SPAM or HAM?

Information relating to the dataset chosen :

I used the “SpamAssassin public mail corpus” for building the spam classifier. Please refer to the following link in order to know more about the corpus - https://spamassassin.apache.org/old/publiccorpus/readme.html

It is a collection of mail messages, suitable for use in testing spam filtering systems. The corpus is split into three parts, as follows:

i) **spam**: contains spam messages, all received from non-spam-trap sources.

ii) **easy_ham**: contains non-spam messages. These are typically quite easy to differentiate from spam, since they frequently do not contain any spammish signatures (like HTML etc).

iii) **hard_ham**: contains non-spam messages which are closer in many respects to typical spam: use of HTML, unusual HTML markup, coloured text, "spammish-sounding" phrases etc.

The corpus contains a total of 6849 emails, with about a 31% spam ratio.

Creating the Dataset from the “SpamAssassin public mail corpus” :

“dataset.zip” contains the three folders “spam”, “easy_ham” and “hard_ham” from the corpus. First I have extracted “dataset.zip” in order to create the “dataset” folder. I have implemented the following three functions which create a tabular structure(i.e. pandas dataframe) from the corpus -

i) **filename_filecontent() :**

Generates pairs of the form (filename, filecontent).

The content of the file is of the form :
header…
<emptyline>
body…

This function skips the headers and returns the body only.

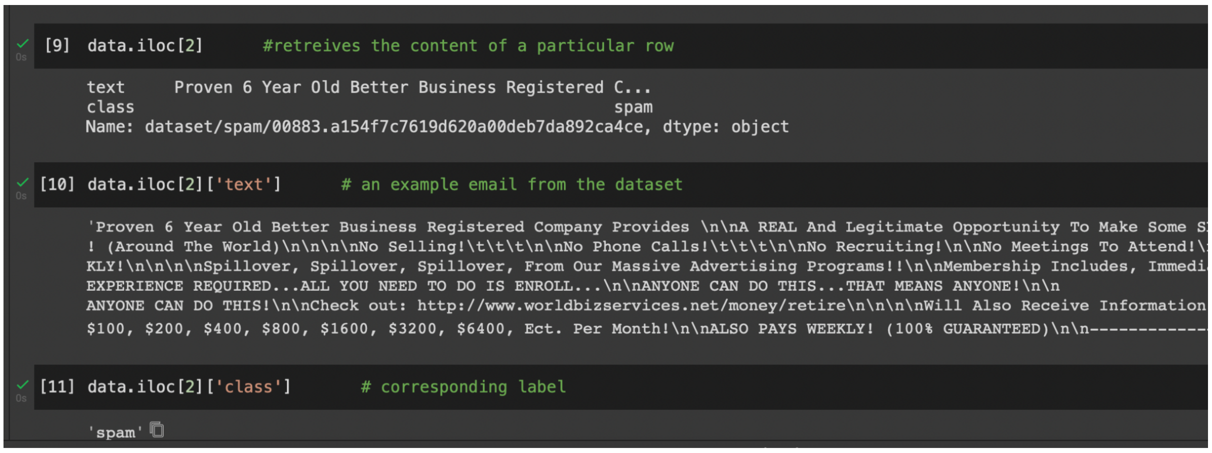
ii) **create_data_frame() :**

This function creates the pandas data frame that will store the dataset.

iii) **load_data() :**

Loads data into the dataframe and randomly shuffles the dataset.

