4_spam_detection

February 28, 2021

Candidate: André Oliveira Françani

1 SMS Ham-Spam Detection

1.1 Description

The SMS Ham-Spam detection dataset is a set of SMS tagged messages that have been collected for SMS Spam research. It contains a set of 5,574 SMS messages in English, considering both train and test data. The tagging standard was defined as `ham` (legitimate) or `spam`.

The `train` and `test` files are formatted using the standard of one message per line. Each line is composed by two columns: one with label (`ham` or `spam`) and other with the raw text. Here are some examples:

```
ham   What you doing?how are you?
ham   Ok lar... Joking wif u oni...
ham   dun say so early hor... U c already then say...
ham   MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*
ham   Siva is in hostel aha:-.
ham   Cos i was out shopping wif darren jus now n i called him 2 ask wat present...
spam   FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time...
spam   Sunshine Quiz! Win a super Sony DVD recorder if you canname the capital...
spam   URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus Caller Prize...
```

`Note: messages are not chronologically sorted.`

For evaluation purposes, the `test` dataset does not prosent the categories (`ham`, `spam`). Therefore, the `train` data is the full source of information for this test.

1.2 Objective

The goal of the this test is to achieve a model that can correctly manage the incoming messages on SMS format (`ham` or `spam`). Considering a real scenario, assume that a regular person does not want to see a `spam` message. However, they accepts if a normal message (`ham`) is sometimes allocated at the `spam` box.

1.3 Important details

- The dataset was split in order to have unseen data for analysis. We took 15% of the total data (randomly)

- Replicate the data format for submission, i.e. the answer must be provided as a CSV file with the detect class in the first column and the text in the second column, similarly to what is provided in the `TrainingSet` file
- The `TestSet` will be used for evalution, therefore the candidate must provide the first column with the predicted classes (`ham` or `spam`)
- Pay attention to the real case scenario that was described in the Objective section. This may drive the problem solving strategy :wink:.
- This test does not require a defined set of algorithms to be used. The candidate is free to choose any kind of data processing pipeline to reach the best answer.

1.4 Implementation

To solve this problem, the first thing to do is to read and prepare the data using the pandas library and store it in a DataFrame.

[1]:
```
#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import string

from nltk.corpus import stopwords #library nltk with common words
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
```

[2]:
```
### GENERATE DATASET ###

#reading and organizing trainig data
training_set = pd.read_csv('TrainingSet/sms-hamspam-train.csv', sep='\n',␣
↪header=None)
training_set = training_set[0].str.split('\t',expand=True)
training_set.columns = ['label','msg']
training_set['usage'] = 'train' #create a new column with 'train'

#binarizing label (spam = 1 and ham = 0)
training_set.label = (training_set.label == 'spam').astype(int)


#reading and organizing test data
test_set = pd.read_csv('TestSet/sms-hamspam-test.csv', sep='\n', header=None,␣
↪names=['msg'])
test_set['usage'] = 'test'      #create a new column with 'test'
```

```
#concatenate training set and test set to do pre-process the data
dataset = pd.concat([training_set, test_set]).reset_index(drop=True)
```

The following procedure is the ***tokenization***, removing the punctuation and some common words, namely pronouns, articles, prepositions etc. These words can be found in stopwords of the nltk library.

