Spam Email Detection

Fannie Yuen

Hams vs Spams

[spam email sample]:

New Page 1
VIAGRA
WITHOUT
A DOCTORS VISIT!!
CLICK
HERE
*Other
Top Medications also available!!
*We
have Doctors on call around the country to view
your information and quickly approve your order.
*Totally
Discreet System allows you to order today and
enjoy your medication tomorrow in most cases.
*Finally
you can try the wonder drug Viagra that
has swept the World without the embarrassment of
having to visit your Doctor and explain your condition!!
TO
ORDER CLICK HERE!

TO GET DELETED
http://194.44.46.21/remove.php

3606uLdz7-798Gxne6717WLiQ1-104VoKJ8349uvAE9-31143

[ham email sample]:
Joseph S. Barrera III wrote:

> Chris Haun wrote:
>
>> A LifeGem is a certified, high quality diamond created from the
>> carbon of your loved one as a memorial to their unique and wonderful
>> life.
>
>
> Why wait until you're dead? I'm sure there's enough carbon in
> the fat from your typical liposuction job to make a decent diamond.
>
> - Joe
>
Oh, hell - what about excrement? I'd love to be able to say - No, the
sun doesn't shine out of my ass, but there's the occasional diamond. ;-).

Owen

http://xent.com/mailman/listinfo/fork

Workflow

1. Data Reading and Inspection

2. Text Preprocessing

3. Modeling

4. Results & Conclusions

5. Suggestions

Data Reading & Inspection

```
Amount of ham files: 2551
Amount of spam files: 501
Spam to Ham Ratio: 19.64%
Spam to All Ratio: 16.42%
Ham to All Ratio: 83.58%
```

- data source: https://www.kaggle.com/veleon/ham-and-spam-dataset
- packages: email & BeautifulSoup from Python
- files ratio

Text Preprocessing

from email to text format:
get_email_structure(),
structures_counter(), html_to_plain()
and email_to_plain()

feature set 1: stopwords + n-gram (bigram)
+ tf-idf

feature set 2: most-frequent-word-count

both feature sets: removal of punctuations,
lower-casing, word stemming

both feature sets: Compressed Sparse Row
(CSR)

```
stop_words = nltk.corpus.stopwords.words('english')
stop_words
```

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
 'theirs',
 'themselves',
 'what',
 'which',
 'who',
 'whom',
 'this',
 'that',
 "that'll",
 'these',
 'those',
 'am',
 'is',
 'are',
 'was',
 'were',
 'be',
 'been',
 'being',
 'have',
 'has',
 'had',
 'having',
 'do',
 'does',
 'did',
 'doing',
 'a',
 'an',
 'the',
```

…, to be, be or, or not, not to, to be, …

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

	Doc 1	Doc 2	...	Doc *n*
Term(s) 1	12	2	...	1
Term(s) 2	0	1	...	0
...	...	...	...	
Term(s) *n*	0	6	...	3