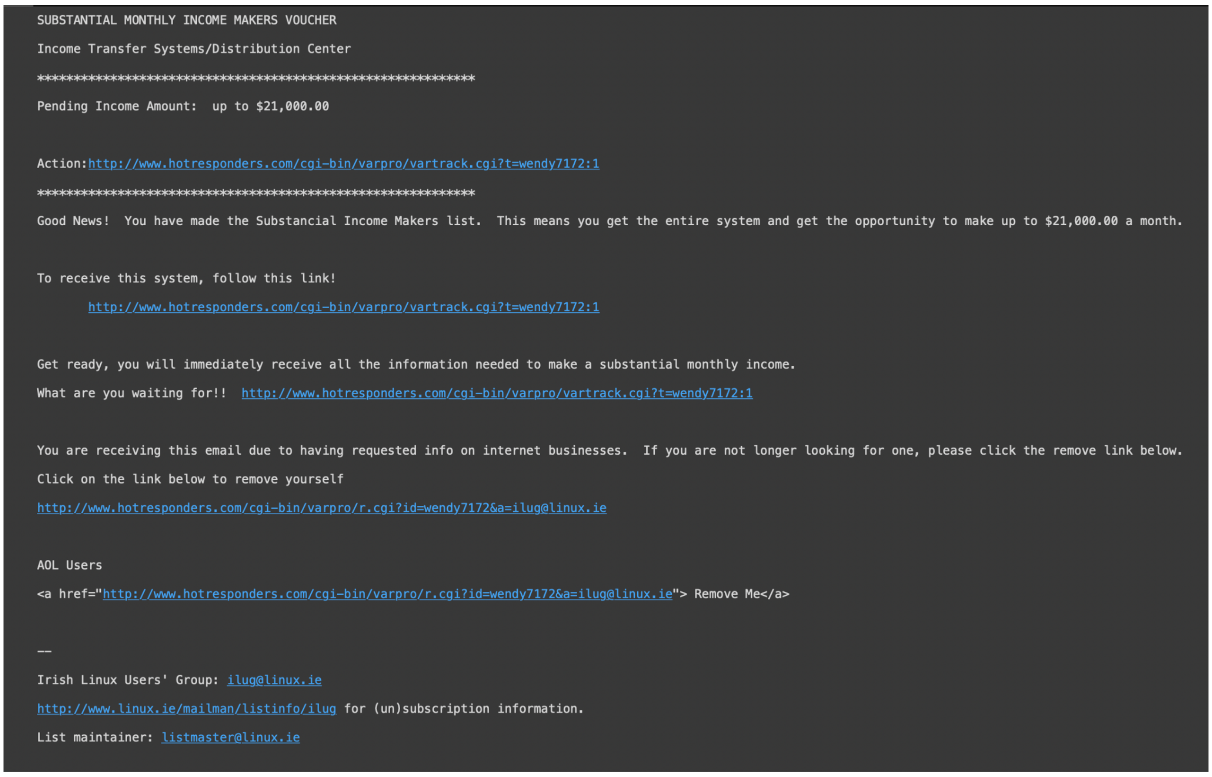
The resulting data frame has a total of 6849 rows(one row corresponding to every email) and three columns(text, class, name). Please refer to the screenshot below.



```
[9]  data.iloc[2]      #retreives the content of a particular row

     text     Proven 6 Year Old Better Business Registered C...
     class                                                spam
     Name: dataset/spam/00883.a154f7c7619d620a00deb7da892ca4ce, dtype: object

[10] data.iloc[2]['text']      # an example email from the dataset

     'Proven 6 Year Old Better Business Registered Company Provides \n\nA REAL And Legitimate Opportunity To Make Some S
     ! (Around The World)\n\n\n\nNo Selling!\t\t\t\n\nNo Phone Calls!\t\t\t\n\nNo Recruiting!\n\nNo Meetings To Attend!\
     KLY!\n\n\n\nSpillover, Spillover, Spillover, From Our Massive Advertising Programs!!\n\nMembership Includes, Immedi
     EXPERIENCE REQUIRED...ALL YOU NEED TO DO IS ENROLL...\n\nANYONE CAN DO THIS...THAT MEANS ANYONE!\n\n
     ANYONE CAN DO THIS!\n\nCheck out: http://www.worldbizservices.net/money/retire\n\n\n\nWill Also Receive Information
     $100, $200, $400, $800, $1600, $3200, $6400, Ect. Per Month!\n\nALSO PAYS WEEKLY! (100% GUARANTEED)\n\n-----------

[11] data.iloc[2]['class']       # corresponding label

     'spam'
```

Preprocessing Emails :

The following figure shows a sample email from the dataset, that contains URLs, numbers, dollar amounts, email addresses etc.



```
SUBSTANTIAL MONTHLY INCOME MAKERS VOUCHER

Income Transfer Systems/Distribution Center

************************************************************

Pending Income Amount:  up to $21,000.00


Action:http://www.hotresponders.com/cgi-bin/varpro/vartrack.cgi?t=wendy7172:1

************************************************************

Good News!  You have made the Substancial Income Makers list.  This means you get the entire system and get the opportunity to make up to $21,000.00 a month.


To receive this system, follow this link!

       http://www.hotresponders.com/cgi-bin/varpro/vartrack.cgi?t=wendy7172:1


Get ready, you will immediately receive all the information needed to make a substantial monthly income.

What are you waiting for!!  http://www.hotresponders.com/cgi-bin/varpro/vartrack.cgi?t=wendy7172:1


You are receiving this email due to having requested info on internet businesses.  If you are not longer looking for one, please click the remove link below.

Click on the link below to remove yourself

http://www.hotresponders.com/cgi-bin/varpro/r.cgi?id=wendy7172&a=ilug@linux.ie


AOL Users

<a href="http://www.hotresponders.com/cgi-bin/varpro/r.cgi?id=wendy7172&a=ilug@linux.ie"> Remove Me</a>


--

Irish Linux Users' Group: ilug@linux.ie

http://www.linux.ie/mailman/listinfo/ilug for (un)subscription information.

List maintainer: listmaster@linux.ie
```

