Super Vector Machine with AI Exploration: E-mail Spam Detection with SVM

Chisel Chavush[1]

Other group members:
Alejandro Fernandez Flores,[2] Santa Sian,[3] Kathryn Elise Van Middlesworth[4]

Abstract. The internet plays a significant role in modern life. The people spend most of their time on internet.The ability to communicate online is one of the Internet's key benefits. Email is a form of communication that is operate in both personal and professional contexts. [14] Spam emails are, that the user is uncomfortable with and do not want receive. It is also called unwanted junk or bulk email.Many people use email every day to communicate with each other globally.Large numbers of spam emails are currently causing major issues for Internet users and Internet services. For instance, it hinders the spread of viruses over networks, degrades user research capabilities, and burdens network traffic.It also wastes users' time and effort by including legitimate emails among the spam. [5] In order to avoid the spam, there are many anti-spam techniques in machine learning that help us to protect ourselves. For example Naive bayes algorithm, decision tree algorithm, Support vector machines or Random forest Algorithm. These algorithms works based on the data-set that provided. In this research, we presented an effective machine learning-based spam filtering technique.[15] The technique used is Support Vector Machine Algorithm (SVM) a supervised machine learning method.In all the algorithm that we going to discuss in that paper the data set was visualised, cleaned and tokenized into a matrix of tokens.

1 Introduction

Spam can be defined as, type of email that is sent to users at one time, usually containing cryptic messages, scams and dangerous phishing contents. These emails can be send by using automated boots or from personal addresses to specific people. Spams are becoming a big scale problem. Approximately 60 percent of all internet traffic in these days is made up of spam, which has quickly become a familiar face in email users inboxes. Technically proficient individuals known as spammers are employed to send spam by businesses. The businesses attempt to avoid taking legal action by using a third party. Nowadays, numerous email servers, including Outlook, Gmail, Yahoo, Cleanfox and others, use various authentication methods while examining email content.In order to do that they are keeping a black list and a white list to classify text.[5] In Artificial Intelligence , the Machine learning methods are able to classify emails into ”spam” (unsolicited words) or ”ham” (non-spam).In its most basic form, machine learning relies on pre-programmed algorithms that take input data and analyse it in order to predict output values that fall within a predetermined range. These algorithms gain ”intelligence” over time when new data is fed into them, learning and optimizing their processes to increase performance. Machine learning algorithms come in four different varieties: supervised, semi-supervised, unsupervised and reinforcement (Figure 1). There are many algorithms to make this classification in machine learning such as Bayesian Logistic Regression (BLR), J48, Logistic Model Tree, Random Forest (RF), Naive Bayesian, Support Vector Machine (SVM), and Decision Tree Algorithm.[2] In this report we will be focusing on Support Vector Machine Algorithms and how efficient is this algorithm comparing the others. Support Vector Machine uses supervised learning method. In Supervised learning method operators provide machine learning algorithms with the necessary inputs and outputs on a known dataset. Machine learning algorithms have to figure out how to get there. Once the algorithm recognizes patterns in the data, learns from observations, and produces predictions, the operator knows the correct answer to the problem. The operator modifies the algorithm when generating forecasts. This cycle repeats until the algorithm works and performs accurately.[12]

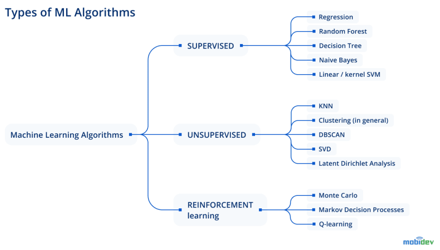


Figure 1. Types of Machine Learning Algorithms

2 Background

Machine Learning is an option to use Artificial Intelligence. It was defined in the 1950's by Artificial pioneer Arthur Samuel as ”the field of study that gives computers the ability to learn without explicitly being programmed.” [16] After then that, 28 years later on