```
[Counter({'the': 15, 'pick': 9, '-lbrace': 6, 'of': 5, '-rbrace': 5, 'i': 4, 'is': 4, '-
list': 4, 'thi': 3, '+inbox': 3, '-subject': 3, 'ftp': 3, '-sequenc': 3, '18:19:04': 3,
'command': 3, 'delta$': 3, 'from': 3, 'error': 2, '18:19:03': 2, '4852-4852': 2, 'mercur
i': 2, '1': 2, 'hit': 2, "that'": 2, 'come': 2, 'version': 2, 'use': 2, 'on': 2, 'and':
2, 'one': 2, 'date:': 1, 'wed': 1, '21': 1, 'aug': 1, '2002': 1, '10:54:46': 1, '-0500':
1, 'from:': 1, 'chri': 1, 'garrigu': 1, '<cwg-dated-103037728706fa6d@deepeddycom>': 1, '
message-id:': 1, '<10299452874797tmda@deepeddyvirciocom>': 1, '|': 1, "can't": 1, 'repro
duc': 1, 'for': 1, 'me': 1, 'it': 1, 'veri': 1, 'repeat': 1, '(like': 1, 'everi': 1, 'ti
me': 1, 'without': 1, 'fail)': 1, 'debug': 1, 'log': 1, 'happen': 1, 'pick_it': 1, '{exe
c': 1, '-rbrace}': 1, '{4852-4852': 1, 'mercury}': 1, 'exec': 1, 'ftoc_pickmsg': 1, '{{1
': 1, 'hit}}': 1, 'mark': 1, 'tkerror:': 1, 'syntax': 1, 'in': 1, 'express': 1, '"int':
1, 'note': 1, 'if': 1, 'run': 1, 'by': 1, 'hand': 1, 'where': 1, '"1': 1, 'hit"': 1, '(o
bviously)': 1, 'nmh': 1, "i'm": 1, '-version': 1, '--': 1, 'nmh-104': 1, '[compil': 1, '
fuchsiacsmuozau': 1, 'at': 1, 'sun': 1, 'mar': 1, '17': 1, '14:55:56': 1, 'ict': 1, '200
2]': 1, 'relev': 1, 'part': 1, 'my': 1, 'mh_profil': 1, 'mhparam': 1, '-seq': 1, 'sel':
1, 'sinc': 1, 'work': 1, 'sequenc': 1, '(actual': 1, 'both': 1, 'them': 1, 'explicit': 1
, 'line': 1, 'search': 1, 'popup': 1, 'that': 1, 'mh_profile)': 1, 'do': 1, 'get': 1, 'c
reat': 1, 'kre': 1, 'ps:': 1, 'still': 1, 'code': 1, 'form': 1, 'a': 1, 'day': 1, 'ago':
1, "haven't": 1, 'been': 1, 'abl': 1, 'to': 1, 'reach': 1, 'cv': 1, 'repositori': 1, 'to
```

Living => live
Live => live
Lives => live
Lived => live

$$A = \begin{pmatrix} 10 & 0 & 0 & 12 & 0 \\ (0,0) & & & (0,3) & \\ 0 & 0 & 11 & 0 & 13 \\ & & (1,2) & & (1,4) \\ 0 & 16 & 0 & 0 & 0 \\ & (2,1) & & & \\ 0 & 0 & 11 & 0 & 13 \\ & & (3,2) & & (3,4) \end{pmatrix}$$

