**Improving Email Filtering: A Pre-Trained FastText Model for Spam Detection**

**Team Members:**

Jyothika Mohan

Chi-En (Ian), Hwang

Sanka Naga Nitesh

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **Page No.** |
| EXECUTIVE SUMMARY | 3 |
| BACKGROUND AND DOMAIN KNOWLEDGE | 4 |
| FIRM STRATEGY | 4 |
| PROJECT ANALYSIS: PROJECT OVERVIEW | 5 |
| PROJECT ANALYSIS: APPROACH 1 | 6 |
| PROJECT ANALYSIS: APPROACH 2 | 7 |
| PROJECT ANALYSIS: APPROACH 3 | 8 |
| RECOMMENDATIONS AND BUSINESS VALUE | 9 |
| SUMMARY AND CONCLUSIONS | 10 |
| APPENDIX | 11 |

**EXECUTIVE SUMMARY**

According to recent studies, spam emails account for over 50% of all email traffic, with millions of spam emails being sent every day. These emails not only clutter inboxes, but also pose significant threats to individuals and businesses, including phishing attacks, malware infections, and data breaches. In fact, it is estimated that spam emails cost businesses billions of dollars each year in lost productivity and security breaches. With the increasing sophistication of spammers and cybercriminals, it is becoming more important than ever to have effective spam filtering techniques in place to protect against these threats.

A spam filter is an essential tool for managing email in the modern digital age. With the amount of spam email being sent each day, it can be difficult for individuals and businesses to sort through the clutter and find important messages. Spam filters work by using various techniques to identify and block unwanted email, allowing users to focus on the messages that matter. In this project, we propose a spam filter based on natural language processing (NLP) techniques. By encoding the text of emails using bag of words and word2vec techniques, as well as a pretrained fasttext model, we are able to classify emails as either spam or not spam with high accuracy. We explore various classification techniques and evaluate their performance using precision, recall, and F1 scores. Our results demonstrate the effectiveness of our approach, and we believe that our spam filter has the potential to provide significant benefits to individuals and businesses alike in the management of their email communications.

**BACKGROUND AND DOMAIN KNOWLEDGE**

Employing a spam filter has a significant impact on various industries, especially those that rely heavily on email communication, such as e-commerce, financial services, and healthcare. By detecting and filtering out unwanted spam emails, these industries can significantly reduce the risk of cyber threats, such as phishing attacks, malware infections, and data breaches. A spam filter can also help improve the overall productivity of an organization by reducing the time and effort required to manually filter out spam emails. Employees can focus on important emails without the distraction of spam messages cluttering their inbox. Additionally, a spam filter can help ensure that important emails are not lost or overlooked due to the volume of spam emails received.

The primary product or service offered by this project is an effective and accurate spam filter that uses natural language processing techniques to identify and classify emails as either spam or not spam. The spam filter can be used to automatically filter out spam emails, reducing the amount of time and effort required to manually sift through unwanted emails. There is significant competition in the market for spam filters, with many companies offering similar products and services. However, this project offers several advantages over its competitors. Firstly, it employs state-of-the-art natural language processing techniques that have been shown to be highly effective at identifying spam emails. Secondly, it offers a high degree of flexibility and customization, allowing users to tailor the filter to their specific needs and preferences. Finally, it is easy to use and integrate into existing email systems, making it a convenient and efficient solution for businesses and individuals alike.

**FIRM STRATEGY:**

Traditionally, industries have attempted to solve the problem of spam emails by using rule-based filters that block emails based on specific keywords or phrases. However, these filters can be unreliable and may block legitimate emails that contain similar keywords. In this project, we are using a machine learning approach to spam filtering that utilizes natural language processing techniques. We have explored various techniques for feature encoding, such as Bag of Words and Word2Vec, and have employed a pretrained FastText model that is then trained on the email dataset.Our spam filtering strategy aligns with the business model of providing a reliable and accurate filter that can effectively reduce the risk of cyber threats and improve productivity for industries that rely on email communication. By using machine learning techniques, our approach is more accurate and less prone to false positives compared to traditional rule-based filters.

