CTU

CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

Spam filter semestral project

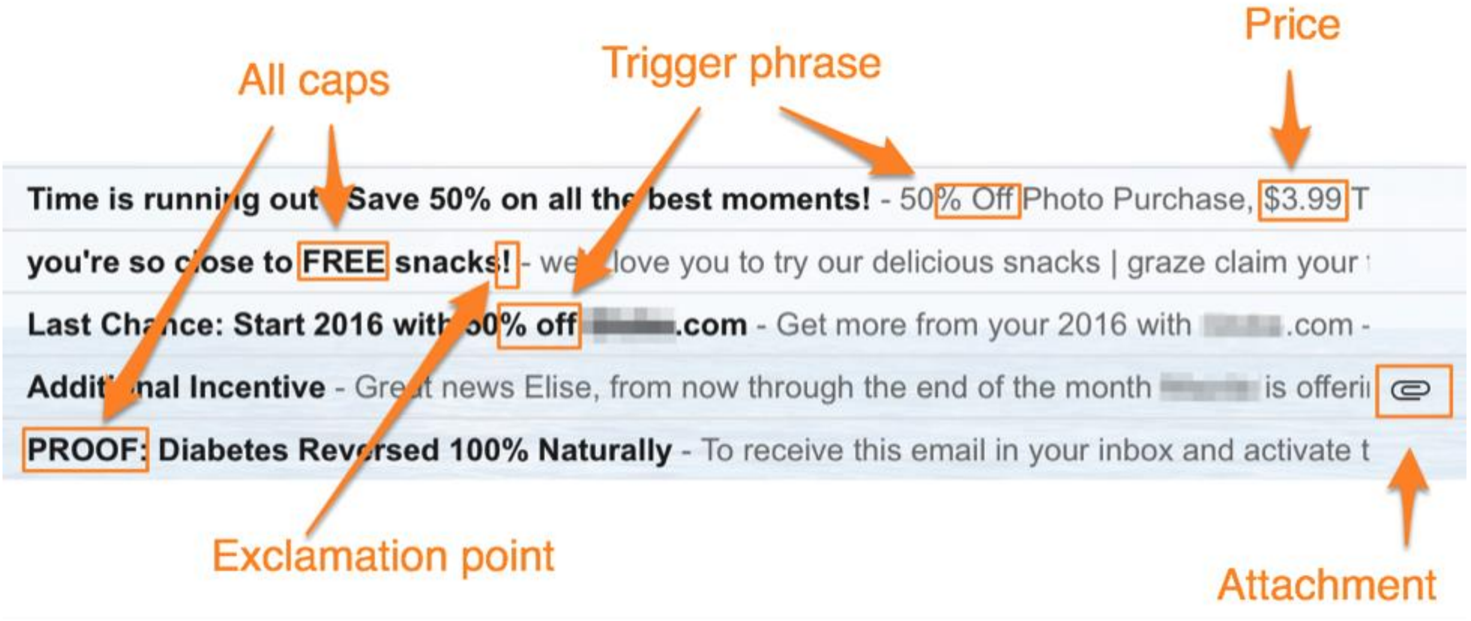
be5b33prg – Programming Essentials

Milan Němý
Czech Technical University in Prague,
Department of Cybernetics
milan.nemy@cvut.cz, nemymila@fel.cvut.cz

November 14, 2019

Spam

- What is spam?
- What is a spam filter? How does it work?
- What makes spam spam?
- Can a spam filter produce errors? Are all equally serious?



Time is running out: Save 50% on all the best moments! - 50% Off Photo Purchase, $3.99 T

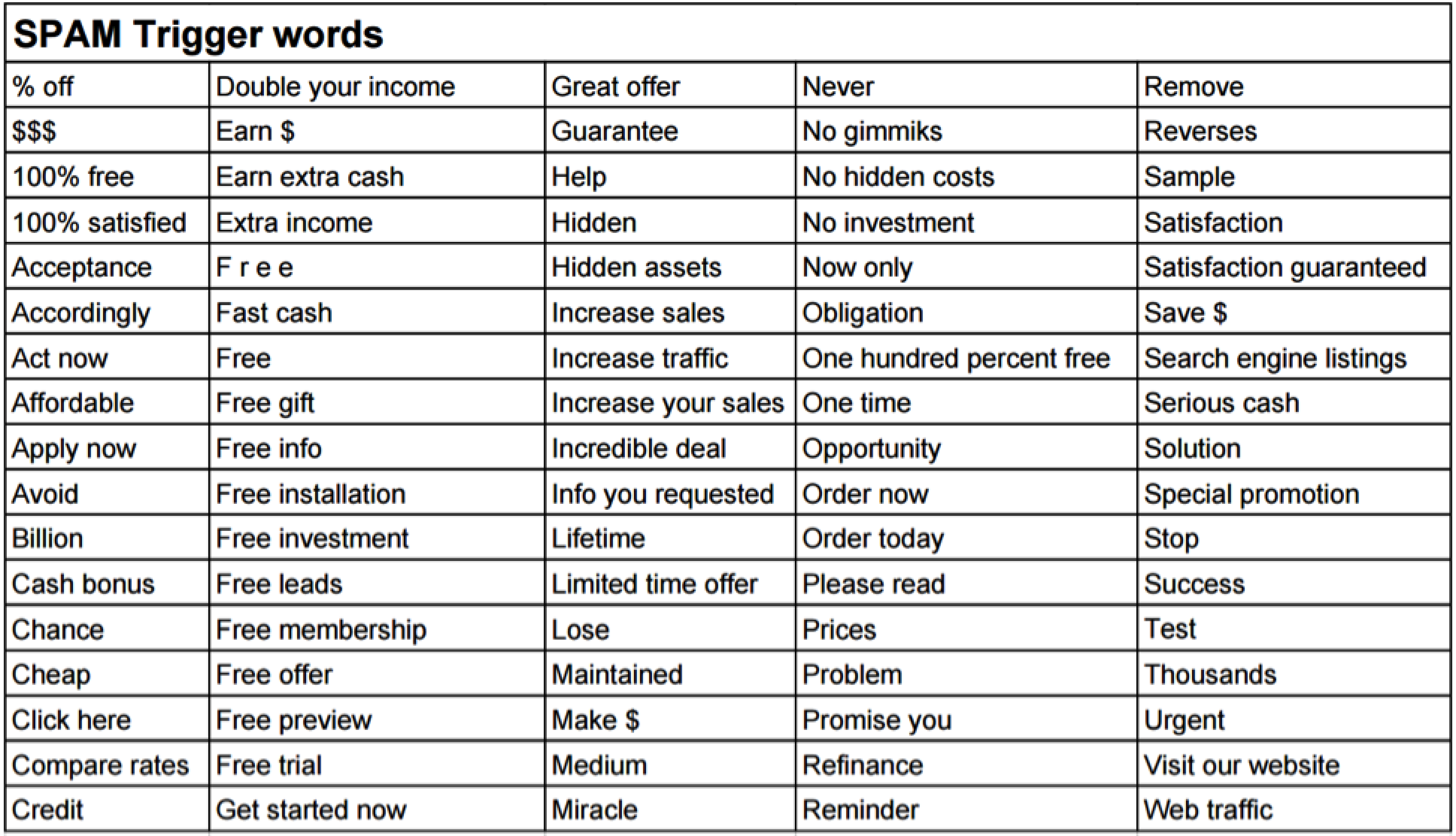
you're so close to FREE snacks! - we'd love you to try our delicious snacks | graze claim your

