Spam mail classifier

Team 3 - Kunal Chugh & Vighnesh Venkatakrishnan

Project summary

Our objective was to detect whether an email was spam or not

Dataset involved details of sender, receiver, email body and subject, message id, xuidl and a lot of other fields in text files

We divided the project into 3 parts – structuring data, exploratory data analysis and feature extraction/training testing data

We used Google colab to run our notebooks and achieved up to 72% accuracy for classifying spam emails correctly.

Project Implementation

Github repo link : https://github.com/kunalchugh91/INFO6101-Spam-Classifier

Packages/Modules needed to run this project are: nltk, wordcloud, xlrd, numpy, pandas, spark, xlsxwriter, BeautifulSoup, matplotlib, findspark, scikit-learn.

The datasets for this project can be downloaded from the below links:

1. 1998 dataset - https://drive.google.com/open?id=1QtoxpJmd1lys7c7LaYXiOjbzMdMOpeVX

2. trec07p dataset - https://drive.google.com/open?id=1xaJL1eoccrCyS45xgF23dVY_KCER-oAD

Part – 1 – Structuring data

Parsed unstructured text files into list of email header : value pairs.

File encoding issues : utf-8 and ISO-8859-1

New line is a new key value pair? (colon is within first 30 chars, first letter is capital, no space in header, create list “notHeader”, hold value in temp variable)

Many email text files contained duplicate fields. Removed such redundant fields.

Many email body fields contained HTML content. Used BeautifulSoup to get visible text from such HTML content.

Store everything in a pandas dataframe and excel file

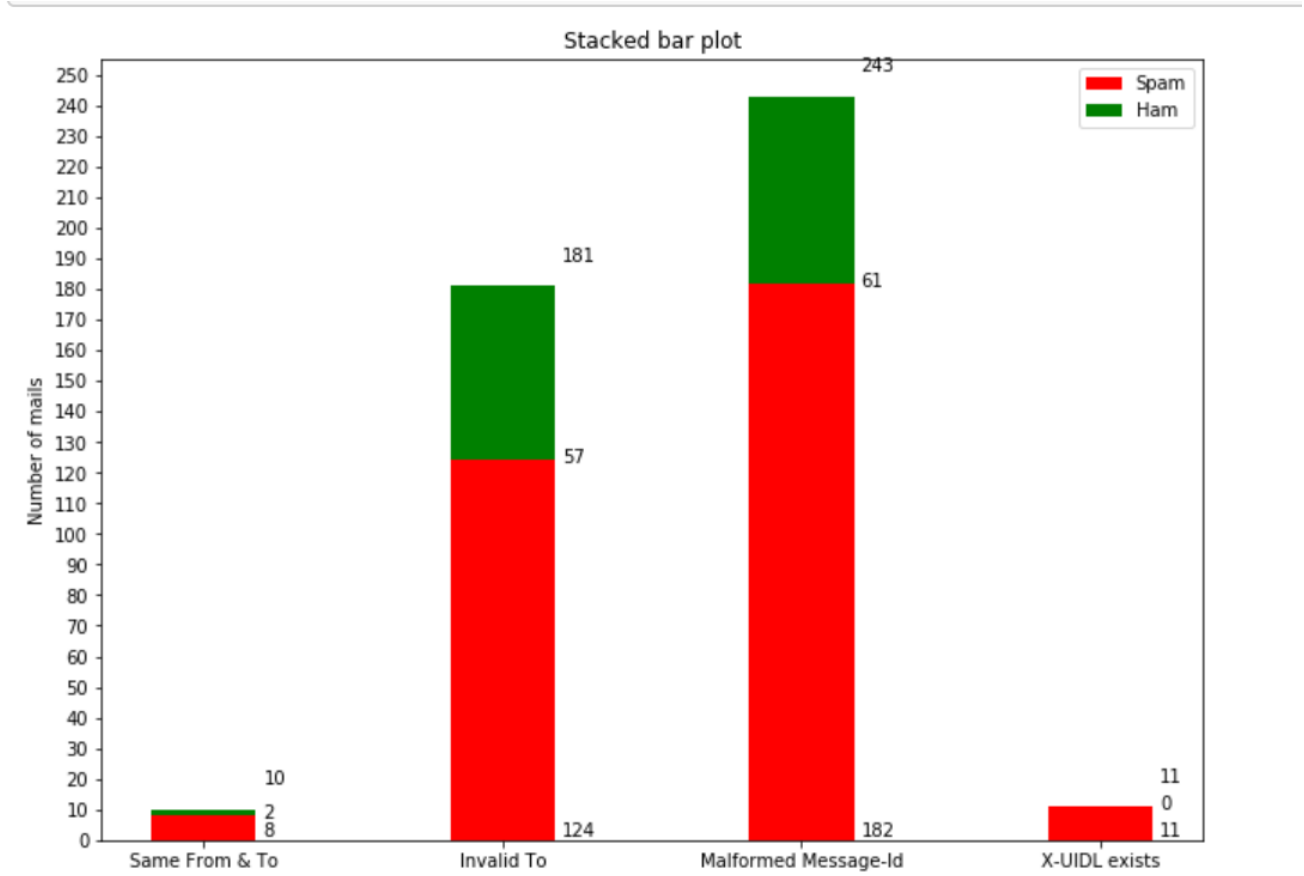
```
From RickyAmes@aol.com  Sun Apr  8 13:07:32 2007
Return-Path: <RickyAmes@aol.com>
Received: from 129.97.78.23 ([211.202.101.74])
    by speedy.uwaterloo.ca (8.12.8/8.12.5) with SMTP id l38H7G0I003017;
    Sun, 8 Apr 2007 13:07:21 -0400
Received: from 0.144.152.6 by 211.202.101.74; Sun, 08 Apr 2007 19:04:48 +0100
Message-ID: <WYADCKPDFWWTWTXNFVUE@yahoo.com>
From: "Tomas Jacobs" <RickyAmes@aol.com>
Reply-To: "Tomas Jacobs" <RickyAmes@aol.com>
To: the00@speedy.uwaterloo.ca
Subject: Generic Cialis, branded quality@
Date: Sun, 08 Apr 2007 21:00:48 +0300
X-Mailer: Microsoft Outlook Express 6.00.2600.0000
MIME-Version: 1.0
Content-Type: multipart/alternative;
    boundary="--8896484051606557286"
X-Priority: 3
X-MSMail-Priority: Normal
Status: RO
Content-Length: 988
Lines: 24

----8896484051606557286
Content-Type: text/html;
Content-Transfer-Encoding: 7Bit

<html>
<body bgcolor="#ffffff">
<div style="border-color: #00FFFF; border-right-width: 0px; border-bottom-width: 0px; margin-bottom: 0px;" align="center">
<table style="border: 1px; border-style: solid; border-color:#000000;" cellpadding="5" cellspacing="0" bgcolor="#CCFFAA">
<tr>
<td style="border: 0px; border-bottom: 1px; border-style: solid; border-color:#000000;">
<center>
Do you feel the pressure to perform and not rising to the occasion??<br>
</center>
</td></tr><tr>
```