[1] School of Engineering and Sciences, University of Greenwich, London SE10 9LS, UK, email: cc4201c@gre.ac.uk
[2] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: af0672k@gre.ac.uk
[3] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: ss8760t@gre.ac.uk
[4] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: kv3734y@gre.ac.uk

03/05/1978 the first digital spam was send by Gary Thuerk as first unsolicited bulk email. It is reported by ARPANET. That email was sent over 300 people who was working for ARPANET in order to sell the computers [7]. It wasn't until 1993 that the word ”spam” started to be actually used. It was used on a USENET message. Prior to the Internet, USENET was a newsgroup-based distributed discussion network that functions essentially as a cross between email and web forums [8]. After then those days spam is expanding at a nearly exponential rate. until it eventually makes up the great majority (80 to 85 percent) of emails sent globally. Around the world, 182.9 billion emails are sent and received every day [1]. The first spam filtering system ”SpamAssassin” was uploaded on April 20, 2001 by Justin Mason, a bit later than that in August 2002 ”A plan for spam” was published by Paul Graham. He was explaining spam filtering technique by using Bayesian filtering and variants of this [9]. Due to artificial intelligence and technology that have developed from the past to the present, the email spam detection algorithms created in these days serve many platforms.

3 Proposed Model ”SVM”

The Support Vector Machine (SVM) was chosen for this report in order to do Email Spam Detection. A supervised machine learning model called Support Vector Machine uses algorithms for classifying data into two groups. A SVM model can classify new text after being given sets of labelled training data for each category. SVM is a reliable classification technique that works quickly when with limited amount of data to analyse.SVM distinguishes classes by drawing decision boundaries (Figure 2). Drawing or determining decision boundaries is the most important part of the SVM algorithm.Each observation (or data point) is plotted in n-dimensional space prior to the creation of the decision boundary. The number of functions used is called ”n”. For instance, when categorising distinct ”cells” using ”length” and ”width,” observations are plotted in a two-dimensional space, and the decision boundary is a line [12].

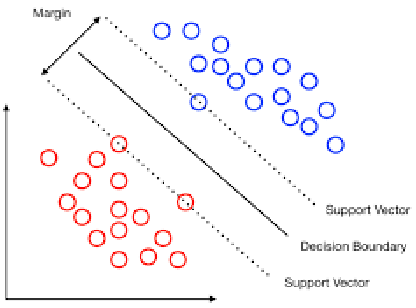


Figure 2. Support Vector Machines

Decision boundaries are drawn to maximize the distance to support vectors. If the decision boundary is too close to the support vectors, it will be very sensitive to noise and will not generalize well. Misclassification can occur even when the change in the independent variables is very small. Data points are not always linearly separable. In such cases, SVM uses a kernel trick that measures the similarity (or proximity) of points in a high-dimensional space to make them linearly separable. The kernel is a parameter that we used in SVM algorithm [11]. The kernel parameters are used as a tuning parameter to improve the classification accuracy. Also SVM has the penalty parameter which controls the trade-off between minimizing the training error and maximizing the classification margin. A kernel function is a kind of similarity. The input is the original features and the output is the similarity in the new feature space. Similarity means a degree of closeness. Actually transforming the data points into a high-dimensional feature space is an expensive operation [13]. This algorithm does not actually transform the data points into a new high-dimensional feature space. A kernelized SVM computes decision boundaries in terms of similarities in a high-dimensional feature space without actually performing any transformations. That's why it's also called the kernel trick. SVM is especially effective when the number of dimensions is greater than the number of samples [11]. When finding decision boundaries, the SVM uses a subset of the training points instead of all points, making it more memory efficient. On the other hand, large datasets increase training time, which negatively impacts performance [13].

