Using Logistic Regression and Support Vector Classification to identify spam emails in the SpamBase dataset

Introduction to the Project and Background Information

This project aimed to determine how accurate the classical machine learning algorithms of logistic regression and support vector machines were in classifying spam emails. For its data source, it uses the SpamBase dataset held at the University of California Irvine Machine Learning Repository.[1]

This project was motivated by an interest in the how spam filters are designed. As it is not always possible to prevent spam before it has been sent, a more practical tactic is to identify if an email is likely spam upon arrival and therefore redirect it away from the recipient's inbox where it could be read and potentially be damaging (in the case of phishing emails).

The logistic regression algorithm had better performance, with fewer false positives and false negatives than the support vector machine. Limitations of this project relate to the time-sensitive nature of the dataset, as the content of spam emails has changed in response to detection methods, and the results could be improved upon by using more recent spam messages.

Literature Review

'Spam' is a broad term, vaguely defined as 'unwanted emails' including those from commercial companies advertising their services, prank emails, phishing schemes and scam messages.[2] Spam messages were estimated to climb from approximately 10% of of total email sent in 1998 to 90% by 2013, encouraging the use of automated classification approaches.[3, p. 1]

A rules-based method for classifying emails was proposed by Cohen in 1996, wherein emails would be represented as vectors of the relative frequency of words in the email, and an algorithm would continually add rules until all emails of a known type were correctly classified.[4, pp. 18–19] This content-based approach was extended by using machine learning algorithms, such as support vector machines, k nearest neighbours and neural nets, whilst other approaches (e.g. using heuristics, comparing to recent emails) developed to identify spam by using different patterns.[5, p. 2] The implementation of some of these approaches have been criticised, such as server-side classification removing the choice of whether to treat something as spam from users rather giving them the final decision.[6] This concern seems reasonable, particularly if non-spam emails might be filtered out at the server and never reach clients, but it does require an effective and efficient algorithm to be in developed.

Content and rules-based approaches are imperfect, however, and require retraining. Wang, Irani and Pu's research, based on a corpus of 5.1 million unique spam messages, reflected this issue showing trends away from adding additional emails in the CC and BCC sections in response to spam filters, and increasing the number of hops emails had to make.[3, pp. 6–7] Though time-limited and now a decade old, the 'arms race' is ongoing, shown by the estimated quadrupling of spam between 2015 and 2016.[5, p. 2] New types of spam suggests opportunities for discovering whether new spam messages form a distinct cluster from non-spam messages.

Owing to privacy concerns, generating an up-to-date corpus of believable non-spam emails is challenging. One approach attempted to subvert this by using the Enron email corpus together with databases of new spam messages.[7] This had the benefit of using real non-spam emails, which were not intended to be publicly available when first written and thus unbiased by that consideration. The Enron corpus was anglophone, however, and writing patterns for non-spam emails (e.g. idioms used) may have changed in the intervening years.

Data and Data Processing

The Spambase dataset contains 57 numerical features for 4,601 emails gathered from spam emails the dataset provider had received and emails from one's own inbox. Approximately 39% of the emails in the dataset are classified as spam.

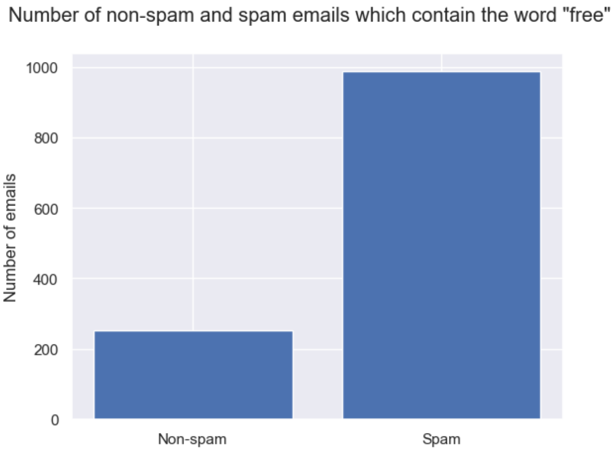
Only the target variable ('spam, non-spam classes') is categorical. All other variables are real numbers between 0 and 1, measuring the proportion of all words and characters which are a given word or character (for instance the word 'free').