Last Chance: Start 2016 with 50% off [redacted].com - Get more from your 2016 with [redacted].com -

Additional Incentive - Great news Elise, from now through the end of the month [redacted] is offeri

PROOF: Diabetes Reversed 100% Naturally - To receive this email in your inbox and activate t

Spam characteristics

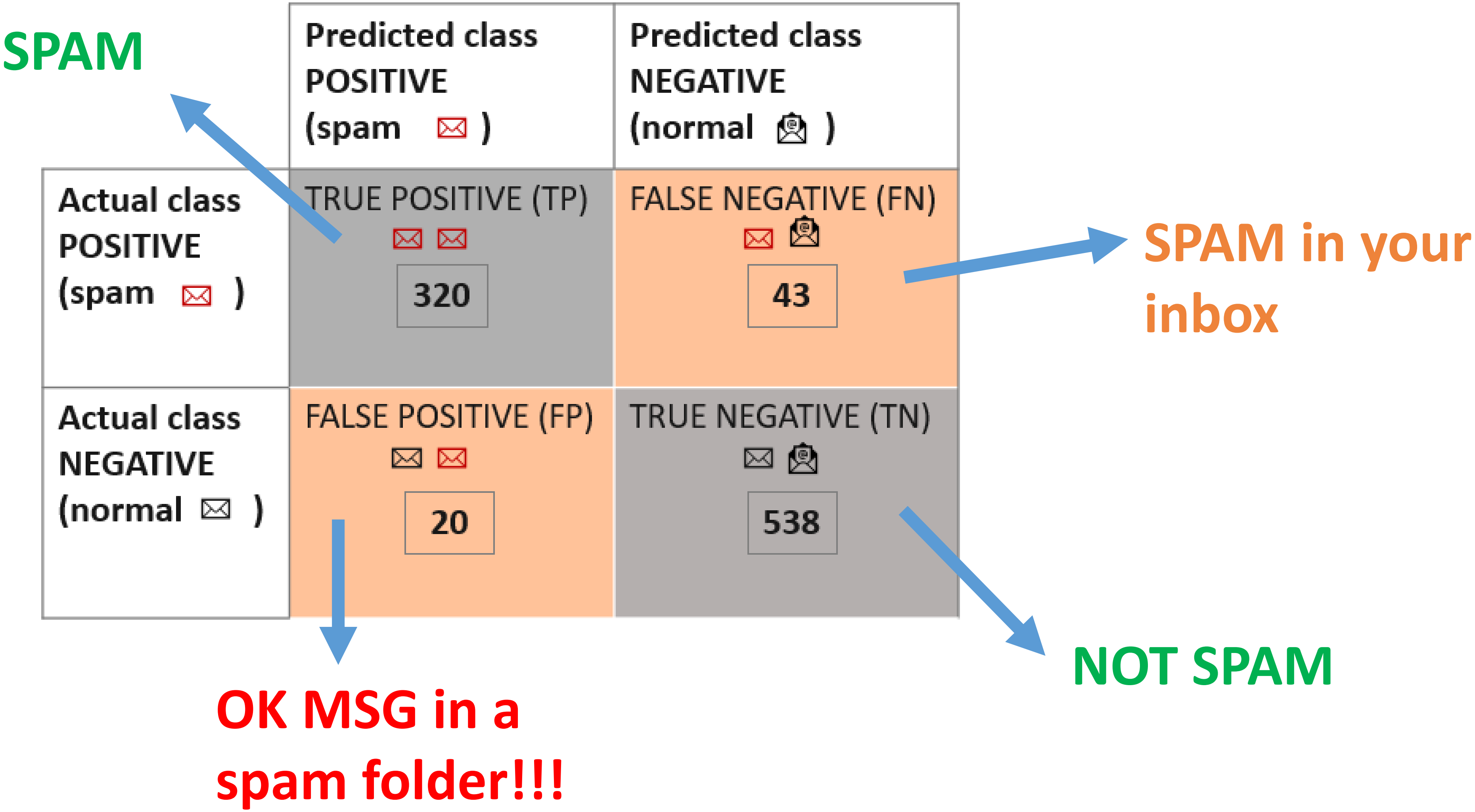


SPAM Trigger words

% off	Double your income	Great offer	Never	Remove
$$$	Earn $	Guarantee	No gimmiks	Reverses
100% free	Earn extra cash	Help	No hidden costs	Sample
100% satisfied	Extra income	Hidden	No investment	Satisfaction
Acceptance	F r e e	Hidden assets	Now only	Satisfaction guaranteed
Accordingly	Fast cash	Increase sales	Obligation	Save $
Act now	Free	Increase traffic	One hundred percent free	Search engine listings
Affordable	Free gift	Increase your sales	One time	Serious cash
Apply now	Free info	Incredible deal	Opportunity	Solution
Avoid	Free installation	Info you requested	Order now	Special promotion
Billion	Free investment	Lifetime	Order today	Stop
Cash bonus	Free leads	Limited time offer	Please read	Success
Chance	Free membership	Lose	Prices	Test
Cheap	Free offer	Maintained	Problem	Thousands
Click here	Free preview	Make $	Promise you	Urgent
Compare rates	Free trial	Medium	Refinance	Visit our website
Credit	Get started now	Miracle	Reminder	Web traffic