**PROJECT ANALYSES**

**Project Overview:**

In this project, we aim to implement a spam filter using NLP techniques. We have tried three different approaches to accomplish this task.

The first approach involves using a pretrained model (FastText) and fine-tuning it on our email dataset to generate document vectors. We then use these document vectors along with other covariates such as the “count of the illegitimate website links” in the email, “number of uppercase letters” to predict whether an email is spam or not.

The second approach involves using a Word2Vec model trained on our email dataset to generate document vectors. These document vectors are then used as feature embeddings to predict whether an email is spam or not. The Word2Vec model captures the semantic meaning of words in the text, which helps in capturing the context of the email and hence better classification of spam and non-spam emails.

The third approach involves using TFIDF on the email dataset for feature extraction. The extracted features are then used to predict whether an email is spam or not. This approach is based on the idea that the frequency of occurrence of a word in the text can be used to distinguish between spam and non-spam emails. We have compared these 3 approaches below:

**APPROACH 1: USING A PRETRAINED FAST TEXT MODEL FOR FEATURE EXTRACTION - THE BEST APPROACH OUT OF ALL 3 METHODS FOR SPAM DETECTION**

This approach involved training a pre-trained FastText model on a dataset of emails to predict whether an email is spam or non-spam.

We started by loading and performing exploratory data analysis on the dataset, followed by cleaning the data by removing punctuation, converting uppercase to lowercase, removing special characters, and removing multiple spaces. The cleaning function also extracted the count of illegitimate email links and the count of uppercase letters in each email. Another thing to note here is that cleaning is done on the email dataset so that the format is similar to the data the fast text model was trained on.

Next the pre-trained FastText model was then implemented, and its ability to capture meaning was verified by checking the similar words for the word "kind". The model was then trained on the cleaned email dataset, and the dimension of the output vector was checked, along with verifying that the similar words detected by the new model could capture context effectively. Document vectors were then generated by taking the mean of all the word vectors in each email, and a logistic regression model was built to predict whether an email is spam or non-spam based on the document vectors and the additional features of count of illegitimate email links and count of uppercase letters. The model achieved an overall accuracy of 0.97, with high precision and recall scores for both spam and non-spam emails, and a high F1-score for non-spam emails.[appendix 3]

Overall, this approach demonstrates the effectiveness of using pre-trained models for natural language processing tasks such as spam prediction. By fine-tuning the pre-trained FastText model on the email dataset, the model can learn to generate document vectors that are tailored to the characteristics of email text, which can improve the accuracy of the logistic regression model in predicting spam and non-spam emails. The inclusion of additional features such as count of illegitimate email links and count of uppercase letters can also improve the model's ability to differentiate between spam and non-spam emails.

**APPROACH 2: USING TF IDF FOR FEATURE EXTRACTION THEN USING DIFFERENT METHODS LIKE LOGIT, RANDOM FORESTS AND NAIVE BAYES FOR CLASSIFICATION**

This approach aimed to explore an alternative approach to spam detection using TF-IDF (Term Frequency-Inverse Document Frequency) for feature extraction, followed by classification using logistic regression, random forests, and Naive Bayes.

We started with loading and performing exploratory data analysis on the dataset, followed by cleaning the data by removing punctuation, converting uppercase to lowercase, removing special characters, and removing multiple spaces. In this approach, lemmatization was also included to ensure that different forms of the same word are treated as the same feature.

**Note regarding lemmatization:**

1. Lemmatization and its relevance in feature extraction: By lemmatizing words before feature extraction, you can ensure that different forms of the same word are treated as the same feature. In summary, lemmatization can help to reduce the number of unique words in a dataset and ensure that different forms of the same word are treated as the same feature, leading to more accurate and effective feature extraction in natural language processing tasks.