No processing has been conducted for missing data. Whilst the dataset is listed as having missing values on the UCI repository, there is no direct indication in the documentation or dataset itself as to whether this is missing values for a given variable or missing samples. If there is a consistent pattern to values for variables being misclassified as 0, then there is the possibility that models trained on this data will pick up on the pattern. If there is a systematic pattern of many spam or non-spam emails having 0 values where they otherwise shouldn't, the models may learn this and become better at classification within the dataset, but not generalise beyond it. If data for additional emails are missing, then whether the impact of these missing data is significant or not is not knowable without the original emails. One possible way to correct for this is to include other spam emails from the same year to see if the model changes as a result.

Distinct groupings between the spam and non-spam emails are expected based on their differences in topics. Non-spam emails have a large amount of overlap with each other which is not shared by the spam emails, owing to the non-spam emails being from a combination of a single employee's email inbox and a set of internal company emails. For instance, the contributors of the dataset note that 'george' and '650' are clear indicators of non-spam emails (only 1% of emails which contain 'george' are spam emails; and only 2% of all emails which contain '650' are spam emails). The word 'free' also occurs in more spam emails than non-spam emails.

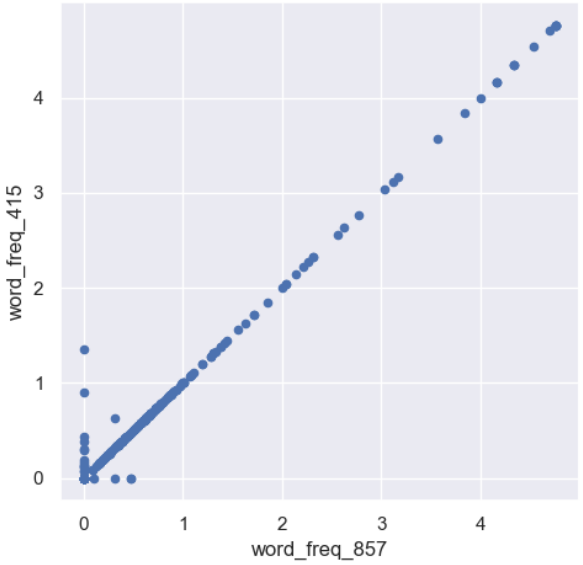
For greater generalisability, the variables for the frequency of 'george' and '650' were dropped.

The 'total', 'longest', and 'average' capital run are closely related (though not directly calculated from each other). The 'total' and 'longest' values were dropped, as the 'average' capital run is expected to capture information which the other two variables capture.

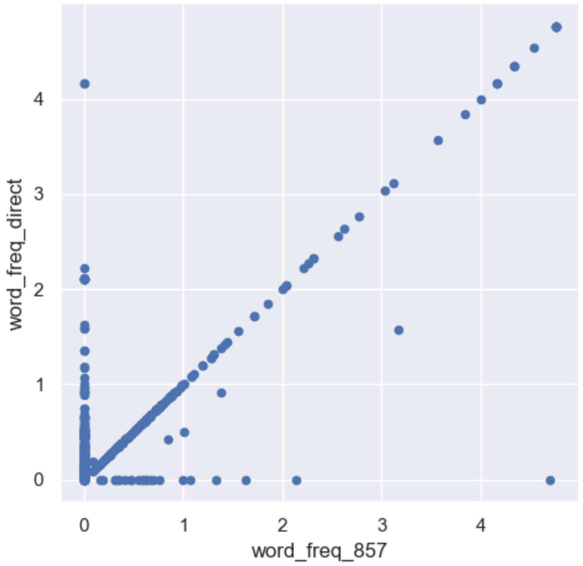


However, this also allows for multicollinearity to arise. At the most extreme 'word_freq_857' and 'word_freq_415' have a correlation coefficient of 0.996 (3 significant figures). Both of these variables cannot be included in the logistic regression model, as doing so would violate the assumption of a lack of multicollinearity between variables. For instance, many emails have the same proportions of ‘direct’ and ‘857’ in them, but there are also a lot of emails with a non-zero proportion of one of these values but zero of the other.