$$val = \begin{pmatrix} 10 & 12 & 11 & 13 & 16 & 11 & 13 \\ (0,0) & (0,3) & (1,2) & (1,4) & (2,1) & (3,2) & (3,4) \end{pmatrix}$$

$$colInd = \begin{pmatrix} 0 & 3 & 2 & 4 & 1 & 2 & 4 \\ (0 &) & (1 &) & (2) & (3 &) \end{pmatrix}$$

$$rowPtr = \begin{pmatrix} 0 & 2 & 4 & 5 & 7 \\ (0) & (1) & (2) & (3) & (4) \end{pmatrix}$$

Modeling

D1: "send us your password" spam
D2: "send us your review" ham
D3: "review your password" ham
D4: "review us" spam
D5: "send your password" spam
D6: "send us your account" spam

new email: "review us now"

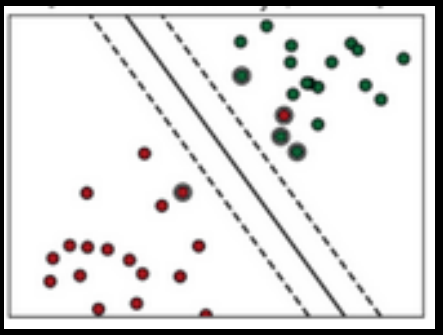
P (spam) = 4/6 P (ham) = 2/6

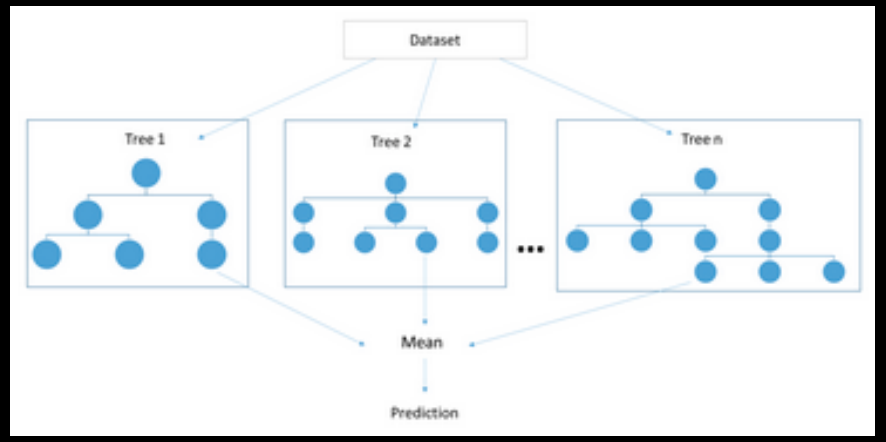
spam	ham	
2/4	1/2	password
1/4	2/2	review
3/4	1/2	send
3/4	1/2	us
3/4	1/2	your
1/4	0/2	account

$$P(\text{review us}\,|\text{spam}) = P(0,1,0,1,0,0\,|\text{spam}) = (1-\tfrac{2}{4})(\tfrac{1}{4})(1-\tfrac{3}{4})(\tfrac{3}{4})(1-\tfrac{3}{4})(1-\tfrac{1}{4})$$

$$P(\text{review us}\,|\text{ham}) = P(0,1,0,1,0,0\,|\text{ham}) = (1-\tfrac{1}{2})(\tfrac{2}{2})(1-\tfrac{1}{2})(\tfrac{1}{2})(1-\tfrac{1}{2})(1-\tfrac{0}{2})$$

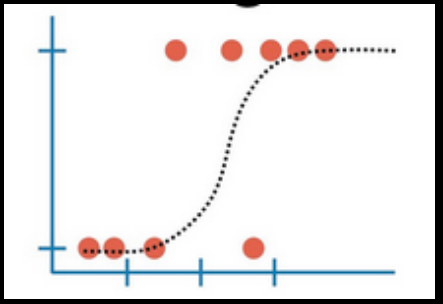
$$P(\text{ham}|\text{review us}) = \frac{0.0625\times 2/6}{0.0625\times 2/6 + 0.0044\times 4/6} = 0.87$$ (note identical example)



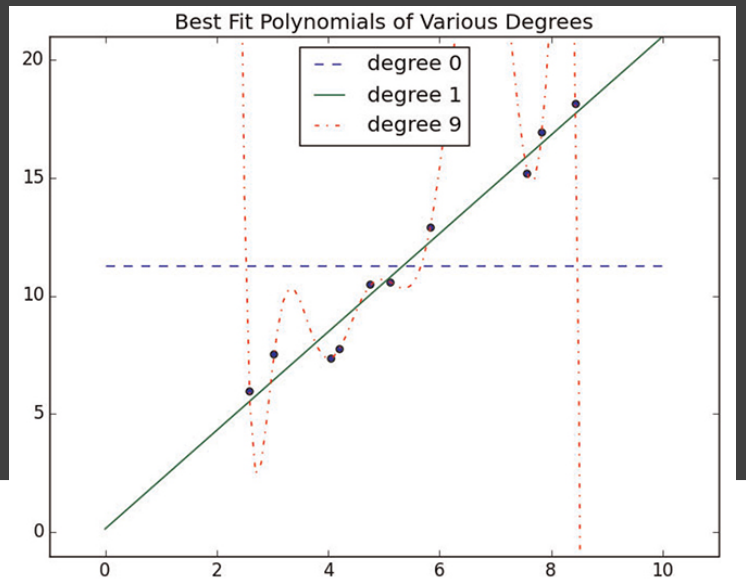


- Dummy
  - stratified
  - most_frequent
  - constant
- Logistic Regression
- Multinomial Naïve Bayes
- Support Vector Machine - Linear Kernel
- Random Forest

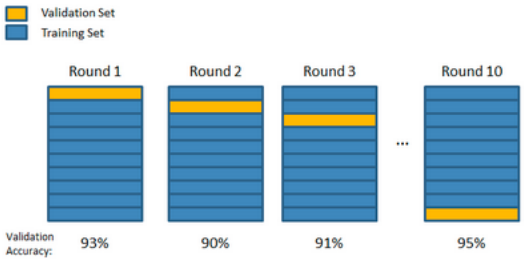
Stratified	Most Frequent	Constant
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
1	0	1



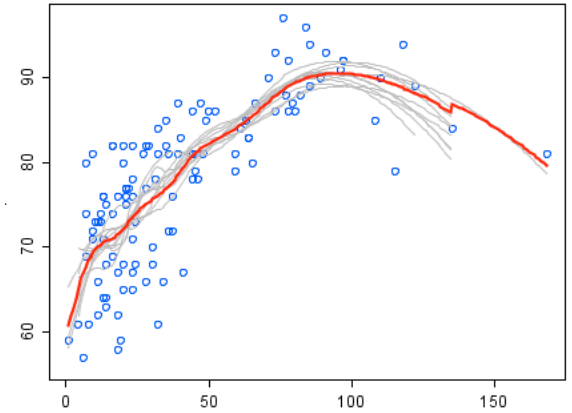
Under- & Overfitting



dataset splitting



CV-score



bootstrap

Evaluation Metrics

accuracy

precision

recall

F1

Results & Conclusions

