University of Ibadan
Journal of Science and Logics in ICT Research

Personalized Email Prioritization Using Multi-attribute and Multi-classification Data Mining Techniques

Adebomi Adebimpe Esther
adebimpeadebomie@gmail.com

Adeyemo Adesesan Barnabas.
sesanadeyemo@gmail.com

Department of Computer Science, University of Ibadan, Ibadan, Nigeria

Abstract

The increase in the volume of electronic email communication that is received daily by an individual is becoming alarming and it threatens to cause a state of "email-overload" where the volume of messages exceeds individual capacity to process. With email being one of the most efficient and effective mode of communication that is widely used among business personnel and organizations, there is need to pay apt attention to the serious problem of email information overload that pose serious productivity challenges for busy professionals and executives. This necessitated the adoption of Data mining techniques to develop a model for prioritizing email using a multi-attribute and multi-classification algorithm for the automatic classification of mails into predefined categories while eliminating the problem of manual labeling or annotation from users (an approach that is tedious and time consuming for users) in previous research work [1]. This study was introduced to automatically classify and prioritize email messages into folder structures, in a declining order of importance according to the priorities of each user's email inbox content, without manual labeling or annotation of email categories from users. This model extends the application of K-means, Hierarchical Clustering and SVM classifier to the domain of email prioritization. The model developed, when used, eliminates the traditional manual labeling/annotation and method of triaging through a large volume of incoming email in no particular order and introduces a well-structured and organized hierarchy and priority level in each user personalized email categories.

Keywords- Email Prioritization, Email-Overload, Data Mining, Machine Learning

1. INTRODUCTION

Email is a more personal and direct way of communication in which Messages is delivered within seconds around the world [2]. Email is important because it creates a fast, reliable form of communication that is free and easily accessible. Email allows people to foster long-lasting, long-distance communication. It is not characterized by the inconveniences that are generally associated with traditional communication media, such as telephone or postal mail. But together with a blessing comes a curse, with 2.3 billion users worldwide and over 205 billion emails sent or received every day [3], users mailbox's may get flooded with large volumes of emails periodically making it difficult for a user to access the most important ones; thereby telling on the users productivity and work morale.

Based on previous research, 58% of emails are irrelevant or unimportant and a person on average has to waste at least one hour per day to handle them [4]. Therefore this issue of overload needs to be addressed, and adequate email management solution needs to be proffered for individuals and Business organizations to avoid wasting time and energy on irrelevant mails. There is therefore a need for an effective email prioritization and classification method for a timely requirement.
Prioritization is the process of regarding one email message as being more important than others (Merriam Webster Dictionary, 2018). Existing email filtering and prioritization use classification. Classification is a data mining technique and is defined as discovering useful knowledge from large data [5]. Classification is the process of finding a model that describes and distinguishes different classes or concepts of data. Classification mainly consists of two steps. First

Adebomi A. E. and Adeyemo A. B. (2020). Personalized Email Prioritization Using Multi-attribute and Multi-classification Data Mining Techniques, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 4 No. 1, pp. 74 - 84

©U IJSLICTR Vol. 4, No. 1, March 2020

is the learning step: where a classification model is constructed and second is the classification step: in this step the extracted model is used to predict the class labels for new data or unknown data depending on the learning step [5].

Data mining algorithms are used for classification of objects of different classes. Such algorithms have proved to be efficient in classifying and prioritizing emails as important or not important. In this study, machine learning algorithms namely, Hierarchical, K-means Clustering and Support Vector Machine (SVM). Gmail Inbox message and Enron Email Corpus downloaded online were used for the research. It is one of the publicly available large datasets of email. The tool used for the application development is Python 3 programming Library and packages. Therefore a good way to alleviate email overload is to automatically prioritize (i.e. rank) received messages according to the priorities of each user [6].

2. Literature Review

Electronic mail (email or e-mail) is a method of exchanging messages ("mail") between people using electronic devices. Email first entered limited use in the 1960s and by the mid-1970s had taken the form now recognized as email. Email operates across computer networks, which today is primarily the Internet. Some early email systems required the author and the recipient to both be online at the same time, in common with instant messaging (Wikipedia, 2018). Today's email systems are based on a store-and-forward model. Email servers accept, forward, deliver, and store messages. Neither the users nor their computers are required to be online simultaneously; they need to connect only briefly, typically to a mail server or a webmail interface, for as long as it takes to send or receive messages.

Originally an ASCII text-only communications medium, Internet email was extended by Multipurpose Internet Mail Extensions (MIME) to carry text in other character sets and multimedia content attachments. International email, with internationalized email addresses using UTF-8, has been standardized, but as of 2017 it has not been widely adopted (Wikipedia, 2018).