Scatter plot of the comparative frequencies of "857" and "415"

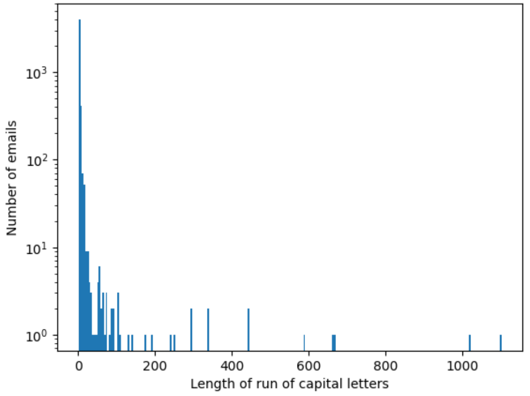


Scatter plot of the comparative frequencies of "857" and "direct"



By default, regularisation will be applied by the Scikit learn library and this will not be changed when training the model. This was due to some variables (such as the lengths of ‘runs’ of capital letters) skewing to the right, potentially allowing the furthest variables to prevent the logistic regression from converging.

Log of the number of emails by their average length of a run of capital letters



The dataset has a number of limitations. Temporally, the dataset is quite old, and therefore the investigation focuses on the development of a spam classifier for 1999 rather one which accounts for more recent types of spam. Only calculated values rather than the raw emails are included, preventing a different set of features from being developed from those currently in the dataset.

<u>Learning Methods</u>

For the specific aim of this project, supervised classification models are most useful.

Logistic regression is a binary classification algorithm which detects linear separations between instances of two classes. It is used to estimate the probability that a given instance belongs to one class or the other, predicting that that instance is class A if the probability of being so is greater than a threshold of 0.5.[8, Sec. 4. Logistic Regression] Logistic regression carries the assumptions of other linear models, namely that the input variables must not be correlated with each other (absence of multicollinearity), that the errors of variables are independent, there are no influential outliers, and that continuous variables are linear.[9, p. 1101]

Support Vector Machines are a family of regression and classification algorithms which do not have to be linear, allowing for more complicated models than logistic regression. However, unlike logistic regression, the training of support vector machines does not scale beyond hundreds of thousands of instances.[8, Ch. 5]

The main motivation behind selecting logistic regression and support vector machines was to compare the relative effectiveness of a linear-boundary model to non-linear model.

A spam identification algorithm must also classify new emails quickly, adding to the appeal of logistic regression's ability to classify a new sample in constant time. By learning a model, both logistic regression and a support vector classifier are suitable for this.

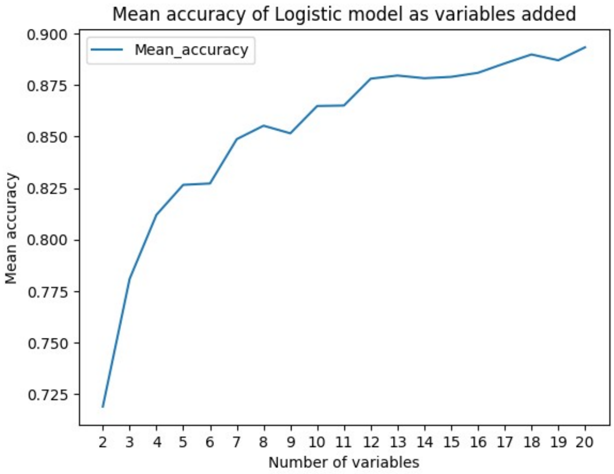
K-nearest neighbours, whilst having performed better that the support vector machine model in tests, was not selected due to scalability concerns. Unlike other classification algorithms which learn a model and can evaluate a new sample quickly, k-nearest neighbours needs to re-identify different groups, making it inappropriate for the context of quickly classifying email messages before they can (potentially) do harm. With only two suspected groupings, K-nearest neighbours may also not provide more insight than a linear model dividing two groups. Instead, it would be more useful to identifying different types of spam messages in a larger corpus where models could then be trained for those different types of spam.
Superficially, being able to identify new spam which is written in a new way (provided that spam message is still distinctive enough compared to non-spam messages) is a strength of k-nearest neighbours over the other classification algorithms. However, this is due to the algorithm effectively being retrained each time it is run, which would need to occur with model-producing algorithms anyway.