Spam

$$q = \frac{TP+TN}{TP+TN+10\cdot FP+FN}$$



	Predicted class POSITIVE (spam ✉)	Predicted class NEGATIVE (normal 📩)
Actual class POSITIVE (spam ✉)	TRUE POSITIVE (TP) ✉ ✉ 320	FALSE NEGATIVE (FN) ✉ 📩 43
Actual class NEGATIVE (normal ✉)	FALSE POSITIVE (FP) ✉ ✉ 20	TRUE NEGATIVE (TN) ✉ 📩 538

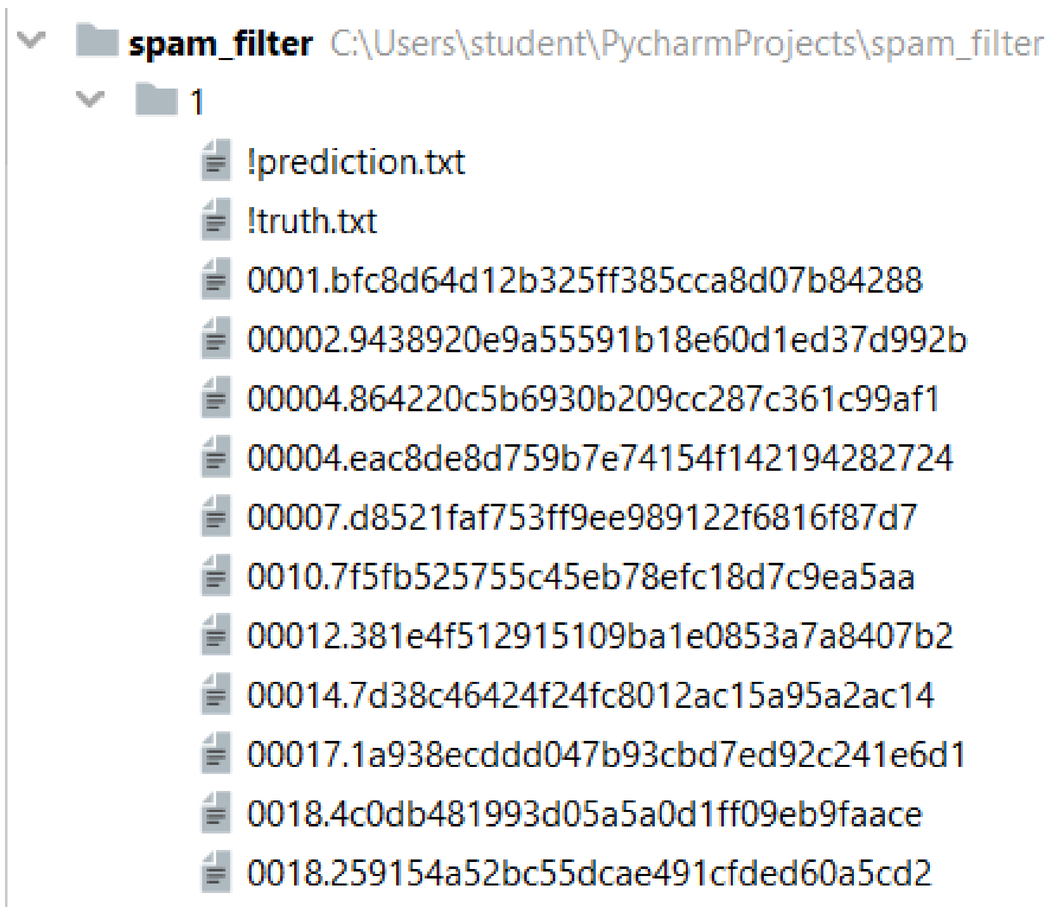
SPAM

SPAM in your inbox

OK MSG in a spam folder!!!

NOT SPAM

Input data



```
spam_filter C:\Users\student\PycharmProjects\spam_filter
  1
    !prediction.txt
    !truth.txt
    0001.bfc8d64d12b325ff385cca8d07b84288
    00002.9438920e9a55591b18e60d1ed37d992b
    00004.864220c5b6930b209cc287c361c99af1
    00004.eac8de8d759b7e74154f142194282724
    00007.d8521faf753ff9ee989122f6816f87d7
    0010.7f5fb525755c45eb78efc18d7c9ea5aa
    00012.381e4f512915109ba1e0853a7a8407b2
    00014.7d38c46424f24fc8012ac15a95a2ac14
    00017.1a938ecddd047b93cbd7ed92c241e6d1
    0018.4c0db481993d05a5a0d1ff09eb9faace
    0018.259154a52bc55dcae491cfded60a5cd2
```

2 sets of data available for training and local testing

!prediction.txt (created by your code)

```
00056.6647a720da7dad641f4028c9f6fbf4e5 ???
00058.64bb1902c4e561fb3e521a6dbf8625be ???
00060.ec71d52a6f585ace52f4a2a2be2adfce ???
00063.2334fb4e465fc61e8406c75918ff72ed ???
00065.9c8ae6822b427f2dbee5339d561a2888 ???
```

