NLP (Text) Assignment

by:

- 13515035 - Oktavianus Handika
- 13515075 - Adrian Mulyana Nugraha

In this text-classification algorithm, the sample texts used here are SMS (Short Message Service) texts in Indonesian. The sample data can be found in http://nlp.yuliadi.pro/dataset.

In [7]:

```
from __future__ import division
import numpy as np
import matplotlib.pyplot as plt
import time
import pandas as pd

#Read the train dataset from csv file
train = pd.read_csv("dataset_sms_spam_v1.csv")
train
```

Out[7]:

	Teks	label
0	[PROMO] Beli paket Flash mulai 1GB di MY TELKO...	2
1	2.5 GB/30 hari hanya Rp 35 Ribu Spesial buat A...	2
2	2016-07-08 11:47:11.Plg Yth, sisa kuota Flash ...	2
3	2016-08-07 11:29:47.Plg Yth, sisa kuota Flash ...	2
4	4.5GB/30 hari hanya Rp 55 Ribu Spesial buat an...	2
5	5 HARI LAGI ! EKSTRA Pulsa 50rb dg beli paket ...	2
6	Ada iRing dgn tarif Rp. 0,1/7hr (perpanjangan ...	2
7	Akhir bulan harus tetap eksis loh! Internetan ...	2
8	Aktifkan iRing Coboy Jr - Terhebat. Tekan *808...	2
9	Ambil bonus harianmu di *600# (Bebas Pulsa). D...	2
10	Anda akan berhenti berlangganan Paket Flash. K...	2
11	Anda akan berlangganan paket Rp. 10000 utk 150...	2
12	Anda akan membeli Paket Gampang Internetan Rp....	2
13	Anda akan menerima setting ponsel, agar ponsel...	2
14	Anda akan mengaktifkan Paket BBM Gratis berlak...	2
15	Anda mendapatkan 1 kupon dalam program Kartu A...	2
16	Anda sedang menikmati Paket Reguler dgn sisa k...	2

17	Anda tidak terdaftar dalam layanan Paket Malam...	2
18	Anda tlh menukarkan poin 95 poin, Tukarkan ter...	2
19	AngpaoPoinSenyum! Dptkan Vchr Lottemart 50rb d...	2
20	Awal bulan saat nya anda eksis lebih lama! Int...	2
21	Ayam (sayap/paha bawah), Nasi, Perkedel Rp. 19...	2
22	Ayo anak Medan, beli sticker line dengan pulsa...	2
23	AYO download AXISnet dari Apple/Play Store dan...	2
24	AYO download AXISnet dari PLAY/Apple Store dan...	2
25	Ayo dukung pelestarian alam Indonesia bersama ...	2
26	Ayo kawal pemanfaatan subsidi BBM & Kompensasi...	2
27	BANTING HARGA !! Internetan dgn Kuota 1,5GB HA...	2
28	BEBAS ekspresikan dirimu bersama Paket Freedom...	2
29	Bebas Pulsa! Ambil bonusmu di *600# (GRATIS). ...	2
...	...	...
1113	Waalaikumsalamin apa yg dpt saya bantu min?	0
1114	Wah iya dy?haha sabar bgt anaknya teh jd we ak...	0
1115	wah mantap(ok) btw nge-cc perihal sponsor jg y...	0
1116	wah repot kalo asumsinya gitu, kadang ada yg g...	0
1117	Waktu itu malas sep hihi. Dilihat jd lebih rap...	0
1118	Wid jgn plg sebelum ketemu aku. Aku brgkt jam ...	0
1119	Wihh nyimper kopernya dmn? Motor gigi kan?	0
1120	wisudaan kan hari jumat	0
1121	Wkwk tumben kar biasanya masing2 punya sendiri...	0
1122	Wooh ada yg mau nikah lagi, udah lulus langsun...	0
1123	Ya ampuun pak jendral wk	0
1124	ya masuk aja, belum ada tugas/quiz kok. cuma p...	0
1125	ya nggk jd masalh, toh model bisa diload dicon...	0
1126	Yah saya dikostan temen dil. Hihi	0
1127	yahh masih lama ya, urgent ini wkwk	0
1128	Yang ada waktu luang besok pada futsal ya jam ...	0
1129	Yang sinonim bukan?	0
1130	Yaudah gausah babakaran atuh, yg pake kompor a...	0
1131	Yaudah sekarang mah eta harddisk di laptop man...	0
1132	yaudah, minta data dummy untuk diagnosa_pasien...	0
1133	Yg butuh kosan perbulan bisa langsung ditempat...	0
1134	Yg dian ge waktu itu yudisium akhirnya sore2, ...	0

1135	Yg mau ngampus aku pengen titip bawain SKL aku...	0
1136	Yg ragu sm bulet/datar atau yg pgn ikutan deba...	0
1137	Yg sebelah warteg bahri apa sebrangnya? Yg 15	0
1138	Yooo sama2, oke nanti aku umumin di grup kelas	0
1139	😁 sebelumnya ga ad nulis kerudung. Kirain warn...	0
1140	Mba mau kirim 300 ya	0
1141	nama1 beaok bwrangkat pagi...mau cas atay tra...	0
1142	No bri atas nama kamu mana	0

1143 rows × 2 columns

System Architecture (Modules)

This classification uses scikit-learn library to run the system from preprocessing, feature extraction, and finally the classification itself.

Preprocessing

Preprocessing in this system uses tokenization only to preprocess the train dataset before going to the feature extraction. The library which is used for tokenization is CountVectorizer. This library will convert the documents to a matrix of token counts from the train dataset. To see all the vocabulary in document which was tokenized, we call the function *vocabulary_* of fitted token vector.