Many emails would contain similar types of entities (numbers, other URLs, other email addresses etc.). One method often used in processing emails is to normalize these values, so that all URLs are treated the same, all numbers are treated the same etc. This basically allows the spam classifier to make a decision based on whether any URL was present, rather than whether a specific URL was present.

In the function **email_preprocessing()** I have implemented the following preprocessing and normalization steps :

Lower-casing : Converted the entire email into lower case(so that “Income” gets treated the same as “income” by the spam classifier)

Removal of HTML tags : Many emails often come with HTML formatting. I removed all the HTML tags so that only the content remains.

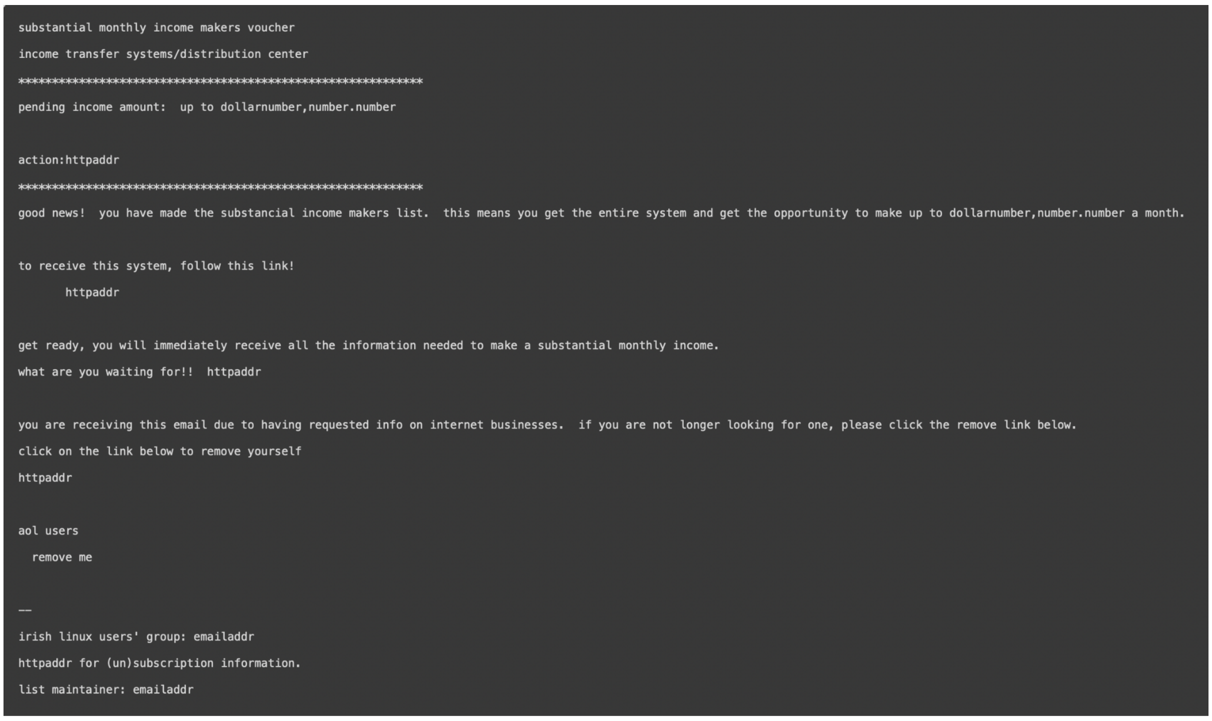
Replacing URLs with unique string : All URLs are replaced with the unique string “httpaddr”.

Replacing email addresses with unique string : All email addresses are replaced with the unique string “emailaddr”.

Handling Numbers and Dollar signs : Replaced all numbers and dollar signs with the strings “number” and “dollar” respectively.

I used regular expressions to implement the above operations.