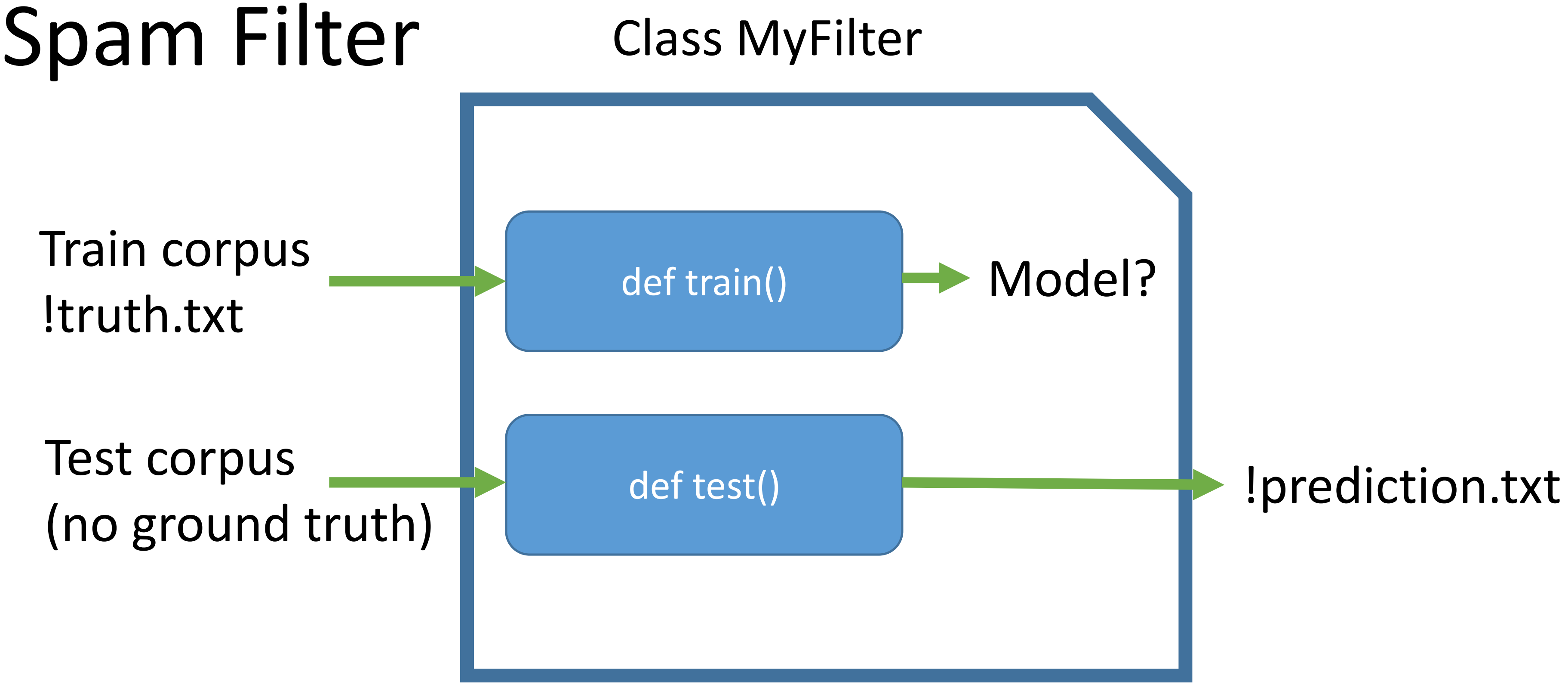
!truth.txt

```
00026.1757d50d495d41e8a5eb30a2f371019c SPAM
00026.da18dbed27ae933172f7a70f860c6ad0 OK
00029.cc0c62b49c1df0ad08ae49a7e1904531 SPAM
00031.e50cc5af8bd1131521b551713370a4b1 OK
00034.8e582263070076dfe6000411d9b13ce6 OK
```

2391.40efcd4ee4a50355cfe8a84c327122c1

```
To: yyyy@example.com
From: boingboing <rssfeeds@example.com>
Subject: Sidekick's browser blows
Date: Sat, 05 Oct 2002 08:00:33 -0000
(...)
```

Spam Filter



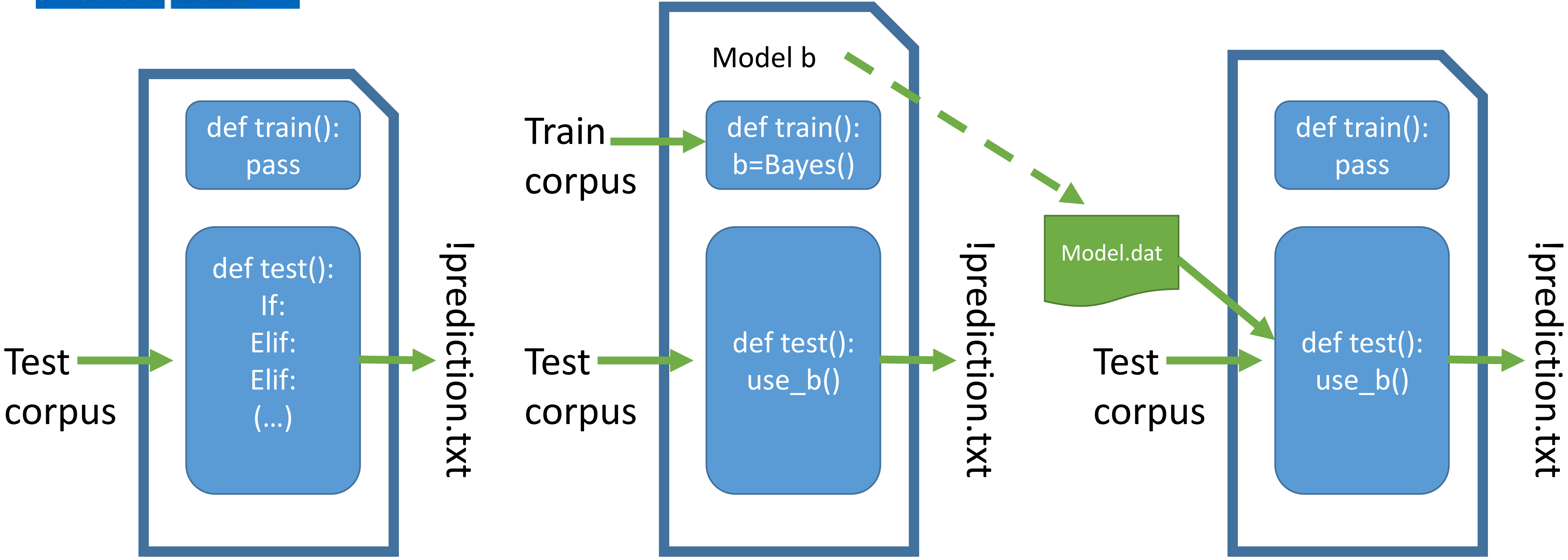
How good is your prediction?