2. It is also important to note that we did not lemmetize in the previous section where we built document vectors using the fast text pretrained model. As we are matching the format of email data to the format of the data that the pretrained model was trained on. (So we do not use lemmatization in the previous section)

TF-IDF was then used for feature extraction, which assigns a weight to each term in a document based on how frequently it appears in the document and how rare it is across all documents in a corpus. The output is a matrix of numerical values that represents each document's features. Next, three different models were built to predict whether an email is spam or non-spam based on the TF-IDF features and additional features such as the count of illegitimate email links and the count of uppercase letters in each email. These models include logistic regression, random forests, and Naive Bayes. The logistic regression model achieved the highest accuracy of 97%, followed by the random forest model with an accuracy of 94%, and the Naive Bayes model with an accuracy of 92%. Even in terms of F1-score, the logistic regression model performs the best with a score of 95%, followed by the random forest model with a score of 89%, and the Naive Bayes model with a score of 84%.[appendix 4]

**APPROACH 3: USING WOR2VEC FOR FEATURE EXTRACTION THEN USING LOGIT, FOR SPAM DETECTION**

The third approach for spam detection involves the use of Word2Vec, a neural network-based approach that captures the semantic relationships between words. This approach is different from the previous two approaches. Word2Vec is more powerful than a TFIDF in capturing the meaning and context of words in a document or corpus as it considers the relationships between words, whereas TF-IDF only considers the frequency of words.

We preprocess the data like the previous steps, then a Word2Vec model is trained on the email data. The document vectors are generated by taking the mean of each word inside each email. The logistic regression model is then used to classify the spam emails based on the Word2Vec doc vectors. The model learns to identify patterns and relationships between the word embeddings to accurately classify spam emails.

The results show that the Word2Vec model has a slightly higher overall accuracy and f1-score, as well as better recall for identifying spam emails compared to the TF-IDF model. Word2Vec captures the semantic relationships between words and documents, which can provide better context for classification tasks like spam detection. Both models have high precision, which indicates a low false positive rate, meaning the models are good at identifying non-spam emails.[appendix 5]

Even Though this model performed better than the TF IDF model as it was able to capture context more effectively than a TFIDF, ***Word2Vec does not perform better than the fine tuned fast text model when it comes to capturing contextual information. Detailed explanation along with output results in the appendix****.*[appendix 2]

**RECOMMENDATIONS AND BUSINESS VALUE**

The model built using a pre-trained FastText model for spam detection can bring significant value for businesses by improving their email filtering system. By accurately detecting and flagging spam emails, businesses can ensure that their employees are not wasting time on unnecessary emails and can focus on more important tasks. This can lead to increased productivity and efficiency within the organization. Moreover, the severity of false positives in email filtering cannot be underestimated. Misclassifying an important email as spam can have serious consequences, such as missed deadlines, lost opportunities, or damaged relationships with clients or partners. Therefore, using a model that can effectively capture the context of emails, such as a pre-trained FastText model, can help reduce the risk of false positives and ensure that important emails are not mistakenly flagged as spam. In addition, implementing a robust spam detection system can also enhance cybersecurity measures for businesses, as spam emails are often used as a vehicle for phishing attacks and other malicious activities. By filtering out spam emails before they reach employees' inboxes, the risk of cyber attacks can be significantly reduced.

One recommendation for the business would be to implement a threshold-based approach that can be a useful addition to the spam detection system for businesses. In industries where the severity of false positives is high, such as in the banking and financial sector, it may be advisable to set a higher threshold to reduce the risk of important emails being mistakenly flagged as spam. On the other hand, for industries that are more prone to phishing attacks, such as healthcare or education, a lower threshold may be more appropriate to ensure that potential phishing emails are caught. By providing a threshold functionality, businesses can have greater control over the spam detection system and can tailor it to their specific needs and risk tolerance. This can further enhance the accuracy and effectiveness of the spam detection system, while also reducing the risk of false positives.