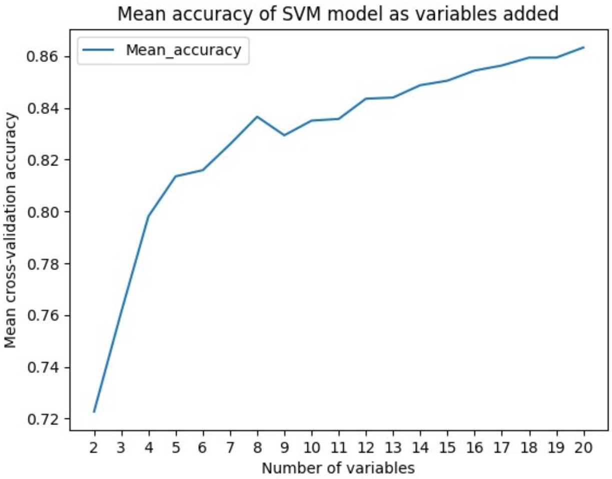
Analysis, Testing, Results

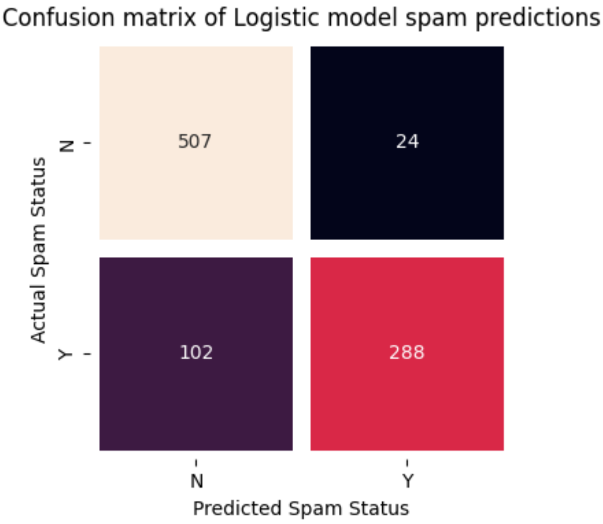
The dataset was randomly split, with sixty percent of the dataset used for training and forty percent for testing.

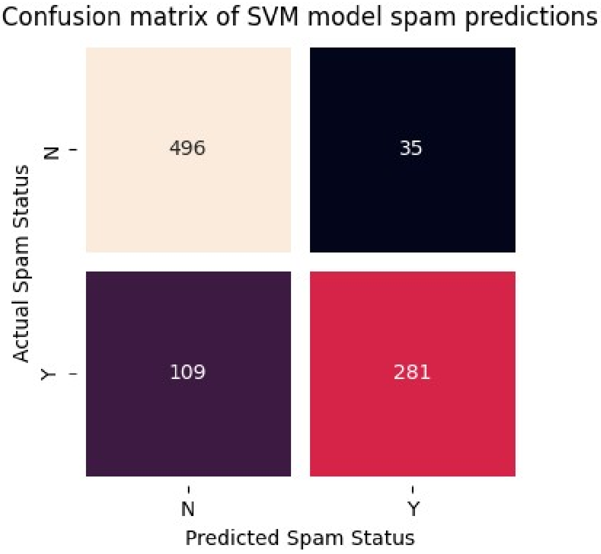
The χ^2 test was used to find the most influential features (after the removal of variables 'george', '650', as well as two of the capital runs variables) from the available variables. As all data is non-negative, the χ^2 test can be used.