1. Use corpus #1 for training and corpus #2 for testing
2. Split the training corpus into a training (60%-80%) and testing (20%-40%) subset

DO NOT use the exact same corpus for training and testing!

You will end up with a biased estimate of testing error!

Spam Filter



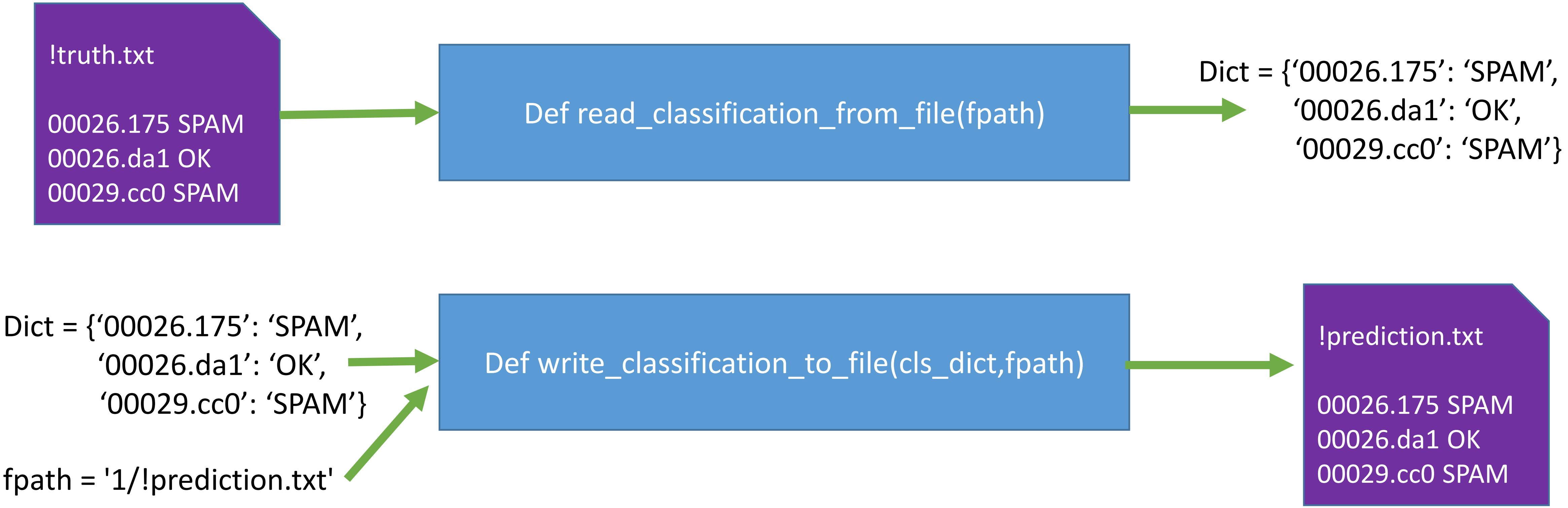
Hardwired filter

2 pts + low accuracy

Learning filter
(5 min timeout)
4 pts + high accuracy

Pre-learned filter
(time-consuming
training phase?)

Submission 1 – step 1



!truth.txt

00026.175 SPAM
00026.da1 OK
00029.cc0 SPAM

Def read_classification_from_file(fpath)

Dict = {‘00026.175’: ‘SPAM’,
‘00026.da1’: ‘OK’,
‘00029.cc0’: ‘SPAM’}

Dict = {‘00026.175’: ‘SPAM’,
‘00026.da1’: ‘OK’,
‘00029.cc0’: ‘SPAM’}

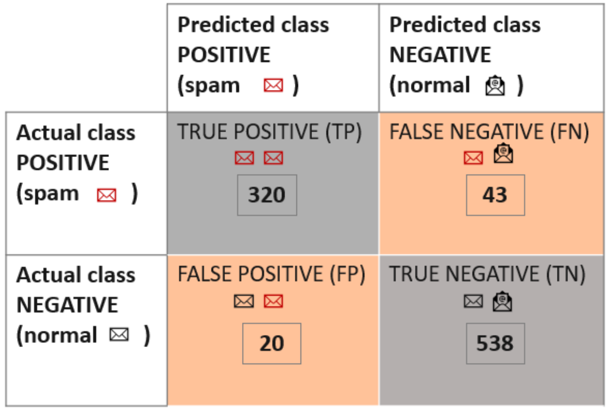
fpath = '1/!prediction.txt'

Def write_classification_to_file(cls_dict,fpath)

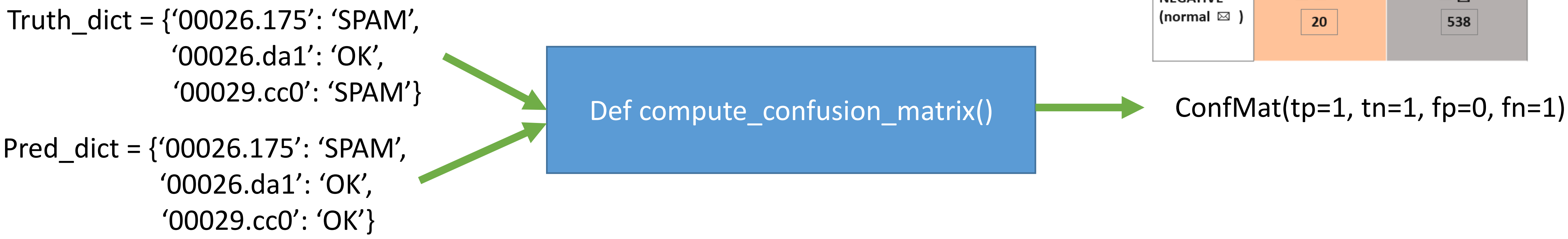
!prediction.txt

00026.175 SPAM
00026.da1 OK
00029.cc0 SPAM

Submission 1 – step 2&3



	Predicted class POSITIVE (spam ✉)	Predicted class NEGATIVE (normal ✉)
Actual class POSITIVE (spam ✉)	TRUE POSITIVE (TP) ✉ ✉ 320	FALSE NEGATIVE (FN) ✉ ✉ 43
Actual class NEGATIVE (normal ✉)	FALSE POSITIVE (FP) ✉ ✉ 20	TRUE NEGATIVE (TN) ✉ ✉ 538