Overall, employing a robust spam detection system that includes a pre-trained FastText model, fine-tuned on the specific email data set, along with a logit classification, can bring significant business value. By reducing the risk of false positives, improving productivity, and enhancing cybersecurity, businesses can ensure that their email communication is secure and efficient, which can have a positive impact on their overall operations and bottom line.

**SUMMARY AND CONCLUSIONS**

In conclusion, we explored three different approaches to spam email detection using machine learning techniques. The first approach involved using a pre-trained FastText model, which was fine-tuned on our email dataset. The second approach involved using a TF-IDF model for feature extraction, followed by logistic regression for classification. The third approach used a Word2Vec model for feature extraction, followed by logistic regression for classification.

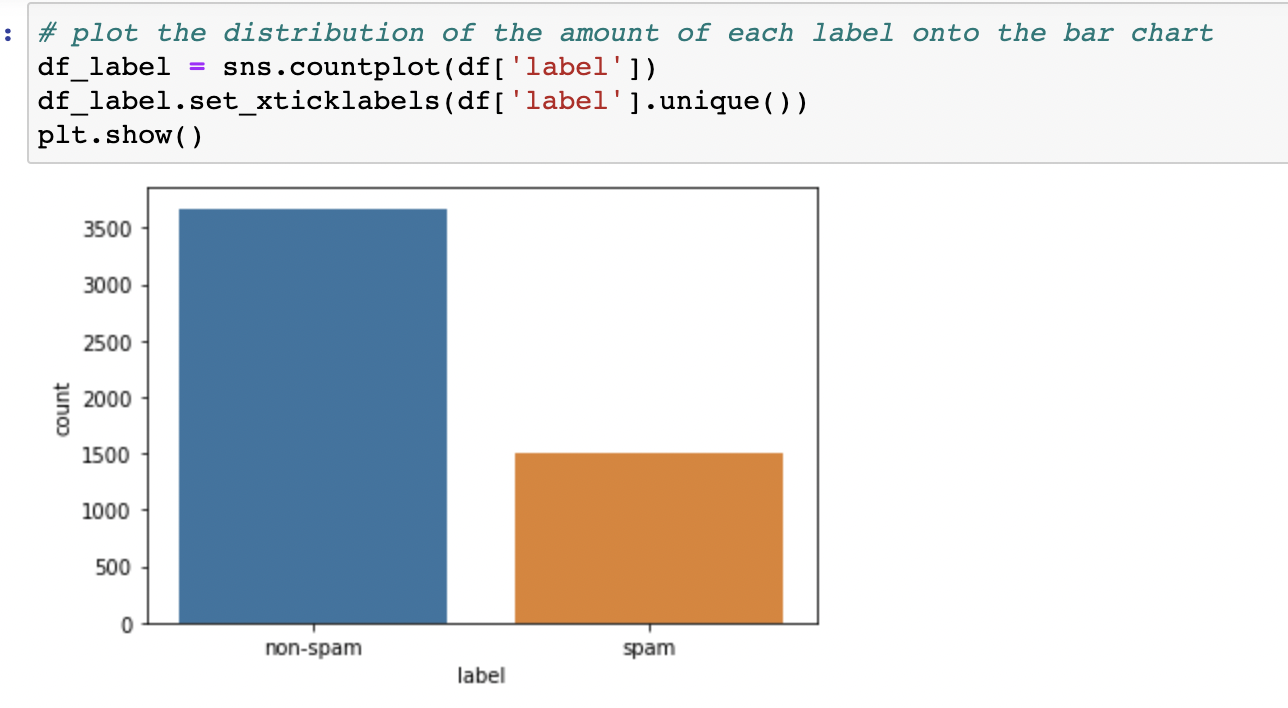
Comparing the three approaches, we found that the pre-trained FastText model performed the best, achieving the highest F1 score and accuracy. This is likely due to FastText's ability to capture subword information, allowing it to better handle out-of-vocabulary words and misspellings. Additionally, the FastText model was able to capture context-specific meanings of words, which may not be possible to this degree with the Word2Vec or TF-IDF models alone. While the Word2Vec model had a slightly higher accuracy and recall compared to the TF-IDF model, the pre-trained FastText model outperformed both.