The history of modern Internet email services reaches back to the early ARPANET, with standards for encoding email messages published as early as 1973 (RFC 561). An email message sent in the early 1970s looks very similar to a basic email sent today (Wikipedia, 2018). Email had an important role in creating the Internet [7], and the conversion from ARPANET to the Internet in the early 1980s produced the core of the current services. The history of email extends over more than 50 years, entailing an evolving set of technologies and standards that culminated in the email systems we use today.

Computer-based mail and messaging became possible with the advent of time-sharing computers in the early 1960s, and informal methods of using shared files to pass messages were soon expanded into the first mail systems (Wikipedia, 2018). Most developers of early mainframes and minicomputers developed similar, but generally incompatible, mail applications. Over time, a complex web of gateways and routing systems linked many of them. Many US universities were part of the ARPANET, which aimed at software portability between its systems; that portability helped make the Simple Mail Transfer Protocol (SMTP) increasingly influential (Wikipedia, 2018).

For a time in the late 1980s and early 1990s, it seemed likely that either a proprietary commercial system or the X.400 email system, part of the Government Open Systems Interconnection Profile (GOSIP), would predominate. However, once the final restrictions on carrying commercial traffic over the Internet ended in 1995 [8], a combination of factors made the current Internet suite of SMTP, POP3 and IMAP email protocols the standard.

2.1 Email message format

The email message format is defined in RFC (Request for Comments) 5322 (released in October 2008) and in some additional RFCs from 2045 to 2049. Collectively, these RFCs are called Multipurpose Internet Mail Extensions, or in short MIME. An email message consists of two major sections:

1. Header contains information about the sender, receiver, subject, date, etc.
2. Body is the message itself as text and is the same as the body of a regular letter; The main fields of an email header are:
    i. Date: time of sending out the email message

ii. From: usually the author of the email
iii. To: one or many recipients of email
iv. Cc: recipients who are not directly related to message but may be interested in the information containing in email
v. Bcc: recipients is field that will remain invisible to other addressees
vi. Subject: a short summary of the contents of email

All these fields contain valuable information to classify an email message. Information in the email body, threads and an attachment can also be used; the email body can be written in plain text or in HTML.

2.2 Message overload problem

Every day, more and more emails are sent and received by users as we depend increasingly on email communication. Messages are not only received from friends or colleagues but also from all kind of social networks and advertising companies. Whittaker and Sidner [9] Pointed out that email is also used for document delivery, sending reminders, scheduling an appointment which shows that email is used for a variety of purposes exceeding its original design as a simple communication application. Organizing this flow is far beyond filtering spam into the Junk folder by different spam filters and the amount of email messages in our Inbox keeps increasing. At some point when we realize that it is not necessary to delete any emails as the capacity of any email account is enough to store hundreds of thousands emails we face a problem where it is almost impossible to find information needed as the number of emails in our inbox keeps growing. In this situation we have created a huge and very chaotic list of email mainly sorted by the date received [9]. Received emails often contain information which is not needed at the time of getting the email. In a situation like this the message is skipped and there is a real danger that this message will get “lost” or overlooked in the increasing amount of emails [9].

2.3 Related Works

Data mining algorithms are used for classification of objects of different classes. Such algorithms have proved to be efficient in classifying and prioritizing emails as important or not important. In this study machine learning algorithms namely, Hierarchical, K-means Clustering and Support Vector Machine (SVM) were used. Gmail Inbox message and Enron Email Corpus downloaded online were used for the research; it is one of the publicly available large datasets on email. Python 3 programming Library and packages tools were used for the application development. A good way to alleviate email overload is to automatically prioritize (i.e. rank) received messages according to the priorities of each user.

Machine Learning techniques have been applied to email-overload issues and many email classification and prioritization techniques have been proposed by several authors to alleviate the email-overload problem. Some works in the literature includes that of Wang, *et.al.*, [10], which introduced the problem of personalized broadcast email prioritization considering large numbers of mailing lists and proposed a novel cross domain recommendation framework CBEP (Cross-Domain Broadcast Email Prioritization) to solve the problem. To select the optimal set of source domains from the large number of domains, they proposed an optimization model that considers multiple selection criteria including the overlap of users, feedback pattern similarity and coverage of users. A weighted low-rank approximation method is proposed to make predictions based on information from both the target domain and the selected source domains using Bayesian Theorem classified email into categories [1**]**. The classifier was trained to recognize attributes for each category. When a new mail arrives, it compares the attributes of the mail with attributes of each category and the mail is classified according to the category having most similar attributes as that of the mail.

Aberdeen *et. al.*, [11] Proposed a simple linear logistic regression model for mail prioritization in Gmail. The final prediction is the sum of the global model and the user model log odds. Four categories of features are considered in the model, including social features, content features, thread features and label features. Johansen [12] and her colleagues used social clustering to predict the importance of email messages. The major difference between their methods is that their clusters were induced from a community social network, not based on personal social networks or the content information in email messages.