The most influential features from this list were progressively added to each of the logistic and SVM models with cross validation performed on them. The first five features resulted in the greatest gains to the mean accuracy of the models. Overall accuracy increased as more variables were added, with the exception of '000', which cause a dip in accuracy as the ninth variable. This has no strong correlations with any of the other variables used, but the particular combination of variables appears inappropriate. Whilst the overall accuracy from the cross-validation scores increased, this may be due to overfitting. Therefore, the logistic and SVM models were each trained with eight variables.









For logistic regression, the mean cross-validation accuracy was 0.858, with a standard deviation of 0.013. On the testing set, the accuracy was 0.863, with a precision of 0.923 and recall of 0.738.

For support vector classification, the mean cross-validation accuracy was 0.837, with a standard deviation of 0.014. On the testing set, the accuracy was 0.844, with a precision of 0.889 and recall of 0.721.

The two models had similar performances overall, with the logistic model performing slightly better. At the scale of this dataset, the difference in performance is minor. If used for practical spam classification, the SVM model will reject a slightly higher number of non-spam emails and let through more spam emails than the logistic model, with the practical disparity potentially becoming quite large at the scale of millions of emails per day.

That both models performed slightly better on the test set than during cross-validation on the training set is possibly due to ‘luck’, and is not expected to replicate on any other testing datasets which might be used.

Conclusion

Whilst the overall accuracy of these models was encouraging, at 84-86% it is likely too low to be of practical use at the scale of billions of emails per day.

Given the accuracy was similar to that of the logistic model and based on the same features being selected for both models, it is not obvious that the trained support vector classification model was non-linear. With this particular dataset and set of features, the spam and non-spam instances may have been cleanly separated, but this might not necessarily be the case for spam emails beyond this dataset, as show by Wang, Irani and Pu’s analysis. For other datasets, particularly if the form and content of the spam emails is closer to that of non-spam emails, it is possible that the logistic regression algorithm would have worse performance overall.

Here, only the frequency of words within the content of an email was considered, ignoring multiple other features (such as email addresses in header files as well as metadata around the emails and spam being analysed) which may be of use in training a model.

Future directions for spam-classification would entail testing more models to see if their performance differs significantly and if they can be used in combination with each other. For instance, unsupervised clustering algorithms might help indicate different types of spam which could then be targeted with supervised approaches. Extending beyond spam classification, general email classification as suggest by Cohen would be the next logical step.

References:

[1] E. R. Mark Hopkins, ‘Spambase’. UCI Machine Learning Repository, 1999. doi: 10.24432/C53G6X.

[2] L. F. Cranor and B. A. LaMacchia, ‘Spam!’, *Commun. ACM*, vol. 41, no. 8, pp. 74–83, Aug. 1998, doi: 10.1145/280324.280336.

[3] D. Wang, D. Irani, and C. Pu, ‘A Study on Evolution of Email Spam Over Fifteen Years’, in *Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Austin, United States: ICST, 2013. doi: 10.4108/icst.collaboratecom.2013.254082.

[4] W. W. Cohen, ‘Learning rules that classify e-mail’, in *1996 AAAI Spring Symposium on Machine Learning in Information Access*, 1996, pp. 18–25.

[5] 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: 10.1016/j.heliyon.2019.e01802.

[6] L. Pelletier, J. Almhana, and V. Choulakian, ‘Adaptive filtering of spam’, in *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.*, Fredericton, NB, Canada: IEEE, 2004, pp. 218–224. doi: 10.1109/DNSR.2004.1344731.

[7] V. Metsis, I. Androutsopoulos, and G. Paliouras, ‘Spam Filtering with Naive Bayes – Which Naive Bayes?’, presented at the CEAS 2006 - Third Conference on Email and Anti-Spam, Mountain View, California, USA, Jul. 2006.

[8] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition*, 3rd edition. O’Reilly Media, Inc., 2022.

[9] J. C. Stoltzfus, ‘Logistic Regression: A Brief Primer: LOGISTIC REGRESSION: A BRIEF PRIMER’, *Acad. Emerg. Med.*, vol. 18, no. 10, pp. 1099–1104, Oct. 2011, doi: 10.1111/j.1553-2712.2011.01185.x.