In conclusion, the use of pre-trained FastText models with fine-tuning is a promising approach for spam email detection, as it offers superior performance compared to other approaches.

**Appendix:**

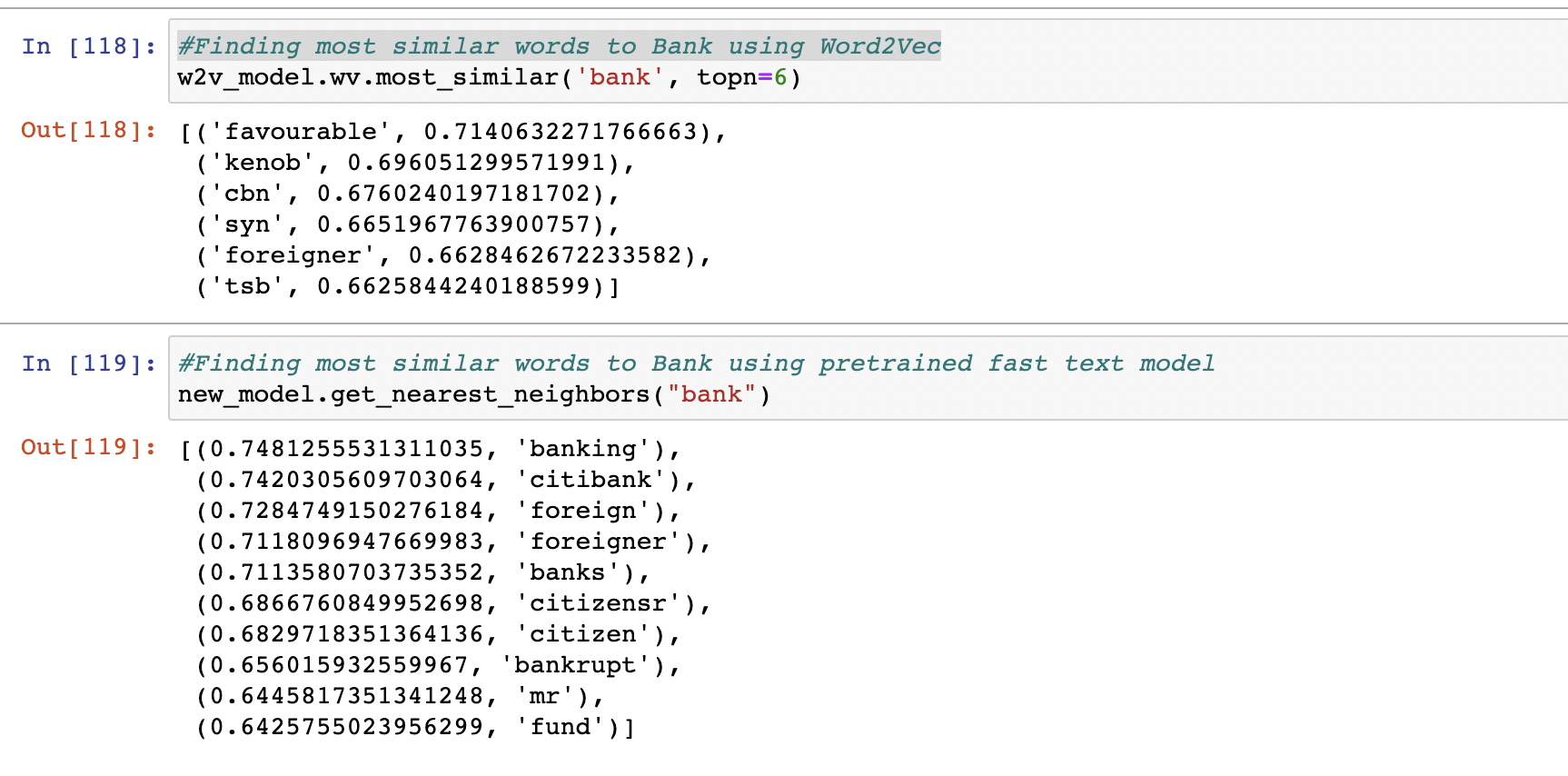
1. **Choice on balancing classes: (This is was discussed with professor)**