4 Implementation and Process of SVM

In order to implement SVM into algorithm Google Colab platform has been used. That platform allows user to write and execute Python code through the browser.The platform is well suited to machine learning. For this project the data-set has been used for ”ham” and ”spam” detection, data-set is taken from ”Kaggle.com”, which is an open source website. The dataset was created by Venkatesh Garnepudi [4]. There are 4 columns in total in the data set.The first column is not named and contains numerical expressions. The second column is named ”label” and indicates that the text given in the 3rd column is ham or spam. The name of the 3rd column is ”text”. All the texts given in this section are written in raw form, and the last column called ”label num” where 0 represents ham message and 1 represents spam messages. In order to start to create the algorithm we need to implement required libraries. For spam detection different libraries need to be used, but the most important one is sklearn library which is the machine learning library. Also, numpy (works with numerical data), matplotlib.pyplot (for graphics), pandas (for reading and processing the given data), nltk (for tokenizing), and string (to process standard Python strings) has been used as a library. Once the required libraries are imported, than data-set needs to be import as well. In order to read data-set which is .csv file and process it, the pandas library has been used. Data-set totally has 4 column where 2 of them are unnecessary for the algorithm, to clean those columns again pandas library is used. Once data has been analysed through the algorithm the total number of emails is 5171(Figure 3).

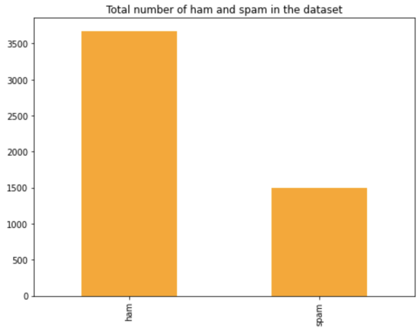
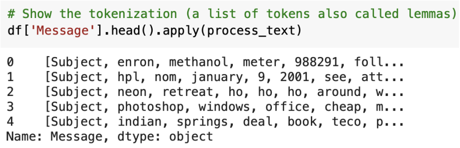


Figure 3. Total number of ham and spam in the data-set

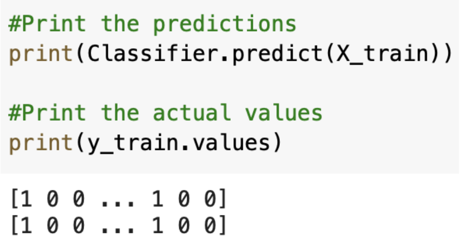
In the next step in order to clean null and and repeated words ”str.punctuation” and ”stopwords” package has been imported. Once the data has been cleaned fully we tokenize the data into a ”bag of words” [10]. The bag of words mean the grammar is totally disregarded and order of the words. Each independent word is taken as its own object and quarreled with a probability value of whather is spam or ham. (Figure 4).



```
# Show the tokenization (a list of tokens also called lemmas)
df['Message'].head().apply(process_text)

0    [Subject, enron, methanol, meter, 988291, foll...
1    [Subject, hpl, nom, january, 9, 2001, see, att...
2    [Subject, neon, retreat, ho, ho, ho, around, w...
3    [Subject, photoshop, windows, office, cheap, m...
4    [Subject, indian, springs, deal, book, teco, p...
Name: Message, dtype: object
```

Figure 4. Tokenizing the data

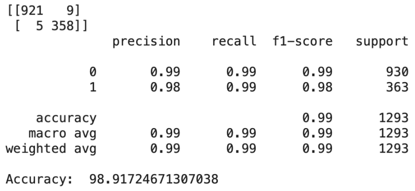
When the data tokenization is complete, we are converting a collection of text to a matrix of token with ”TfidfVectorizer”. That converts the text into a feature vector that can be used as input to the estimator. Tokenized data’s gives more accurate results. The last part of the pre-processing step is the splitting the data into 80 percent training and 20 percent testing by using sklearn library. In order to do that we imported ”train-test-split” method, by doing that we can reduce the impact of data discrepancies and comprehend the model’s properties better. Once all this pre-proccess is complete we can start to create SVM classifier and training the model on the training data and labels. Also the program using the model, to predict the labels of the test data. When the program print the predictions and actual values the visible data’s seems equal (Figure 5).



```
#Print the predictions
print(Classifier.predict(X_train))

#Print the actual values
print(y_train.values)

[1 0 0 ... 1 0 0]
[1 0 0 ... 1 0 0]
```