The following figure shows the result of the above steps :



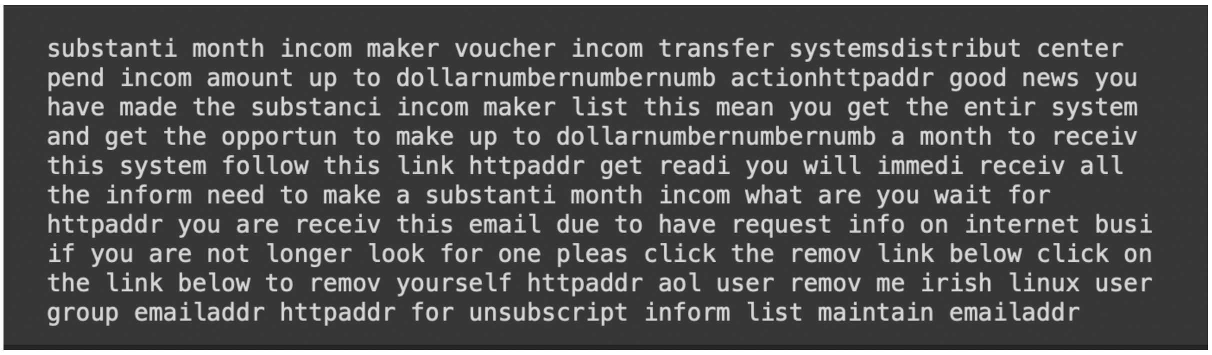
```
substantial monthly income makers voucher
income transfer systems/distribution center
************************************************************
pending income amount:  up to dollarnumber,number.number

action:httpaddr
************************************************************
good news!  you have made the substancial income makers list.  this means you get the entire system and get the opportunity to make up to dollarnumber,number.number a month.

to receive this system, follow this link!
       httpaddr

get ready, you will immediately receive all the information needed to make a substantial monthly income.
what are you waiting for!!  httpaddr

you are receiving this email due to having requested info on internet businesses.  if you are not longer looking for one, please click the remove link below.
click on the link below to remove yourself
httpaddr

aol users
  remove me


--
irish linux users' group: emailaddr
httpaddr for (un)subscription information.
list maintainer: emailaddr
```

Word stemming, removal of non-words and punctuations :

Reduced words to their stemmed form. For example, “discount”, “discounts”, “discounted” and “discounting” were all replaced with “discount”. Sometimes word stemming eliminates additional characters, so “include”, “includes”, “included” and “including” are all replaced with “includ”.

Non-words and punctuations have also been removed. Replaced all white spaces (tabs, newlines, spaces) with a single space character.

The following screenshot shows the result after applying the two operations mentioned above -



```
substanti month incom maker voucher incom transfer systemsdistribut center
pend incom amount up to dollarnumbernumbernumb actionhttpaddr good news you
have made the substanci incom maker list this mean you get the entir system
and get the opportun to make up to dollarnumbernumbernumb a month to receiv
this system follow this link httpaddr get readi you will immedi receiv all
the inform need to make a substanti month incom what are you wait for
httpaddr you are receiv this email due to have request info on internet busi
if you are not longer look for one pleas click the remov link below click on
the link below to remov yourself httpaddr aol user remov me irish linux user
group emailaddr httpaddr for unsubscript inform list maintain emailaddr
```

It is clear from the above screenshot that the stemming operation has left word fragments and non-words. But this form is very useful for performing feature extraction.

Vocabulary :

After the preprocessing steps we get a list of words for each email. I constructed the vocabulary that contains only the most frequently occurring words. The vocabulary is stored in the file “vocabulary.txt”. The vocabulary was constructed by choosing all words which occur at least a 100 times in the corpus.

Word Vector for every email :

I mapped each word in the preprocessed emails into a list(i.e word vector of the corresponding email, we have one word vector for each email) that contains the index of the word in the vocabulary dictionary. If a word exists in the vocabulary then the index gets added in “word vector”, otherwise we skip the word.

Constructing Feature Vectors from Word Vectors :

create_feature_vector() function takes a word vector as input and returns a feature vector. Specifically, the feature $x\_i \in \{0, 1\}$ for an email corresponds to whether the i-th word in the dictionary occurs in the email. That is, $x\_i = 1$ if the i-th word is in the email and $x\_i = 0$ if the i-th word is not present in the email.

Two matrices train_x and train_y :

Corresponding to every email we have a feature vector. The matrix “train_x” stores the vectorial representations of all the 6849 emails(every row corresponds to the feature vector of an email) that we have in the “SpamAssassin Public Corpus” and “train_y” stores the corresponding labels. I have used these two matrices two train my Support Vector Machine classifier.