In some cases, one class may be rare compared to the other class, and the goal of the model is to accurately predict the rare event. In this scenario, balancing the classes may lead to a model that is biased towards the majority class, resulting in poor performance on the rare class**.**

****

We can see that there is an imbalance in the classes - between the spam emails and the non spam emails, but we have chosen to not balance the classes because in most real life data, spam emails would be lesser than the legitimate emails one receives. Percentage of the spam class is clase to 40% which is not very less, that the model may fail to capture it. This has been further confirmed by the models performance on the test dataset.

1. **Superiority of the Fine Tuned Fast text model in capturing meaning over a Word2Vec trained on the email dataset only:**

****

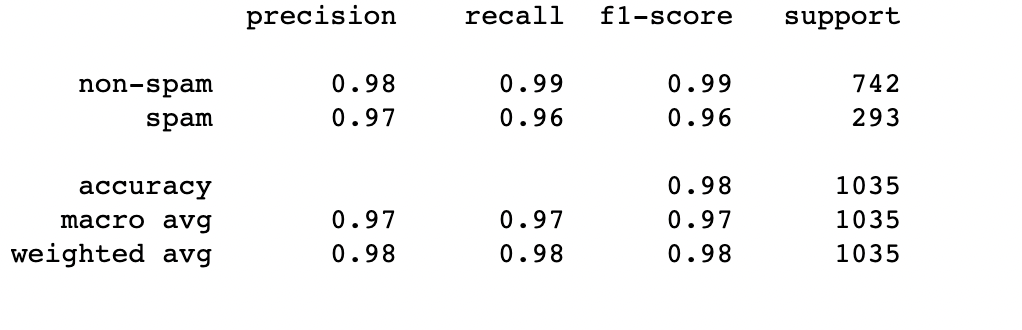
Above we can see that the meaning of Bank is captured better by the pretrained model, as the most similar words are more relevant in that case than using just a word2Vec.

FastText is a supervised learning algorithm for text classification that is built upon word embeddings (like Word2Vec), but with the additional capability of capturing subword information. Specifically, FastText uses character n-grams (sequences of consecutive characters) to represent words as vectors, in addition to the traditional word-level embeddings. This allows for the representation of words that may not appear in the training data set, as well as better handling of misspellings and out-of-vocabulary words.

When we fine-tune a pre-trained FastText model on our email data set, we are essentially adapting the existing word embeddings to the specific language and vocabulary used in our emails. This is different from training a Word2Vec model from scratch on our data set, which may not be able to capture the nuances and context-specific meaning of the words used in the emails.

By using a pre-trained FastText model as a starting point, we are able to leverage the large amounts of data used to train the original model and still achieve good performance on our specific task of spam detection. Fine-tuning the model on our email data set further improves its ability to capture meaning and context, leading to more accurate predictions compared to just using a Word2Vec model trained on our data set.