[3]:
```
#list with stopwords
stopwords_list = stopwords.words('English')

#list with all punctuation
punctuation_str = string.punctuation
punctuation_list = [0]*len(punctuation_str)
for i in range(len(punctuation_list)):
    punctuation_list[i] = punctuation_str[i]

#remove words and punctuation of all messages
idx_row = 0
for sms in dataset.msg:
    msg = [word for word in sms if word not in punctuation_list] #remove␣
 ↪punctuation of message
    msg = ''.join(msg)                                          #rebuild string
    msg = msg.lower()
    msg = [word for word in msg.split(' ') if word.lower() not in␣
 ↪stopwords_list and word.isalpha()] #remove stopwords from message

    dataset.msg[idx_row] = ' '.join(msg)
    #dataset.msg[idx_row] = msg
    idx_row += 1
```

```
C:\Users\andre\Anaconda3\envs\processos_seletivos\lib\site-
packages\ipykernel_launcher.py:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Now the data should be divided into train and test sets. Here, I got only 10% of test data because I want the more information as possible during training, since the real test data is already given in the exercise, however without label.

[4]:
```
X_train, X_val, Y_train, Y_val = train_test_split(dataset.loc[dataset.usage ==␣
 ↪'train','msg'], dataset.loc[dataset.usage == 'train','label'], test_size = 0.
 ↪1, random_state = 1)
```

After that, the ***Vectorization*** is performed since we are dealing with strings. Therefore we should encode the words into values to apply the machine learning algorithms.

[5]:
```
#Term Frequency-Inverse Document Frequency (TF-IDF)
vectorizer = TfidfVectorizer()              #build vectorizer
X_train = vectorizer.fit_transform(X_train) #train vectorizer

#Apply vectorizer in val set
X_val = vectorizer.transform(X_val)
```

Now the classifier is built to classify the messages. The first approach is using a **Support Vector Machine (SVM)** to detect spam messages.

[6]:
```
#Support Vector Machine (SVM)
svm_model = svm.SVC(C=10) #tested with different C values
svm_model.fit(X_train, Y_train)

#Predict Class
Y_pred = svm_model.predict(X_val)

#Compute confusion matrix
cf = metrics.confusion_matrix(Y_val, Y_pred)
```

1.4.1 Confusion Matrix - SVM:

	Predicted: Ham	Predicted: Spam
Actual: Ham	392 (TN)	0 (FP)
Actual: Spam	12 (FN)	68 (TP)

From the goal of this test, a regular person does not want to see a spam message. However, they accept if a normal message (ham) is sometimes allocated at the spam box. This means that we accept more false positives (Type I error) than false negatives (Type II error).

The next tested model is a **Naive Bayes** classifier.

[7]:
```
#Naive Bayes classifier
nbc_model = MultinomialNB()
nbc_model = nbc_model.fit(X_train, Y_train)

#Predict Class
Y_pred = nbc_model.predict(X_val)

#Compute confusion matrix
cf = metrics.confusion_matrix(Y_val, Y_pred)
```

1.4.2 Confusion Matrix - Naive Bayes:

	Predicted: Ham	Predicted: Spam
Actual: Ham	392 (TN)	0 (FP)
Actual: Spam	18 (FN)	63 (TP)

Since there were more false negatives using Naive Bayes, it can be concluded that the SVM had a better performance than the Naive Bayes model.

The next tested classifier is a **Decision Tree** model.

[8]:
```
#Decision Tree classifier
dtc_model = DecisionTreeClassifier(min_samples_split=7, random_state=111)
dtc_model = dtc_model.fit(X_train, Y_train)

#Predict Class
Y_pred = dtc_model.predict(X_val)

#Compute confusion matrix
cf = metrics.confusion_matrix(Y_val, Y_pred)
```

1.4.3 Confusion Matrix - Decision Tree:

	Predicted: Ham	Predicted: Spam
Actual: Ham	383 (TN)	9 (FP)
Actual: Spam	17 (FN)	64 (TP)

The last classifier tested is a **Random Forest** model

[9]:
```
#Random Forest classifier
rfc_model = RandomForestClassifier(n_estimators=31, random_state=111)
rfc_model = rfc_model.fit(X_train, Y_train)

#Predict Class
Y_pred = rfc_model.predict(X_val)