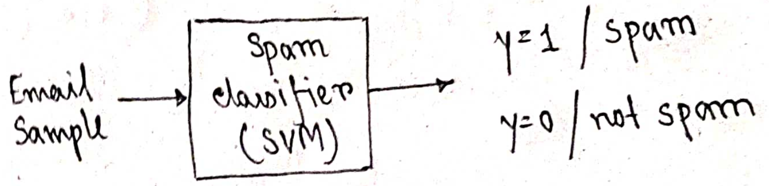
Algorithm used for training :

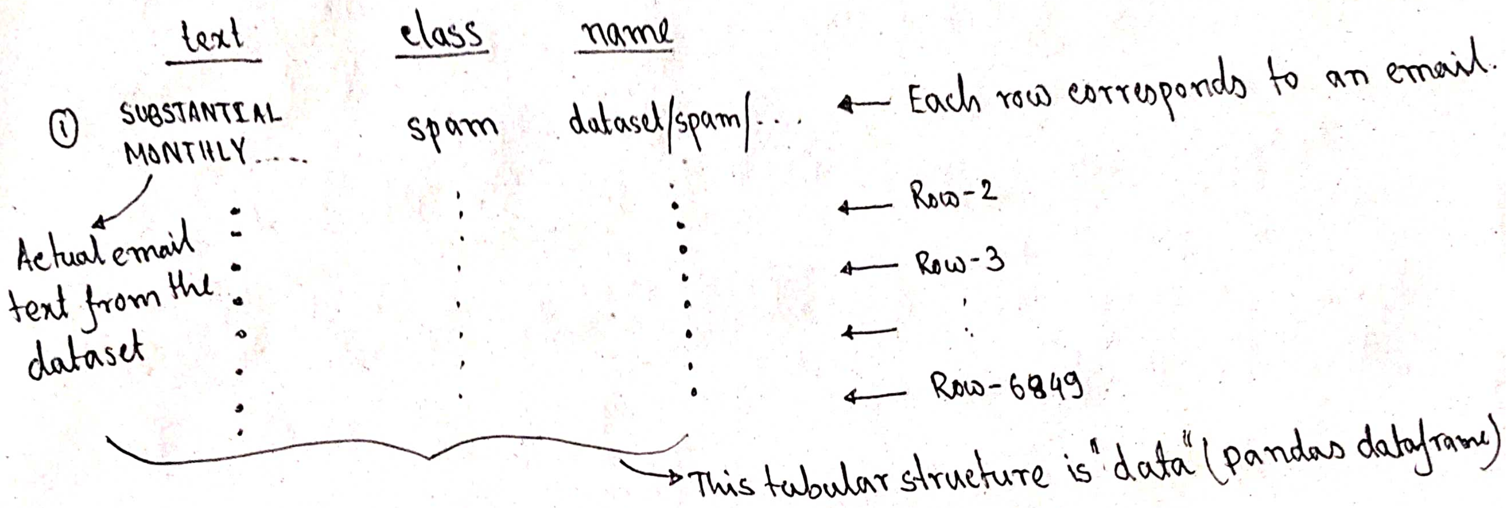
I trained a SVM classifier to classify whether a given email(x) is spam(y=1) or non-spam(y=0). As per the question **“Algorithms except SVM need to be coded from scratch”.** Hence I did not implement SVM from scratch. I have used “sklearn” for building the SVM classifier. **I used C(regularization parameter) = 0.1 and the “linear” kernel. I obtained a training accuracy of - 99.77%.**