Figure 5. Predictions and Actual value

The performance analyse is based on Confusion Matrix. The confusion matrix is a table that is defining the performance of a classification algorithm. This table includes 4 different combinations of predicted and actual values which is TP (True Positive), TN (True Negative), FN (False Negative), FP (False Positive). Performance based on TP (True Positive) TN (True Negative). In here TP represents that the prediction is positive and it is true and TN is that prediction is negative and it is true [6]. In this case our program has 921 TN, 9 FP, 5 FN, and 358 TP. If we need to calculate accuracy, the formula is (True Positive)+(True Negative) / Total in our case that is going to be (921 + 358)/1293 = 98.9172 [figure 6]. The accuracy score, out of 1278 test instances, the algorithm misclassified 14. Which is really efficient and accurate for determining the text was spam or ham comparing to other algorithms and classifiers. Also there is F1 score in classification report. F1 score is a weighted average of precision and recall.False positive and false negative results can occur in accuracy and recall, as is well known, thus both are taken into consideration. In most cases, the F1 score is more helpful than accuracy, particularly if your class is distributed unevenly. When false positives and false negatives cost about the same, accuracy performs best. It is preferable to include both Precision and Recall if the costs of false positives and false negatives are significantly different [3].

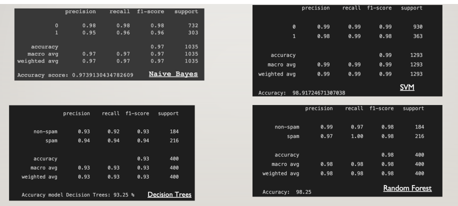


```
[[921   9]
 [  5 358]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       930
           1       0.98      0.99      0.98       363

    accuracy                           0.99      1293
   macro avg       0.99      0.99      0.99      1293
weighted avg       0.99      0.99      0.99      1293

Accuracy:  98.91724671307038
```

Figure 6. Classification report and Confusion Matrix

5 Discussion

When we compare the other algorithms that we performed in our project Random Forest had the accuracy score 98.25 where precision is higher than recall.It is the second highest accuracy score. The Naive Bayes provided 97.39 accuracy which is also good. The last one is the Decision Tree which not accurate as the other ones but also the accuracy score is 93.25 for Decision Tree. In summary, Support Vector Machine achieved the highest accuracy score with 98.91 proving that SVM performs the most accurate email spam detection among the algorithms tested.[figure 7]. Advantages of the SVM’s are relatively memory efficient comparing the others. Also, it works relatively well when there is a clear margin of seperation between the classes as we have been doing. In the other hand there is a disadvantages of the SVM among the other algorithms. SVM algorithm is not suitable for large datasets. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform. As all the algorithm there will be some positive and negative affects. In our case as we have limited amount of data in our datasets and they all categorised. That’s why SVM is working properly and accurate in our program.



```
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       732
           1       0.95      0.96      0.96       303

    accuracy                           0.97      1035
   macro avg       0.97      0.97      0.97      1035
weighted avg       0.97      0.97      0.97      1035

Accuracy score: 0.9739130434782609
```

Naive Bayes

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       930
           1       0.98      0.99      0.98       363

    accuracy                           0.99      1293
   macro avg       0.99      0.99      0.99      1293
weighted avg       0.99      0.99      0.99      1293

Accuracy:  98.91724671307038
```

SVM

```
              precision    recall  f1-score   support

    non-spam       0.93      0.92      0.93       184
        spam       0.94      0.94      0.94       216

    accuracy                           0.93       400
   macro avg       0.93      0.93      0.93       400
weighted avg       0.93      0.93      0.93       400