Part – 2 – Exploratory data analysis

- The purpose of this notebook is to explore the data and find out key features that identify whether an email is spam or not
- Used NLTK for data preprocessing of the body – regex clean, tokenize, stem, stopword removal and generated a wordcloud
- Words like mailing, list, retail, price, per, item, posting are frequently found in spam emails. They indicate that the spammer wants to sell an item or send spams after user is subscribed to a mailing list or job posting.

Part – 2 – Exploratory data analysis

- Feature 1 : from field == key field
- Feature 2 : invalid email format of recipient(in to field, not bcc)
- Feature 3 : malformed message id (<>\n)
- Feature 4 : X-UIDL is not empty(mailserver keeps receiving multiple copies)



Feature 5 – TFIDF engineering

- Used spark to create body tokens, generate tf and calculate tf-idf
- Transform our train dataset to extract features (4 mentioned earlier and TFIDF) using pyspark and train a Multinomial Naive Bayes model
- Used Spark Vector Assembler to combine all raw features into one feature vector

```
+--------------------+----------+----------+----------+----------+----+-----+--------------------+----
--------------+--------------------+--------------------+--------------------+
|                Body|RawFeature1|RawFeature2|RawFeature3|RawFeature4|Spam|label|         Body_Tokens|
   Body_Tokens2|          TermFreqs|         RawFeature5|            Features|
+--------------------+----------+----------+----------+----------+----+-----+--------------------+----
--------------+--------------------+--------------------+--------------------+
|email marketing w...|         0|         0|         0|         0|Spam|  0.0|[email, marketing...|[ema
il, marketing...|(20,[0,1,2,3,4,5,...|(20,[0,1,2,3,4,5,...|[0.0,0.0,0.0,0.0,...|
|   this is the mo...|         0|         0|         1|         0|Spam|  0.0|[this, is, the, m...|[exc
iting, breakt...|(20,[0,1,2,3,4,5,...|(20,[0,1,2,3,4,5,...|[0.0,0.0,1.0,0.0,...|
+--------------------+----------+----------+----------+----------+----+-----+--------------------+----
--------------+--------------------+--------------------+--------------------+
only showing top 2 rows
```

Part 3 - feature extraction, training, prediction

- Partition dataset randomly into train-test splits (75-25)
- Attempted to calculate percentage of misspelled words as our 6th feature, but did not use it
- Used feature vector from Vector assembler to fit on Multinomial NaïveBayes model
- Used the test dataset to calculate the accuracy of our prediction.