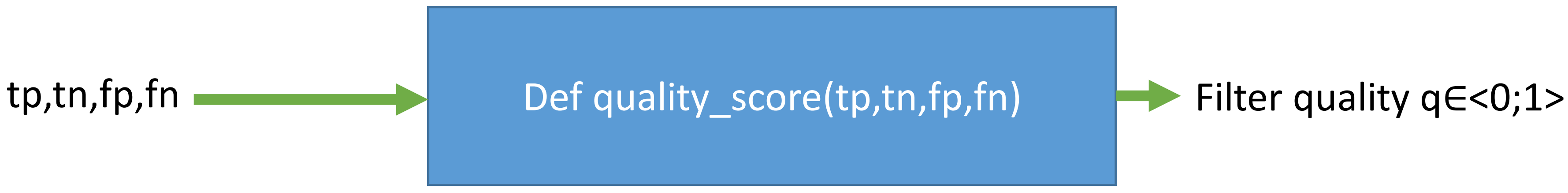


Truth_dict = {‘00026.175’: ‘SPAM’,
‘00026.da1’: ‘OK’,
‘00029.cc0’: ‘SPAM’}

Pred_dict = {‘00026.175’: ‘SPAM’,
‘00026.da1’: ‘OK’,
‘00029.cc0’: ‘OK’}

Def compute_confusion_matrix()

ConfMat(tp=1, tn=1, fp=0, fn=1)



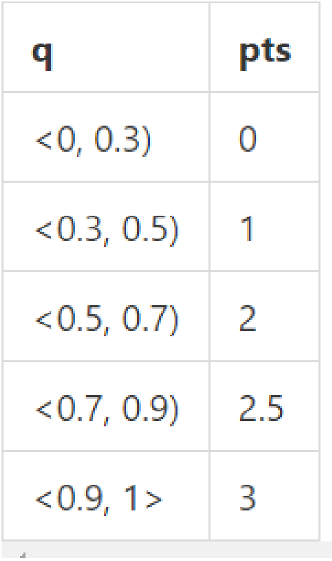
tp,tn,fp,fn

Def quality_score(tp,tn,fp,fn)

Filter quality q∈<0;1>

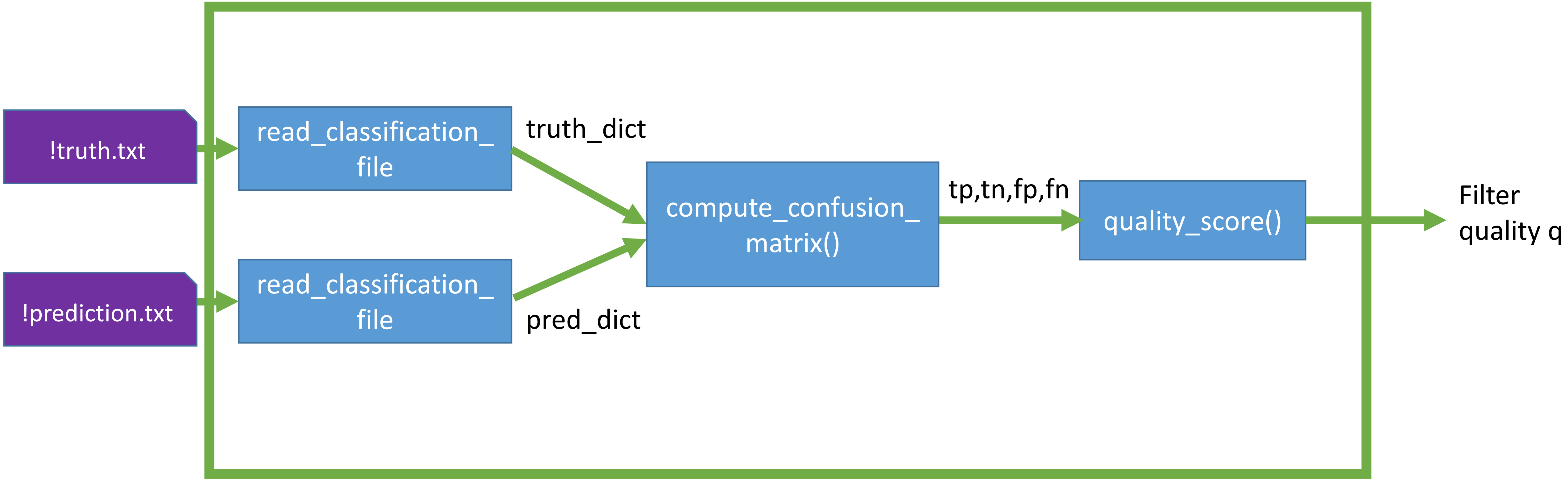
$$q = \frac{TP+TN}{TP+TN+10 \cdot FP+FN}$$

