve Bayes | Agglomerative | 3 | 99.20% |

For K-means, the elbow plot shows inertia for each value of K. We can see that the optimal number of clusters is 2 for this clustering algorithm. Considering what we know about the data 2 clusters is expected since we are looking at spam or not spam.

*Figure 1: Elbow Plot for K-Means using inertia*.

Chart, line chart

Description automatically generated

For Agglomerative Clustering, we want to observe the effect of adding a small number of clusters. As noted from the results chart, we can see that clustering has no effect on accuracy. In fact, we think that clustering does not have an impact on prediction accuracy for this model because the data is so sparse, to begin with. Therefore, adding these features has no impact on the model. Using the bag of words to vectorize the training data results in over 70,000 input features. Adding a single additional cluster feature to the training data results in no improvement to the model due to the high dimensionality of the count vectorizer features.

# 4 Conclusion

In all, we see that the Naïve Bayes is a robust classification algorithm for spam classification. Furthermore, Naïve Bayes’ largest advantage is low computation cost. Additionally, Naïve Bayes handles class imbalance effectively. However, the disadvantage to Naïve Bayes is concept drift. Our study found that the Multinomial Naïve Bayes classifier resulted in 99.20% accuracy for every model tested. The bag of words model is an effective way to vectorize text data for spam classification. Lastly, our study utilized Agglomerative and K-means clustering labels to be added as new features. We found that these added features have no impact on the model due to the already high dimensionality and sparsity of the training data.

# 5 Appendix Code