3. METHODOLOGY

At First datasets that contain 100 rows by 11 columns of email inbox messages that was extracted from the author's Gmail account into an excel format was used alongside 10,000 datasets extracted from online Enron-Corpus of about 500,000 dataset available on www.cs.cmu.edu/enron/ Therefore a total of 10,100 data samples were used for this study, from which different data samples sizes were extracted from it, in a 70% & 30% train/test split.

The complete 10,100 data was not used at once, the algorithm was built to pull data of variant data samples quantity from this dataset, so as to have a detailed view of how the algorithm behaved on different data sample sizes. The Data contain 11 columns namely: Message ID, Serial number, Date (year, month and day), To (describes the recipient of a user's message), From (the sender), CC (others included in a message apart from the main recipient), BCC (other hidden recipients included in a message apart from the main recipient), Subject (message topic), Body (The content of the message i.e. the main message), Reply to (messages the recipient responded to leading to a thread), Attachment name (messages that contains attached document or files).

The datasets was used to give a localized analysis of the performance of machine learning algorithm on real life dataset that will be easy to relate with by users at large. The Enron corpus was used to give a globalized analysis of the performance of the algorithms on online datasets that has undergone some restructuring. It enabled the SVM classifier to fit the model appropriately and give an accurate performance working on larger datasets. The dataset was in text format with each column containing strings of text.

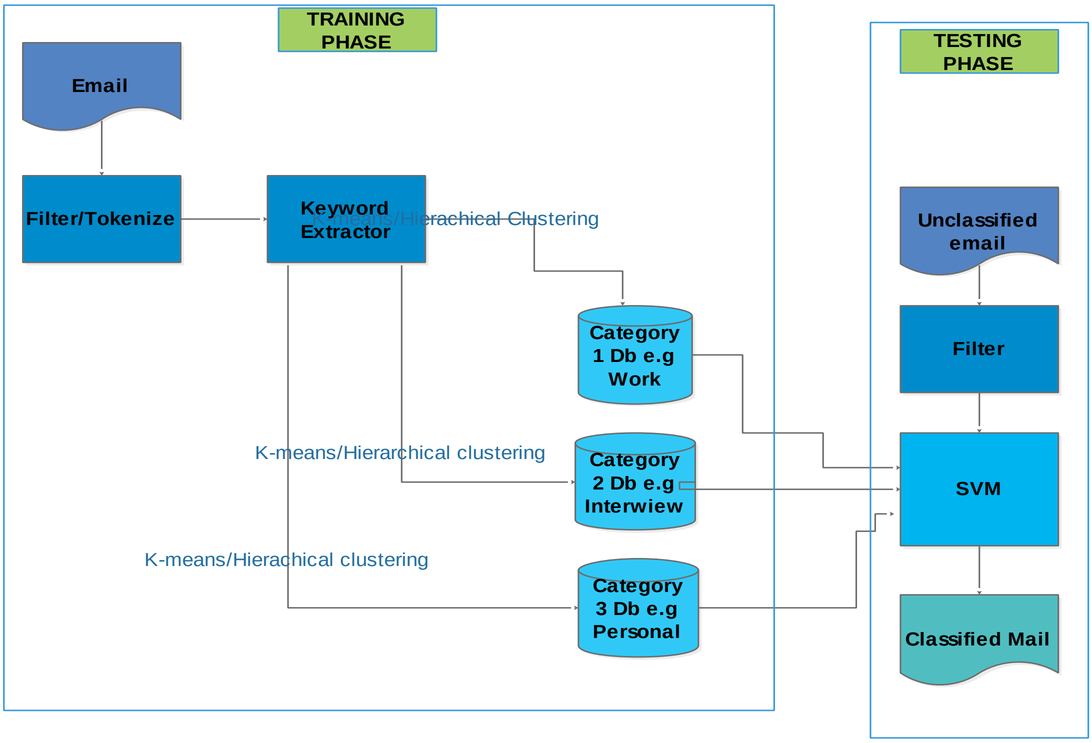

TRAINING PHASE
Email
Filter/Tokenize
Keyword Extractor
K-means/Hierachical Clustering
Category 1 Db e.g Work
K-means/Hierarchical clustering
Category 2 Db e.g Interwiew
K-means/Hierachical clustering
Category 3 Db e.g Personal
TESTING PHASE
Unclassified email
Filter
SVM
Classified Mail