3 datasets



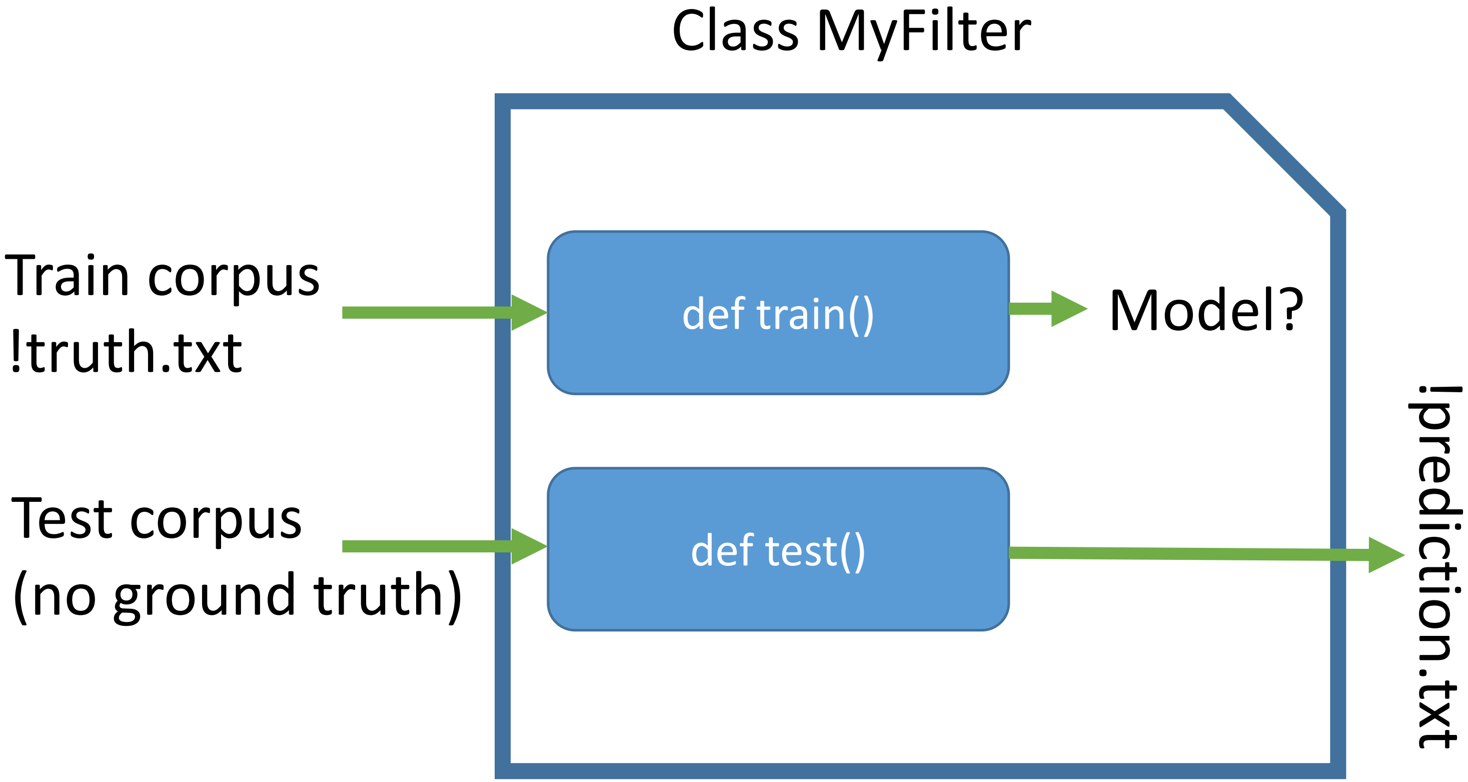
q	pts
<0, 0.3)	0
<0.3, 0.5)	1
<0.5, 0.7)	2
<0.7, 0.9)	2.5
<0.9, 1>	3

Submission 1 – steps 1-3 (recommended structure)



def compute_quality_for_corpus()

Submission 2 – steps 4&5



Try simple filters first:
(no training phase)

1. Naïve Filter – all OK
2. Paranoid Filter – all SPAM
3. Random Filter – randomly assigns label

This step won't be evaluated.

Implement your own MyFilter class:

1. Hard-wired rules (if-else, switch, trigger words, …)
2. Machine learning techniques
   1. **K-nearest neighbours (cosine similarity)**
   2. **Bayesian decision making**
   3. Perceptron
   4. Support Vector Machines
   5. Neural Networks

Submission 2 – steps 4&5

Possible text pre-processing:

1. Tokenization
2. Lower case
3. Remove punctuation
4. Remove stop words
5. Stemming

The cat is sitting on the mat!

{The, cat, is, sitting, on, the, mat!}

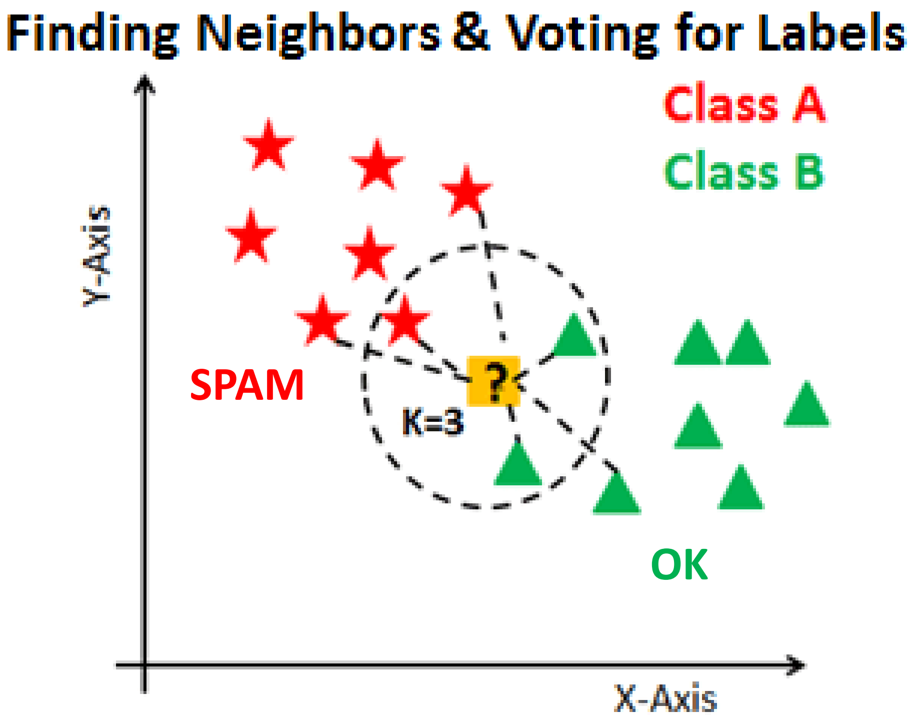
{the, cat, is, sitting, on, the, mat!}

{the, cat, is, sitting, on, the, mat}

{cat, sitting, mat}

{cat, sit, mat}

Submission 2 – steps 4&5 examples of classifiers



K-nearest neighbours (k-NN)