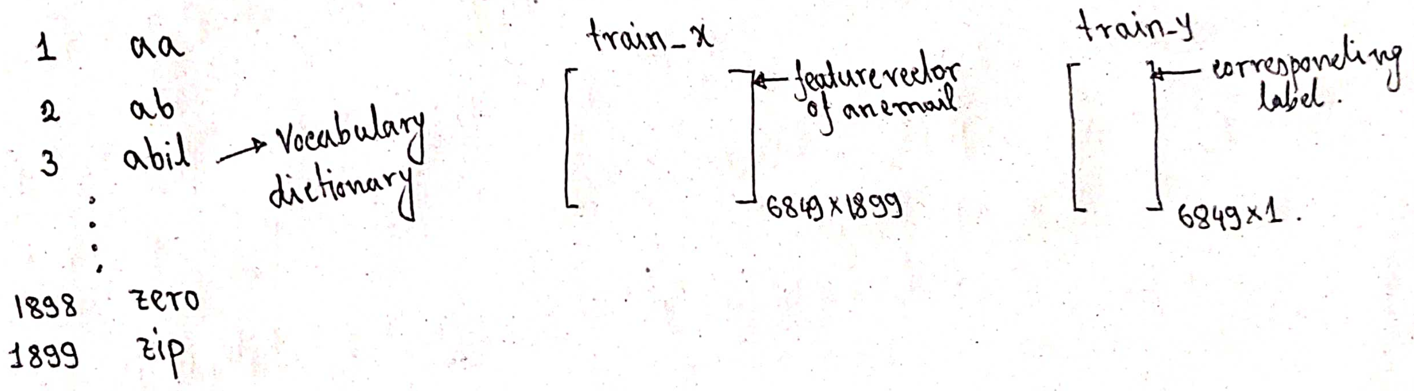
Instructions for testing the spam classifier :

I have submitted the dataset(dataset.zip), vocabulary and test.zip along with the code and report. Please keep these three files and the code in the same directory. Running one code block after another will perform the desired operations. The ‘test’ folder contains a few sample emails(including the two emails that were shared with us). At the end I have a function named **“read_and_predict” -** this function reads the emails of the ‘test’ folder, generates predictions for each of the test emails and returns a prediction vector.

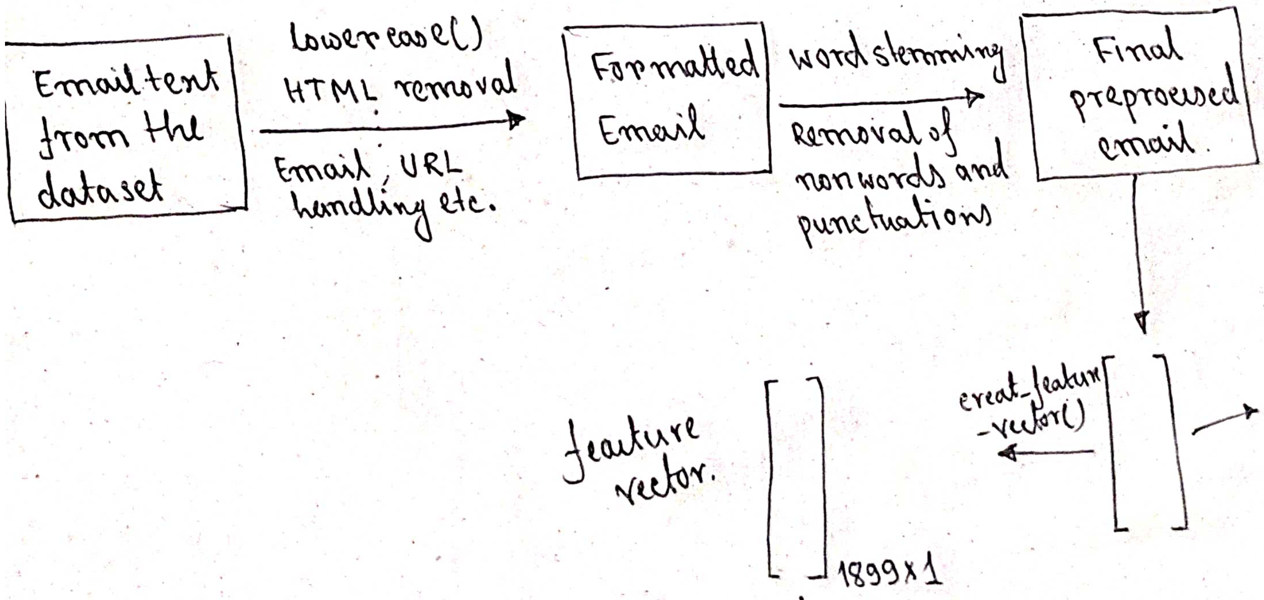
dataset (spam Assassin public corpus)
- easy-ham
- hard-ham
- spam




text
class
name
① SUBSTANTIAL MONTHLY....
spam
dataset/spam/...
Each row corresponds to an email.
Actual email text from the dataset
Row-2
Row-3
Row-6849
This tabular structure is "data" (pandas dataframe)



Email preprocessing and feature extraction:-


Email text from the dataset
lowercase()
HTML removal
Email, URL handling etc.
Formatted Email
word stemming
Removal of nonwords and punctuations
Final preprocessed email
word vector representation.
creat_feature-vector()
feature vector.
1899x1