Feature Extraction

After preprocessing the dataset into a matrix of token counts, we still have to do feature extraction to eliminate some token that are not very meaningful. We use TF-IDF *(Term Frequency - Inverse Document Frequency)*. This library will summarize how often a given word appears within a document and downscales words that appear a lot across documents.

Dataset Training

After Preprocessing and Feature Extraction from previous steps, dataset will be trained with several learning algorithm with library to test whether a text is a 'Spam' or not spam ('Ham') according to the training dataset, text processing, and its machine learning algorithm that is used. Given some SMS text that will be tested with those classification algorithm.

In [8]:

```python
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer

classifier_label = ['Ham','Spam']
count_vect = CountVectorizer()
train_counts = count_vect.fit_transform(train['Teks'].values)
tf_transformer = TfidfTransformer(use_idf=False).fit(train_counts)
train_tf = tf_transformer.transform(train_counts)

print(train_tf.shape)
```

```
(1143, 4951)
```

In [9]:

```
#Testing the learning method with some input
test = ['Selamat! Anda mendapatkan uang sebesar 100 juta rupiah. Untuk informasi
 lebih lanjut, ' +
        'silakan hubungi nomor berikut +628654321234', #spam
        'Ma, boleh transfer pulsa dulu ke nomor ini? Aku belum bisa isi ulang mas
ih di kampus dulu sekarang', #ham
        'aaa', #ham
        'Transfer saldonya ke rekening ini ya 542 098 7543', #spam
        'Tolong kirim fotocopy KTP dan KK ke email berikut', #ham
        'Registrasi kartumu segera sebelum 1 Oktober 2019', #ham
        'Halo, ada yang bisa dibantu?', #ham
       ]
test_count = count_vect.transform(test)
test_tfidf = tf_transformer.transform(test_count)
```

In [10]:

```
from sklearn.naive_bayes import MultinomialNB
classifier_NB = MultinomialNB().fit(train_tf,train.label)

test_predict = classifier_NB.predict(test_tfidf)

for doc, category in zip(test, test_predict):
    print('%r => %s' % (doc, classifier_label[category]))
```

```
'Selamat! Anda mendapatkan uang sebesar 100 juta rupiah. Untuk info
rmasi lebih lanjut, silakan hubungi nomor berikut +628654321234' =>
Spam
'Ma, boleh transfer pulsa dulu ke nomor ini? Aku belum bisa isi ula
ng masih di kampus dulu sekarang' => Ham
'aaa' => Ham
'Transfer saldonya ke rekening ini ya 542 098 7543' => Ham
'Tolong kirim fotocopy KTP dan KK ke email berikut' => Ham
'Registrasi kartumu segera sebelum 1 Oktober 2019' => Ham
'Halo, ada yang bisa dibantu?' => Ham
```

In [11]:

```
from sklearn.ensemble import RandomForestClassifier
classifier_RF = RandomForestClassifier(n_estimators=200, max_depth=3, random_stat
e=0).fit(train_tf,train.label)

test_predict = classifier_RF.predict(test_tfidf)

for doc, category in zip(test, test_predict):
    print('%r => %s' % (doc, classifier_label[category]))
```

```
'Selamat! Anda mendapatkan uang sebesar 100 juta rupiah. Untuk info
rmasi lebih lanjut, silakan hubungi nomor berikut +628654321234' =>
Ham
'Ma, boleh transfer pulsa dulu ke nomor ini? Aku belum bisa isi ula
ng masih di kampus dulu sekarang' => Ham
'aaa' => Ham
'Transfer saldonya ke rekening ini ya 542 098 7543' => Ham
'Tolong kirim fotocopy KTP dan KK ke email berikut' => Ham
'Registrasi kartumu segera sebelum 1 Oktober 2019' => Ham
'Halo, ada yang bisa dibantu?' => Ham
```

In [12]:

```
from sklearn.svm import LinearSVC
classifier_SVC = LinearSVC().fit(train_tf,train.label)

test_predict = classifier_SVC.predict(test_tfidf)

for doc, category in zip(test, test_predict):
    print('%r => %s' % (doc, classifier_label[category]))
```

```
'Selamat! Anda mendapatkan uang sebesar 100 juta rupiah. Untuk info
rmasi lebih lanjut, silakan hubungi nomor berikut +628654321234' =>
Spam
'Ma, boleh transfer pulsa dulu ke nomor ini? Aku belum bisa isi ula
ng masih di kampus dulu sekarang' => Spam
'aaa' => Ham
'Transfer saldonya ke rekening ini ya 542 098 7543' => Ham
'Tolong kirim fotocopy KTP dan KK ke email berikut' => Spam
'Registrasi kartumu segera sebelum 1 Oktober 2019' => Ham
'Halo, ada yang bisa dibantu?' => Ham
```

Analysis

There are 3 algorithms used in this program:

- Multinomial Naive Bayes Classifier
- Random Forest Classifier
- Linear SVM (Support Vector Classifier)

Comparing the 3 algorithms, Multinomial Naive Bayes achieves 85.7% accuracy (6/7 correct), Random Forest with 71.4% accuracy (5/7 correct), and Linear SVM with 57.1% accuracy (4/7 correct).

This is caused by the nature of each algorithm. Multinomial Naive Bayes matches the words from the test case to the word appearances from the tokenized word bank, when the words match the categories, the chances of the test case entering the matching category increases. Random Forest Classifier uses multiple decision trees trained at the subsets of the data, with a random replacement in the data sets in every iteration. Linear SVM divide 2 classifier (spam and ham) with linear equation line in a vector space, but some data in a classification probably isn't in the area of its cluster.