```
1 model_result_dummy
```

	feature	model_name	cv_score_mean	cv_score_std	accuracy	precision	recall	F1
0	-	Dummy_Stratified	0.7432	0.0177	0.7381	0.2222	0.24	0.2308
1	-	Dummy_Frequent	0.8357	0.0010	0.8363	0.0000	0.00	0.0000
2	-	Dummy_Constant	0.1643	0.0010	0.1637	0.1637	1.00	0.2813

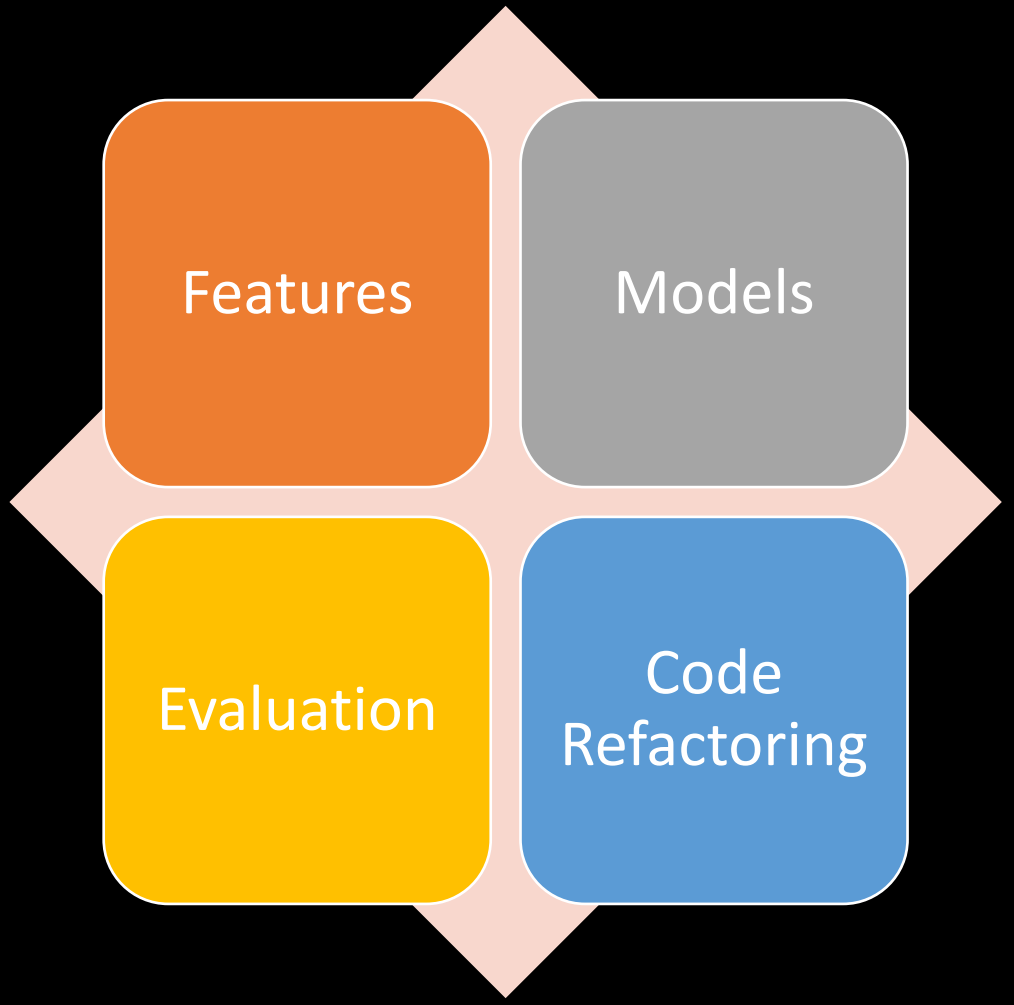
```
1 model_result_wc
```

	feature	model_name	cv_score_mean	cv_score_std	accuracy	precision	recall	F1
0	word-count	NB_Multinomial	0.9803	0.0073	0.9787	0.9579	0.91	0.9333
1	word-count	LR	0.9865	0.0080	0.9885	0.9895	0.94	0.9641
2	word-count	RF	0.9767	0.0082	0.9755	0.9570	0.89	0.9223
3	word-count	SVM	0.9791	0.0097	0.9853	0.9505	0.96	0.9552

```
1 model_result_stopword_ngram_tdidf
```

	feature	model_name	cv_score_mean	cv_score_std	accuracy	precision	recall	F1
0	stopword + n-gram + td-idf	NB_Multinomial	0.9222	0.0162	0.9394	1.0000	0.63	0.7730
1	stopword + n-gram + td-idf	LR	0.8849	0.0094	0.8903	1.0000	0.33	0.4962
2	stopword + n-gram + td-idf	RF	0.9062	0.0171	0.8887	0.6159	0.85	0.7143
3	stopword + n-gram + td-idf	SVM	0.9509	0.0153	0.9591	1.0000	0.75	0.8571

Suggestions



Thank You!