- Easy to implement
- Distance function?
- Must story all training data

$$\mathbf{P(\,spam \mid W_1, W_2, \ldots, W_n\,)} = \frac{P(\,W_1, W_2, \ldots, W_n \mid spam\,)\,P(spam)}{P(W_1, W_2, \ldots, W_n)}$$

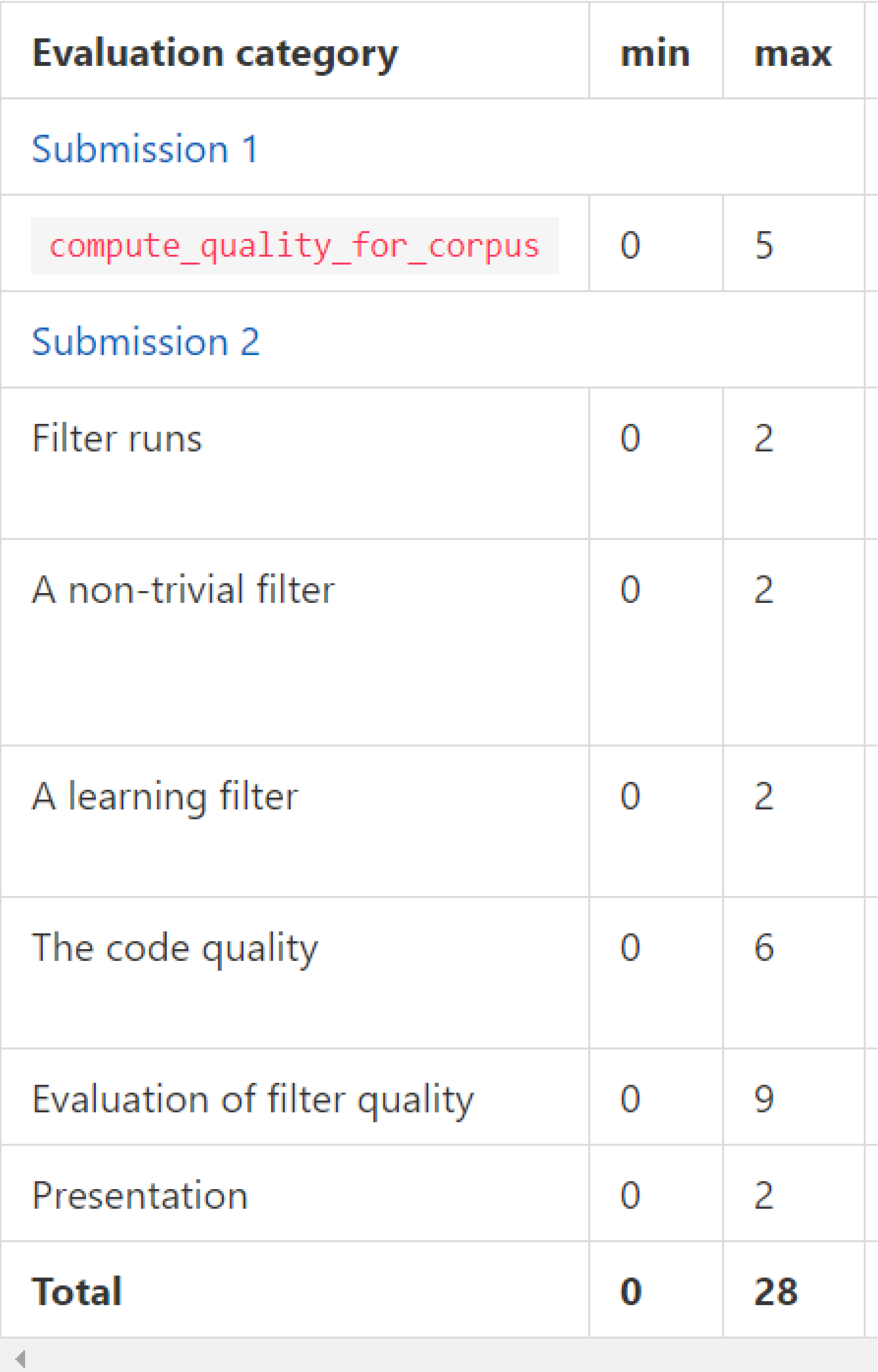
$$\mathbf{P(\,not\ spam \mid W_1, W_2, \ldots, W_n\,)} = \frac{P(\,W_1, W_2, \ldots, W_n \mid not\ spam\,)\,P(not\ spam)}{P(W_1, W_2, \ldots, W_n)}$$

Bayesian decision making

- Probability model
- Can be tricky
- Store probabilities
- ML ready-to-use modules

Spam filter project

- Important dates
  - Part 1: due Dec 6, 2019 (5 pts)
  - Part 2: due Jan 9, 2020 (23 pts)
  - Short presentation: Jan 10, 2020 (during PC labs) or any earlier PC lab
- All info here: https://cw.fel.cvut.cz/wiki/courses/be5b33prg/homeworks/spam/start
- come to PC lab 10-13 and/or work on your own



Evaluation category	**min**	**max**
Submission 1		
`compute_quality_for_corpus`	0	5
Submission 2		
Filter runs	0	2
A non-trivial filter	0	2
A learning filter	0	2
The code quality	0	6
Evaluation of filter quality	0	9
Presentation	0	2
Total	**0**	**28**