Figure 1: Model Diagram

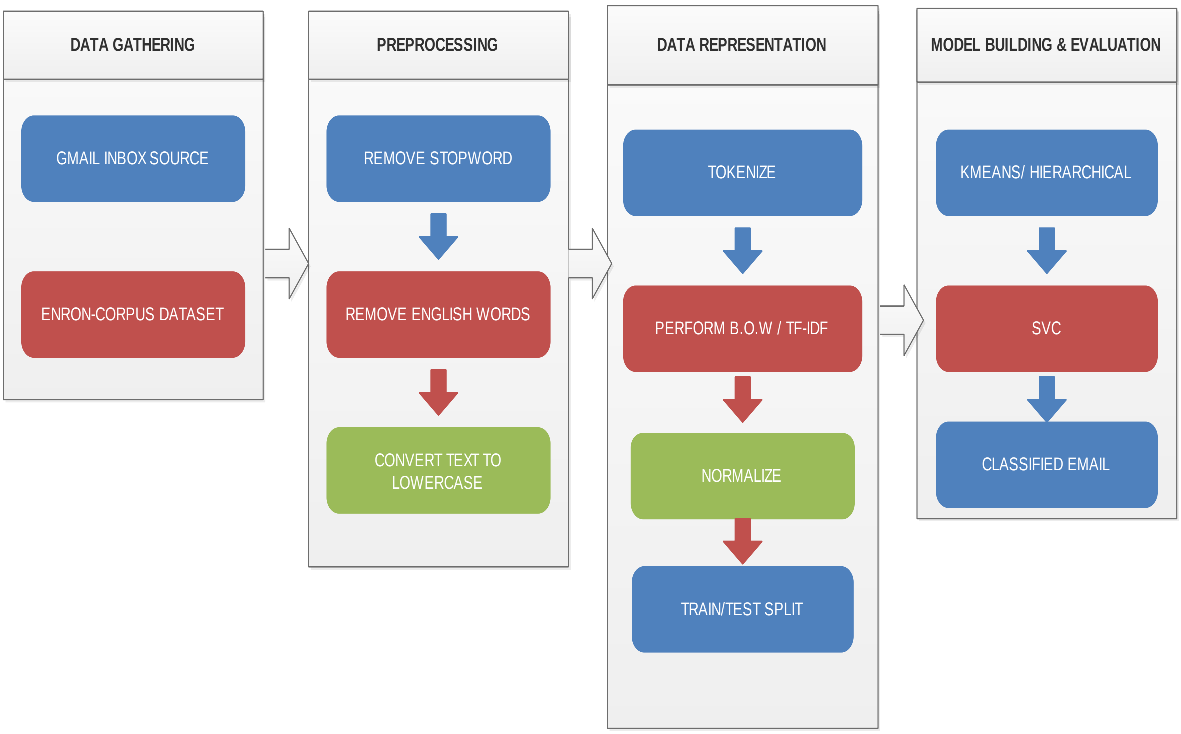


Figure 2: Model Development process

The email classification process is divided into four (4) phases namely: The preprocessing phase, the Training and Learning phase, the Classification phase and the Ranking and Prioritization phase.

3.1 Phase one: Preprocessing Phase

Here the less informative and noisy terms in the message body were eliminated to lower the feature space dimensionality and enhance the classifier performance. Stop word removal, Lemmatization, Tokenizing and Normalization was performed on the data source using the NLTK (Natural Language Tokenized) library available in python 3 programming language. The tokens are separated by blank space; proverbs, articles, html tags, noise words and other unnecessary contents which were removed and keywords were extracted. The tokens generated were then used to build a keyword of database using the bag of words techniques and the frequency of each key term were calculated using TF-IDF techniques.

3.2 Phase Two: The Training/Learning Phase

Phase: In this phase the K-Means and Hierarchical clustering (HC) algorithm were trained to recognize attributes for each category, from the generated tokens that were extracted from email messages in the preprocessed phase. The tokens were then used to classify emails into different categories based on their attributes. When a new mail arrives, it compares the attributes of the mail with attributes of each category and the mail is classified into the category having most similar attributes as that of the mail. To build the attribute list (also referred as keywords database) for each category, the emails were classified by k-means and HC algorithm into different categories or folders for the user. For this phase, three categories for emails which are, work, interview and personal was created based on the cluster generated from 100 email samples from a Gmail account. The SVM classifier was then made to learn the categories (i.e. labels) created by the k-means and HC algorithm. This was carried out using the Scikit learns library available in python 3 programming language, where the SVC classifier, k-means and HC were imported alongside a train/test split.

3.3 Phase Three: The Classification Phase

After a successful learning process by the classifiers, the new unclassified email that arrives would be automatically assigned their categories or labels. Basically, comparison was performed on the contents of the mail with each category having a database of keywords. SVC algorithm was then used to determine the best matching category for the mail. The new incoming mail (also referred to as unclassified mail) was broken into tokens and filtered. The tokens were then compared with keyword databases of each category. The hyper-plane was used to determine the category to which a mail belongs to and this was carried out for each email to find their membership for each category. The category for which the mail falls in the optimal labeling and the optimal hyper plane were checked and the mail becomes classified else the emails stays unclassified.