#Compute confusion matrix
cf = metrics.confusion_matrix(Y_val, Y_pred)
```

1.4.4 Confusion Matrix - Random Forest:

	Predicted: Ham	Predicted: Spam
Actual: Ham	392 (TN)	0 (FP)
Actual: Spam	13 (FN)	68 (TP)

Comparing all models, the SVM had the best performance since it scored the lower false negatives and no false positives. Therefore, this model will be use to classify the given test messages in this exercise. The predicted results are saved in ***'sms-hamspam-test-solution.csv'***.

[10]:
```
#pre-process test data
X_test = vectorizer.transform(dataset.loc[dataset.usage == 'test','msg']) #test␣
 ↪vectorizer

#predict test label
Y_test_pred = pd.Series(svm_model.predict(X_test))

#categorical label
Y_test_pred[Y_test_pred == 1] = 'spam'
Y_test_pred[Y_test_pred == 0] = 'ham'

#make csv file
save_file = pd.DataFrame(
            {'label': Y_test_pred,
             'message': test_set.msg})
save_file.to_csv('sms-hamspam-test-solution.csv', index=False, header=False,␣
 ↪sep='\t')
```

Let's read the generated file with the predicted class and visualize some messages detected as spam.

[11]:
```
pred_test = pd.read_csv('sms-hamspam-test-solution.csv', sep='\t', header=None,␣
 ↪names=['label','msg'])
spam_idx = pred_test[pred_test.label=='spam'].index

#print the first 20 messages detected as spam
for i in spam_idx[:20]:
    print('SMS '+ str(i+1) + ' : '+ pred_test.loc[i,'msg'] + '\n')
```

```
SMS 10 : Congratulations ur awarded 500 of CD vouchers or 125gift guaranteed &
Free entry 2 100 wkly draw txt MUSIC to 87066 TnCs www.Ldew.com1win150ppmx3age16

SMS 14 : Hi. Customer Loyalty Offer:The NEW Nokia6650 Mobile from ONLY £10 at
TXTAUCTION! Txt word: START to No: 81151 & get yours Now! 4T&Ctxt TC 150p/MTmsg

SMS 23 : Send a logo 2 ur lover - 2 names joined by a heart. Txt LOVE NAME1
NAME2 MOBNO eg LOVE ADAM EVE 07123456789 to 87077 Yahoo! POBox36504W45WQ TxtNO 4
no ads 150p

SMS 44 : Want 2 get laid tonight? Want real Dogging locations sent direct 2 ur
mob? Join the UK's largest Dogging Network bt Txting GRAVEL to 69888! Nt. ec2a.
31p.msg@150p

SMS 47 : BangBabes Ur order is on the way. U SHOULD receive a Service Msg 2
download UR content. If U do not, GoTo wap. bangb. tv on UR mobile
```

internet/service menu

SMS 51 : URGENT! Your Mobile number has been awarded with a £2000 prize GUARANTEED. Call 09058094455 from land line. Claim 3030. Valid 12hrs only

SMS 55 : FREE for 1st week! No1 Nokia tone 4 ur mobile every week just txt NOKIA to 8077 Get txting and tell ur mates. www.getzed.co.uk POBox 36504 W45WQ 16+ norm150p/tone

SMS 56 : FreeMsg Why haven't you replied to my text? I'm Randy, sexy, female and live local. Luv to hear from u. Netcollex Ltd 08700621170150p per msg reply Stop to end

SMS 57 : Congratulations ur awarded 500 of CD vouchers or 125gift guaranteed & Free entry 2 100 wkly draw txt MUSIC to 87066 TnCs www.Ldew.com1win150ppmx3age16

SMS 59 : Congratulations ur awarded either £500 of CD gift vouchers & Free entry 2 our £100 weekly draw txt MUSIC to 87066 TnCs www.Ldew.com1win150ppmx3age16

SMS 80 : URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18

SMS 85 : T-Mobile customer you may now claim your FREE CAMERA PHONE upgrade & a pay & go sim card for your loyalty. Call on 0845 021 3680.Offer ends 28thFeb.T&C's apply

SMS 92 : Orange customer, you may now claim your FREE CAMERA PHONE upgrade for your loyalty. Call now on 0207 153 9996. Offer ends 14thMarch. T&C's apply. Opt-out availa

SMS 93 : PRIVATE! Your 2004 Account Statement for 07742676969 shows 786 unredeemed Bonus Points. To claim call 08719180248 Identifier Code: 45239 Expires

SMS 95 : 22 days to kick off! For Euro2004 U will be kept up to date with the latest news and results daily. To be removed send GET TXT STOP to 83222

SMS 103 : URGENT! Your Mobile No. was awarded £2000 Bonus Caller Prize on 5/9/03 This is our final try to contact U! Call from Landline 09064019788 BOX42WR29C, 150PPM

SMS 105 : England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/ú1.20 POBOXox36504W45WQ 16+

SMS 119 : You have won a guaranteed £200 award or even £1000 cashto claim UR award call free on 08000407165 (18+) 2 stop getstop on 88222 PHP. RG21 4JX