Accuracy model Decision Trees: 93.25 %
```

Decision Trees

```
              precision    recall  f1-score   support

    non-spam       0.99      0.97      0.98       184
        spam       0.97      1.00      0.98       216

    accuracy                           0.98       400
   macro avg       0.98      0.98      0.98       400
weighted avg       0.98      0.98      0.98       400

Accuracy:  98.25
```

Random Forest

Figure 7. Comparing Classification report and Confusion Matrix

6 Conclusion and future work

The ability to communicate online is one of the Internet’s key benefits. Email is a form of communication that operate in both personal and professional contexts. Large numbers of spam emails are causing major issues for Internet users and Internet services. There are many anti-spam techniques in machine learning that help us to protect ourselves.In this report, we have explained how spam can be prevented in digital environment and with which methods it can be used.The parameters used in the algorithm and their properties are detailed. Also, the report observed how the word tokenization could perform better by using libraries. As a feature SVM can be improved and perform better by using equal amount of data for Spam and Ham.

ACKNOWLEDGEMENTS

I want to thank my team for giving me a great working experience as a team. Also I would like to thank to our Lecturers and tutor for providing us wealth of information and resources in this area, it was an amazing experience to starting to AI.

REFERENCES

[1] Anugya Asthana, 'Social media and the quality of life of women', *Journal of Media and Communication*, **4**(2), 40–56.

[2] Giuseppe Bonaccorso, *Machine learning algorithms*, Packt Publishing Ltd, 2017.

[3] Davide Chicco and Giuseppe Jurman, 'The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation', *BMC genomics*, **21**(1), 1–13, (2020).

[4] Gordon V Cormack, Mark D Smucker, and Charles LA Clarke, 'Efficient and effective spam filtering and re-ranking for large web datasets', *Information retrieval*, **14**(5), 441–465, (2011).

[5] Lorrie Faith Cranor and Brian A LaMacchia, 'Spam!', *Communications of the ACM*, **41**(8), 74–83, (1998).

[6] Xinyang Deng, Qi Liu, Yong Deng, and Sankaran Mahadevan, 'An improved method to construct basic probability assignment based on the confusion matrix for classification problem', *Information Sciences*, **340**, 250–261, (2016).

[7] Steve Hedley, 'A brief history of spam', *Information & Communications Technology Law*, **15**(3), 223–238, (2006).

[8] Barry Leiba and Nathaniel S Borenstein, 'A multifaceted approach to spam reduction.', in *CEAS*, (2004).

[9] Qin Luo, Bin Liu, Junhua Yan, and Zhongyue He, 'Design and implement a rule-based spam filtering system using neural network', in *2011 International Conference on Computational and Information Sciences*, pp. 398–401. IEEE, (2011).

[10] Ulf T Mattsson, 'A new scalable approach to data tokenization', *Available at SSRN 1627284*, (2010).

[11] David Meyer and FT Wien, 'Support vector machines', *The Interface to libsvm in package e1071*, **28**, 20, (2015).

[12] William S Noble, 'What is a support vector machine?', *Nature biotechnology*, **24**(12), 1565–1567, (2006).

[13] Derek A Pisner and David M Schnyer, 'Support vector machine', in *Machine learning*, 101–121, Elsevier, (2020).

[14] Gary Taubes, 'Publication by electronic mail takes physics by storm', *Science*, **259**(5099), 1246–1248, (1993).

[15] Konstantin Tretyakov, 'Machine learning techniques in spam filtering', in *Data Mining Problem-oriented Seminar, MTAT*, volume 3, pp. 60–79. Citeseer, (2004).

[16] Gio Wiederhold and John McCarthy, 'Arthur samuel: Pioneer in machine learning', *IBM Journal of Research and Development*, **36**(3), 329–331, (1992).