3.4 Phase Four: The Ranking and Prioritization Phase

The emails in each category, labels or folders were then ranked in their order of priority using Python function that was built with the following conditions: Outbound (i.e. the email message with the highest number recipients from a sender within the space of one week), Inbound (i.e. the email message with highest term frequency), Thread (the message with the highest number of threads) and date and time of Arrival

3.5 Classification Rule and Algorithm for Each Phase

3.5.1 Preprocessing phase:

1. Load the data source in .xlxs or.tsv format
2. Import nltk library
3. Import/call the stop words method from the library above to filter out stop words such as html tags, articles, punctuations and noise words
4. Divide the mail into tokens (both body and subject)
5. Extract keywords and store them alongside frequency count using the Tf-IDF transformer from sklearn.

3.5.2 Training /Learning

1. For each email, specify its category using kmeans algorithm
2. From Sklearn import kmeans to label and categorize mail into clusters

Kmeans algorithm model **[**13**]** [14] Decision steps:

Step 1:

Begin with a decision on the value of k = number of clusters (I.e. Folders = 3).

Step 2:

Put any initial partition that classifies the data into k clusters. The training samples are assigned randomly, or systematically as the following:

i. Take the first k training sample as single-element clusters
ii. Assign each of the remaining (N-k) training samples to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

Step 3:

Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Step 4:

Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

3.6 Classification Phase

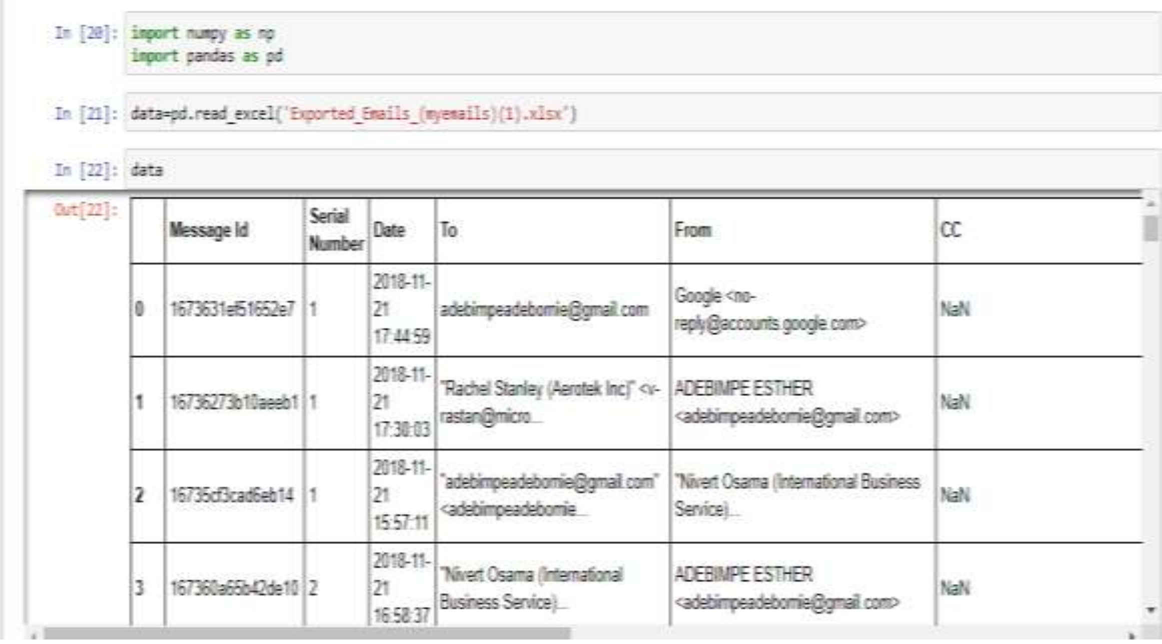
1. For each newly arrived/ unclassified email, divide the mail into set of tokens (consider both the subject and the body)
2. Filter out stop words such as html tags, articles, punctuations and noise words and extract the keywords.
3. Load vector output label generated by K-Means and Hierarchical algorithm.
4. From Sklearn.svm import SVC to learn category in step 3 above.

SVC algorithm model Decision steps [15]:

```
i.    Initialize $y_i = Y_I$ for i ∈ I
ii.   Repeat
        Compute SVM solution w, b for data set with imputed labels
        Compute outputs $f_i = (w, x_i) + b$ for all xi in positive bags
        Set $y_i = sgn(f_i)$ for every i ∈ I, $Y_I$ = 1
        FOR (every positive bag BI)
            IF ( $\sum_{i \in I} (1 + y_i) /2 == 0$)
                i.  Compute $i *= argmax_{i \in I} f_i$
                ii. Set $y_i * = 1$
            END
        END
iii.  WHILE (imputed labels have changed)
iv.   OUTPUT (w, b)
```

4. RESULTS AND DISCUSSIONS

The modeling of the email prioritization and classification system was carried out using the first 100 dataset downloaded from the author's Gmail account. Also another set of different data samples sizes from Enron Corpus downloaded online was also used. This is presented in the Figures 3 and 4, respectively. The dataset was divided into 70% for training set and 30% for test set. The split was carried out randomly. A total of 10,100 emails were used for the purpose of this research. The Pandas Library was used to load the two datasets.

