Capstone Proposal

Bipin Sawant

Machine Learning Engineer Nanodegree

July 05, 2020

Definition

Domain Background

The 1990s marked the beginning of internet with Hotmail being the first web-based email provider. Standards such as sender authentication, whitelisting, etc. were unknown. Marketers exploited this opportunity. As a result, our inboxes get flooded with junk emails every day.

In late 1990 to early 2000, with email abuse rising, ISPs started to use crude filters based on keywords, patterns or special characters. Blacklists began to emerge, listing the IPs of known bad senders. Misuse continued as the ISPs often still didn't have enough data to prove the claims of their senders to be legitimate and the process was still based on human decision.

During the mid-to-late 2000s, next major development in email deliverability included the standardization of reputation scores. False positives were further reduced by adding sender authentication mechanisms like SPF and SenderID, and following them also Domainkeys and DKIM (Domainkeys Identified Mail).

2010 onwards we have seen important changes to email spam. Instead of relying on their own judgement, email clients have put the power firmly in the hands of their users. This in turn has given ISPs the opportunity to embrace 'Big Data' and "Machine Learning", to record all the inbox actions of all their users to determine what really constitutes "wanted" or "unwanted" emails. For example, below are the few action which could be used to measure a sender's engagement:

- Moving an email to spam or trash is a negative reaction of a recipient and reflects badly on the sender and help in classifying the mail content as spam.
- Replying or forwarding shows positive engagement.
- Adding the sender to the address book is the best indication of all that an email is genuinely wanted.

The data collected through above feedback is used to further train the model and hence making the Spam Filter more efficient.

Problem statement

The project is to implement a Spam Filter with the help of a ML classifier which would classify as given mail, as spam or ham.

Data Source:

Apache SpamAssasin public corpus has spam and ham mails data. Below is the overview of data
URL:

- http://spamassassin.apache.org/old/publiccorpus/

Content Folder:

- spam: 501 spam messages, all received from non-spam-trap sources.
- easy_ham: 2501 non-spam messages. These are typically quite easy to differentiate from spam, since they frequently do not contain any spammish signatures (like HTML etc).
- hard_ham: 251 non-spam messages which are closer in many respects to typical spam: use of HTML, unusual HTML markup, coloured text, "spammish-sounding" phrases etc.
- easy_ham_2: 1401 non-spam messages. A more recent addition to the set.
- spam_2: 1397 spam messages. Again, more recent.

Total count: **6051** messages, with about a **31.26%** spam ratio.

Spam Mail Example

```
From 12a1mailbot1@web.de  Thu Aug 22 13:17:22 2002
Return-Path: <12a1mailbot1@web.de>
Delivered-To: zzzz@localhost.spamassassin.taint.org
Received: from localhost (localhost [127.0.0.1])
     by phobos.labs.spamassassin.taint.org (Postfix) with ESMTP
id 136B943C32
     for <zzzz@localhost>; Thu, 22 Aug 2002 08:17:21 -0400 (EDT)
Received: from mail.webnote.net [193.120.211.219]
     by localhost with POP3 (fetchmail-5.9.0)
     for zzzz@localhost (single-drop); Thu, 22 Aug 2002 13:17:21
+0100 (IST)
Received: from dd_it7 ([210.97.77.167])
     by webnote.net (8.9.3/8.9.3) with ESMTP id NAA04623
     for <zzzz@spamassassin.taint.org>; Thu, 22 Aug 2002
13:09:41 +0100
From: 12a1mailbot1@web.de
Received: from r-smtp.korea.com - 203.122.2.197 by dd_it7  with
Microsoft SMTPSVC(5.5.1775.675.6);
      Sat, 24 Aug 2002 09:42:10 +0900
To: <dcek1a1@netsgo.com>
Subject: Life Insurance - Why Pay More?
Date: Wed, 21 Aug 2002 20:31:57 -1600
MIME-Version: 1.0
Message-ID: <0103c1042001882DD_IT7@dd_it7>
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META content=3D"text/html; charset=3Dwindows-1252" http-
equiv=3DContent-T=
ype>
<META content=3D"MSHTML 5.00.2314.1000" name=3DGENERATOR></HEAD>
<BODY><!-- Inserted by Calypso -->
<TABLE border=3D0 cellPadding=3D0 cellSpacing=3D2
id=3D_CalyPrintHeader_ r=
ules=3Dnone
style=3D"COLOR: black; DISPLAY: none" width=3D"100%">
```