```
SMS 125 : Dear Voucher Holder, 2 claim this weeks offer, at your PC go to
http://www.e-tlp.co.uk/expressoffer Ts&Cs apply.2 stop texts txt STOP to 80062.

SMS 148 : Will u meet ur dream partner soon? Is ur career off 2 a flyng start? 2
find out free, txt HORO followed by ur star sign, e. g. HORO ARIES
```

Let's read now some of the ham messages to check if some of them are false negatives.

[12]:
```
ham_idx = pred_test[pred_test.label=='ham'].index

#print the first 20 messages classified as ham
for i in ham_idx[:20]:
    print('SMS '+ str(i+1) + ' : '+ pred_test.loc[i,'msg'] + '\n')
```

```
SMS 1 : I know that my friend already told that.

SMS 2 : It took Mr owl 3 licks

SMS 3 : Dunno y u ask me.

SMS 4 : K.k:)advance happy pongal.

SMS 5 : I know but you need to get hotel now. I just got my invitation but i had
to apologise. Cali is to sweet for me to come to some english bloke's weddin

SMS 6 : Do you know what Mallika Sherawat did yesterday? Find out now @
&lt;URL&gt;

SMS 7 : Just got up. have to be out of the room very soon. …. i hadn't put the
clocks back til at 8 i shouted at everyone to get up and then realised it was 7.
wahay. another hour in bed.

SMS 8 : Do well :)all will for little time. Thing of good times ahead:

SMS 9 : 8 at the latest, g's still there if you can scrounge up some ammo and
want to give the new ak a try

SMS 11 : Hi Princess! Thank you for the pics. You are very pretty. How are you?

SMS 12 : Not getting anywhere with this damn job hunting over here!

SMS 13 : Good. Good job. I like entrepreneurs

SMS 15 : Hi da:)how is the todays class?

SMS 16 : No calls..messages..missed calls
```

```
SMS 17 : Yo carlos, a few friends are already asking me about you, you working
at all this weekend?

SMS 18 : I'm sorry. I've joined the league of people that dont keep in touch.
You mean a great deal to me. You have been a friend at all times even at great
personal cost. Do have a great week.|

SMS 19 : Haha mayb u're rite… U know me well. Da feeling of being liked by
someone is gd lor. U faster go find one then all gals in our group attached
liao.

SMS 20 : Hmm .. Bits and pieces lol … *sighs* …

SMS 21 : All was well until slightly disastrous class this pm with my fav
darlings! Hope day off ok. Coffee wld be good as can't stay late tomorrow. Same
time + place as always?

SMS 22 : HEY GIRL. HOW R U? HOPE U R WELL ME AN DEL R BAK! AGAIN LONG TIME NO C!
GIVE ME A CALL SUM TIME FROM LUCYxx
```

From the messages listed above, only one looks like spam (SMS 6). So it looks that the detector performed reasonably well.

Conclusion: Looking at some of the messages that are classified as spam, one can say that they really do look like spam, i.e. the SVM model worked pretty well at detecting spam.