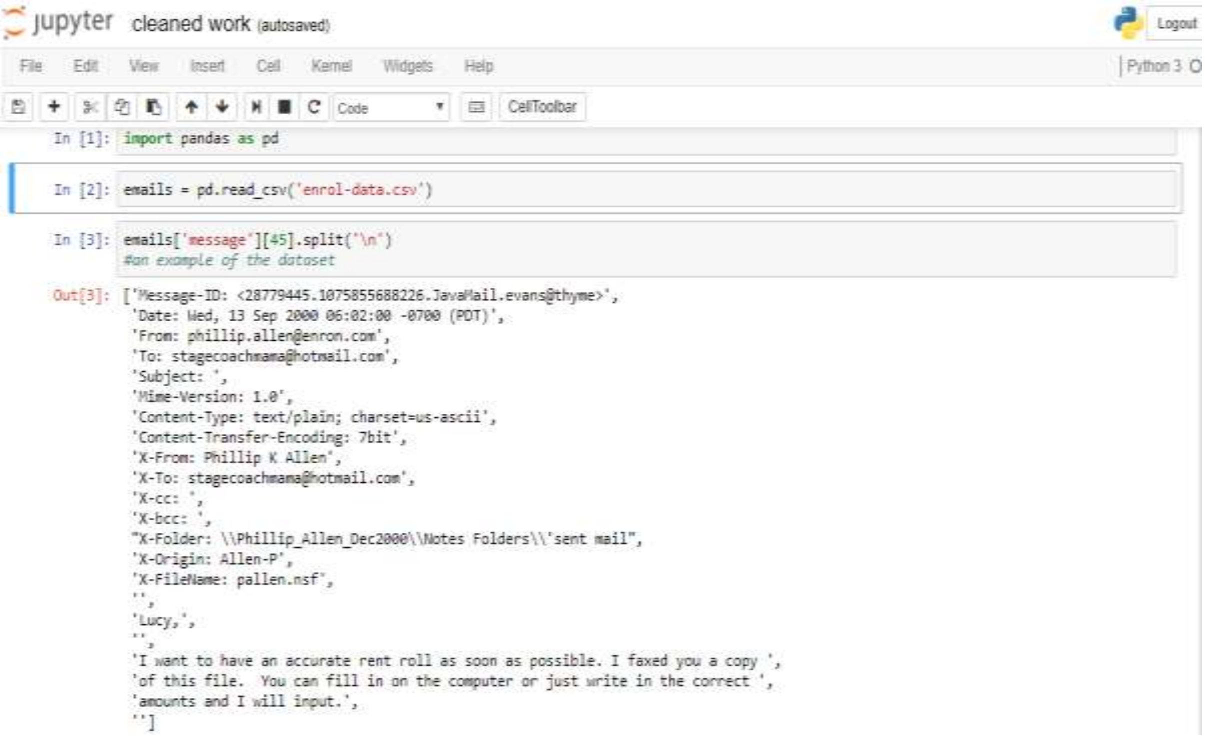


```
In [20]: import numpy as np
         import pandas as pd

In [21]: data=pd.read_excel('Exported_Emails_(myemails)(1).xlsx')

In [22]: data
```

Out[22]:

	Message Id	Serial Number	Date	To	From	CC
0	1673631ef51652e7	1	2018-11-21 17:44:59	adebimpeadebomie@gmail.com	Google <no-reply@accounts.google.com>	NaN
1	16736273b10aeeb1	1	2018-11-21 17:30:03	"Rachel Stanley (Aerotek Inc)" <v-rastan@micro...	ADEBIMPE ESTHER <adebimpeadebomie@gmail.com>	NaN
2	16735cf3cad6eb14	1	2018-11-21 15:57:11	"adebimpeadebomie@gmail.com" <adebimpeadebomie...	"Nivert Osama (International Business Service)...	NaN
3	167360a65b42de10	2	2018-11-21 16:58:37	"Nivert Osama (International Business Service)...	ADEBIMPE ESTHER <adebimpeadebomie@gmail.com>	NaN

Figure 3 Sample Dataset from Gmail



```
In [1]: import pandas as pd

In [2]: emails = pd.read_csv('enrol-data.csv')

In [3]: emails['message'][45].split('\n')
        #an example of the dataset

Out[3]: ['Message-ID: <28779445.1075855688226.JavaMail.evans@thyme>',
         'Date: Wed, 13 Sep 2000 06:02:00 -0700 (PDT)',
         'From: phillip.allen@enron.com',
         'To: stagecoachmama@hotmail.com',
         'Subject: ',
         'Mime-Version: 1.0',
         'Content-Type: text/plain; charset=us-ascii',
         'Content-Transfer-Encoding: 7bit',
         'X-From: Phillip K Allen',
         'X-To: stagecoachmama@hotmail.com',
         'X-cc: ',
         'X-bcc: ',
         "X-Folder: \\Phillip_Allen_Dec2000\\Notes Folders\\'sent mail",
         'X-Origin: Allen-P',
         'X-FileName: pallen.nsf',
         '',
         'Lucy,',
         '',
         'I want to have an accurate rent roll as soon as possible. I faxed you a copy ',
         'of this file.  You can fill in on the computer or just write in the correct ',
         'amounts and I will input.',
         '']
```