Ham Mail Example

```
Return-Path: <malcolm-sweeps@mrichi.com>
Delivered-To: rod@arsecandle.org
Received: (qmail 16821 invoked by uid 505); 7 May 2002 14:37:01
-0000
Received: from malcolm-sweeps@mrichi.com by
blazing.arsecandle.org
      by uid 500 with qmail-scanner-1.10 (F-PROT: 3.12. Clear:0.
Processed in 0.260914 secs); 07 May 2002 14:37:01 -0000
Delivered-To: rod-3ds@arsecandle.org
Received: (qmail 16811 invoked by uid 505); 7 May 2002 14:37:00
-0000
Received: from malcolm-sweeps@mrichi.com by
blazing.arsecandle.org
      by uid 502 with qmail-scanner-1.10 (F-PROT: 3.12. Clear:0.
Processed in 0.250416 secs); 07 May 2002 14:37:00 -0000
Received: from bocelli.siteprotect.com (64.41.120.21)
  by h0090272a42db.ne.client2.attbi.com with SMTP; 7 May 2002
14:36:59 -0000
Received: from mail.mrichi.com ([208.33.95.187])
     by bocelli.siteprotect.com (8.9.3/8.9.3) with SMTP id
JAA14328;
     Tue, 7 May 2002 09:37:01 -0500
From: malcolm-sweeps@mrichi.com
Message-Id: <200205071437.JAA14328@bocelli.siteprotect.com>
To: <rod-3ds@arsecandle.org>
Subject: Malcolm in the Middle Sweepstakes Prize Notification
Date: Tue, 7 May 2002 9:38:27 -0600
X-Mailer: sendEmail-v1.33

May 7, 2002
Dear rod-3ds@arsecandle.org:

Congratulations!  On behalf of Frito-Lay, Inc., we are pleased
to advise you
 that you've won Fourth Prize in the 3D's(R) Malcolm in the
Middle(TM)
 Sweepstakes.   Fourth Prize consists of 1 manufacturer's coupon
redeemable at
 participating retailers for 1 free bag of 3D's(R) brand snacks
(up to 7 oz.
 size), with an approximate retail value of $2.59 and an
expiration date of
 12/31/02.
```

Solution statement

To create solution of the above problem we would be performing below steps:

- Pre-processing the mail - In this step we will process the mail content with following changes:
    - Remove html tags
    - Normalize numbers, Urls, email address and Dollar sign (Replace 0-9 with ‘number, hyperlinks with 'httpaddr', any email address with 'emailaddr' and $ with 'dollar')
    - Tokenize the words
    - Take only alphanumeric words
    - Lemmatize and change case to lower to reduce the words to their stemmed form.
    - Filter stop words (words which do not have significance like: to, the, a etc)
- Extracting the features: Create a training set from above processed mail. From this set calculate frequency of each word and select high frequency word as feature for be used for learning.
- Training different classifier: Train the classifiers like Logistic Regression, K-Neighbour Classifier, SVM, Naïve Bayes etc
- Evaluating the classifiers: The above classifiers will be evaluated against the dummy classifier, by comparing their prediction time and f1score.

Benchmark model

In this project we will use the sklearn dummy classifier as a simple baseline to compare with. DummyClassifier is a classifier that makes predictions using simple rules, so the goal is to perform better that this.

Evaluation metrics

In real world scenario volume of ham males are generally higher that the spam mails. For such problem f1score is a better metric to analyze performance of a classifier. In the given dataset also the spam ratio is 31% only, so f1score is applicable for the given dataset as well. The solution will evaluate f1score of ML model against the benchmark f1score.

Project design

Implement a supervised learning classifier model using the SpamAssasin public corpus data to train and test the model. Project design would consist of following sections

1. Data Pre-Processing
   - Remove html tags and Normalize numbers, Urls, email address and Dollar sign with regex rules.
   - Tokenize the words using nltk word_tokenize
   - Change to lower case and take only alphanumeric words.
   - Lemmatize the words to their stemmed form using nltk WordNetLemmatizer
   - Filter stop words using nltk.corpus stopwords

2. Data Visualization: using matplotlib and WordCloud
   - Plot spam mail words and its frequency using histogram and word cloud.
   - Plot ham mail words and its frequency using histogram and word cloud

3. Train- Test Split: Split Data in Training and Testing Set using sklearn.model_selection train_test_split

4. Feature Extraction: From the training set take high frequency spam and ham mail words to be used as feature.

5. Create Feature Data: using the processed mail (in step 1) and extracted feature (in step 4), create feature data for train and test set

6. Train different classifiers: Train different classifiers like Logistic Regression, K-Neighbor Classifier, SVM, Naïve Bayes etc.

7. Evaluate classifiers and create evaluation matrix: The above classifiers will be evaluated against the dummy classifier, by comparing their prediction time and f1score.

8. Fine Tune the best classifier: Use GridSearch technique to fine tune the best classifier using different parameters like C, gamma and other applicable parameters