1. **Output of approach 1 using Pre Trained fast text model:**

****

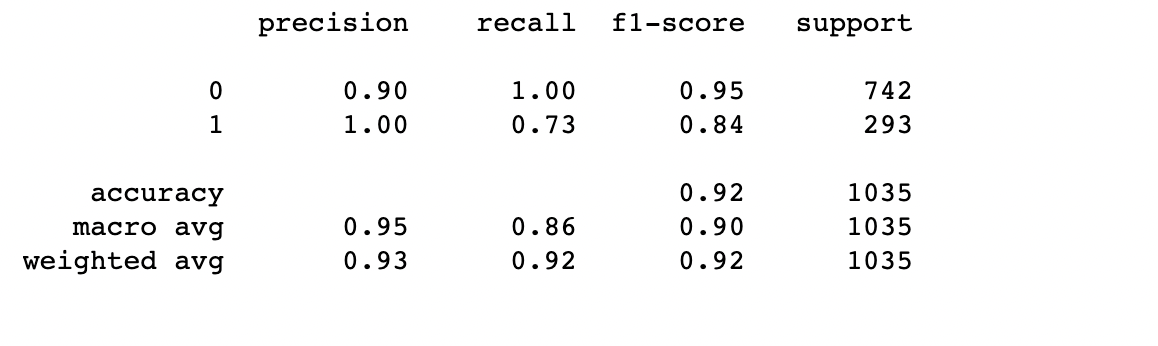
Based on the output of the logistic regression, the model achieved an overall accuracy of 0.97 in predicting whether an email is spam or non-spam. The precision for non-spam emails is 0.97, meaning that 97% of the emails classified as non-spam were actually non-spam. The recall for non-spam emails is 0.99, indicating that the model correctly identified 99% of the non-spam emails.

For spam emails, the precision is 0.98, implying that 98% of the emails classified as spam were indeed spam. The recall for spam emails is 0.92, meaning that the model correctly identified 92% of the spam emails.

The F1-score, which is the harmonic mean of precision and recall, is 0.98 for non-spam emails and 0.95 for spam emails. These scores suggest that the model performed well in identifying non-spam emails but may have some difficulty in correctly identifying all spam emails.

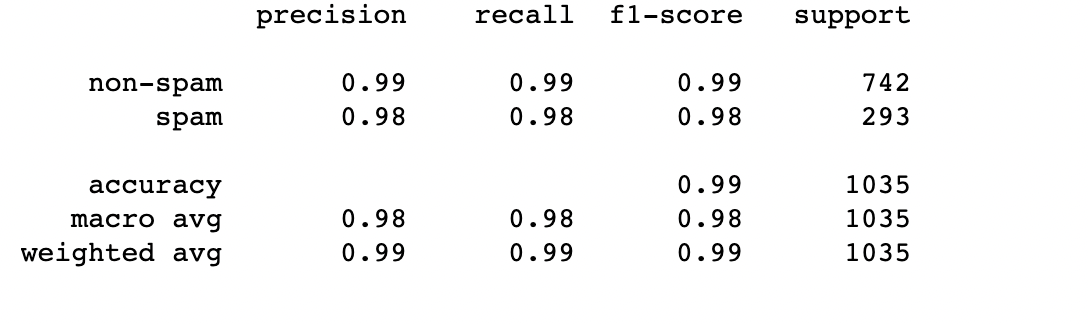
Overall, the model's performance appears to be satisfactory, with high precision and recall scores for both spam and non-spam emails, and a high F1-score for non-spam emails.

1. **Output of approach 2 by using TFIDF for feature extraction:**

****

These were output metrics after running a naives bayes on the TFIDF extracted document vectors. The fast text model definitely outperformed the TFIF

1. **Output of approach 3 by using Word2Vec for feature extraction**

****

Comparing the logistic regression model outputs after using Word2Vec and TFIDF for feature extraction to detect spam emails, it appears that the Word2Vec model has a slightly higher overall accuracy and f1-score, as well as better recall for identifying spam emails. This is likely because Word2Vec captures the semantic relationships between words and documents, which can provide better context for classification tasks like spam detection. However, both models have high precision, which indicates a low false positive rate, meaning the models are good at identifying non-spam emails. Overall, using either TFIDF or Word2Vec models with logistic regression can be an effective approach for detecting spam emails, with Word2Vec offering a slight advantage in accuracy and recall.