Figure 4 Sample data from Enron-Corpus Dataset

The performance of the algorithm in the categorization and classification phase was measured in terms of Classification report, Confusion Matrix, Precision, Recall, F1 score and Accuracy. Four methods were used to check if the predictions were right or wrong:

i. **TN / True Negative:** case was negative and predicted negative

ii. **TP / True Positive**: case was positive and predicted positive

iii. **FN / False Negative:** case was positive but predicted negative

iv. **FP / False Positive**: case was negative but predicted positive

v. **Precision:** show the percent of the predictions that were correct.

Precision – Accuracy of positive predictions.

Precision = TP/ (TP + FP)

vi. **Recall:** show the percent of the positive cases that was catch that is the Fraction of positives that were correctly identified.

Recall = TP/ (TP+FN)

vii. **F1 Score:** The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. The weighted **average of** F1 should be used to compare classifier models, not global accuracy.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

viii. **Accuracy:** this refers to the overall correctness of the classifier.

Accuracy = (TP+TN)/total support

The classification accuracy is dependent on several parameters such as:

- Number of Input Size (Datasets size)
- Number of Clusters
- Total number of emails in considered during the training phase.

Only two clusters were created at first, then the cluster was then increased gradually and the process was repeated for different combination of parameters. The results were averaged over ten iterations and the best performing configuration for the algorithm was then chosen.

Table 1 Performance result of Email Classifier for K-Means and SVM

Size of Training data	No. of categories	Accuracy	Precision	Recall	F1 Score
100	2	0.833	0.96	0.83	0.76
200	2	0.983	0.98	0.98	0.98
600	2	0.977	0.98	0.98	0.98
1000	2	0.993	0.98	0.98	0.98
1000	3	0.950	0.95	0.95	0.95
200	3	0.850	0.88	0.85	0.81
600	**3**	**0.991**	**0.91**	**0.90**	**0.90**
600	4	0.882	0.89	0.88	0.87
200	4	0.666	0.68	0.67	0.65
600	5	0.838	0.87	0.84	0.82
1000	5	0.863	0.87	0.86	0.86

Table 2 Performance result of Email Classifier for HC and SVC

Size of Training data	No. of categories	Accuracy	Precision	Recall	F1 Score
50	4	0.466	0.40	0.47	0.42
200	3	0.18	0.34	0.45	0.39
100	3	0.40	0.54	0.40	0.26
600	5	0.487	0.45	0.54	0.47
RESULT FOR KMEANS + HC + SVC					
200	2	0.866	0.86	0.87	0.86
600	2	0.977	0.98	0.98	0.98
600	**3**	**0.822**	**0.82**	**0.82**	**0.82**
200	5	0.85	0.88	0.85	0.81
600	5	0.77	0.80	0.77	0.75

Classification report and confusion matrix output generated from experiment in Tables above:

When Input size = 600, clusters = 2, with a Total support = 180

```
              Precision   Recall   f1-score   Support

          0      0.98       1.00      0.99       162

          1      1.00       0.78      0.88        18

avg / total        0.98      0.98      0.98       180
```

Confusion matrix:

```
162   0
  4  14
```

The last line gives a weighted average of precision, recall and f1-score where the weights are the support values. So for precision the avg for the 600 dataset samples to 2 clusters is: **(0.98*162 + 1.00*18)/180 = 0.98**). The total is just for total support which is 180 here.

4.1 Discussion of Results

KMEANS and SVC

- The result shows that the larger the size of the training data the higher and better the accuracy of the classifier with respect to a small number of cluster.
- Also, when two clusters produce several keywords in common the classifier's accuracy becomes low, but with distinct keywords in each clusters, the better the classifier accuracy.

KMEANS, HC and SVC

- The result generated here, shows that the introduction of HC into the problem domain does not yield any significant difference compare to when we used K-Means and SVC
- Also, the result is the same for all number of inputs sample with respect to the number of clusters.
- Except for a case where the number of inputs = 600 and clusters = 3, here K-means and SVC gave an accuracy **of 0.991** while HC + K-means + SVC gave an accuracy of **0.822 which shows that**

k-means + svc performs efficiently than HC and SVC combined

HC and SVC

- The result showed that HC performs very poorly on the email domain. It could not handle the classification problem efficiently.

- Also, the result shows that HC does not perform well with increase in the number of dataset that is, the larger the size of the training data the lower the accuracy of the classifier with respect to any number of clusters.

An overall Accuracy of **0.99** was obtained using 10,000 email dataset extracted from Enron corpus used alongside with 100 email datasets from the authors email to make a total of 10,100 email used, this was compared with [1], in which their classifier accuracy which was above **0.9,** in which the implementation of the Bayesian algorithm was done in Java and A total of 5175 e-mails are used for the purpose of their experiment. The results also show that larger the size of training data better is the accuracy [1].

5. CONCLUSION

This research work introduces an approach to classify and prioritize email messages into folder structures, in a declining order of importance according to the priorities of each user's email inbox content using SVM, K-means and Hierarchical Clustering method of machine learning techniques. En-corpus online datasets and few of Author's personal Gmail inbox messages were used to carry out this research work. SVM classifier, K-means and Hierarchical Clustering Algorithm were used, the classifiers gives a reasonable Accuracy and performance matrix as mentioned in the result discussion section above. The Result shows that the proposed model in this study gives a significant increase in performance when compared to statistical method or other machine learning algorithm in previous research work.

REFERENCES

[1] Vira D. R., Pradeep G. & Shidharth (2012). An Approach to Email Classification Using Bayesian Theorem. The Global Journal of Computer Science and Technology Software & Data Engineering, Volume 12, Issue 13.

[2] Pawan K. (2016), [Online]. Available: https://www.quora.com/What-is-the-importance-of-email, Retrieved on 12th July, 2018.

[3] Bhowmick, A., & Hazarika, S. M. (2018). E-mail spam filtering: A review of techniques and trends. In *Lecture Notes in Electrical Engineering*, [Online]. Available: https://doi.org/10.1007/978-981-10-4765-7_61, Retrieved on 26th September, 2018.

[4] Chui M., Manyika J., and Bucghin J., et al (2013), "The social economy: Unlocking value and productivity through social technologies" Published by McKinsey Global Institute, [Online]. Available: https://www.mckinsey.com Retrieved on 12 July, 2018.

[5] Krempl, G., Spiliopoulou, M., Stefanowski, J., Žliobaite, I., Brzeziński, D., Hüllermeier, E., … Sievi, S. (2014). Open challenges for data stream mining research. *ACM SIGKDD Explorations Newsletter*, *16*(1), pp, 1–10, [Online]. Available: https://doi.org/10.1145/2674026.2674028

[6] Yoo, S., Yang, Y., Lin, F., & Moon, I. (2010). Mining Social Networks for Personalized Email Prioritization. Published by IEEE Computer Society on IEE Intelligent System, [Online]. Available: https://nyc.lti.cs.cmu.edu Retrieved on 12 July, 2018

[7] Partridge, & Craig (2008). "The Technical Development of Internet Email" (PDF). *IEEE Annals of the History of Computing*. **30** (2): pp, 3–29. doi:10.1109/mahc.2008.32. ISSN 1934-1547.

[8] Susan R. Harris, Ph.D., & Elise Gerich, (1996). "Retiring the NSFNET Backbone Service: Chronicling the End of an Era" Archived 2016-01-01 at the Wayback Machine, , *ConneXions*, Vol. 10, No. 4.

[9] Whittaker, S., & Sidner, C. (1996). Email overload: exploring personal information management of email. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Common Ground*, *35*, pp, 276–283, [Available Online] : https://doi.org/10.1145/238386.238530

[10] Wang, B., Ester, M., Liao, Y., Bu, J., Zhu, Y., Guan, Z., & Cai, D. (2016). The Million Domain Challenge : Broadcast Email Prioritization by Cross-domain Recommendation, pp. 1895–1904.

[11] Aberdeen, D., Pacovsky, O., & Slater, A. (2010). The learning behind gmail priority inbox. *2010 Workshop on Learning behind Gmail Priority Inbox*, pp 3–6, [Online]. Available: http://www.acecampaign.com/goto/http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/en/pubs/archive/36955.pdf%5Cnftp://24.151.202.80/AiDisk_a1/Full/Completed/36955.pdf Retrieved on 26th September, 2018

[12] Johansen, L., Rowell, M., Butler, K., & McDaniel, P. (2007). Email Communities of Interest. *Ceas*, [Online]. Available: http://www.cse.psu.edu/~butler/pubs/ceas07.pdf Retrieved on 12th July, 2018

[13] Hartigan J. A. & Wong M. A., (1979) "A K-Means Clustering Algorithm", Applied Statistics, Vol. 28, No. 1, p100-108.

[14] Zha H., Ding C., Gu M., He .X and Simon H.D. (Dec 2001). "Spectral Relaxation for K-means Clustering", Neural Information Processing Systems vol.14 (NIPS 2001). pp. 1057-1064, Vancouver, Canada.

[15] Andrews, S., Tsochantaridis, I., & Hofmann, T. (2002). Support Vector Machines for Multiple-Instance Learning. *Advances in Neural Information Processing Systems (NIPS '02)*, *53*(9), 1689–1699, [Online]. Available: https://doi.org/10.1017/CBO9781107415324.004