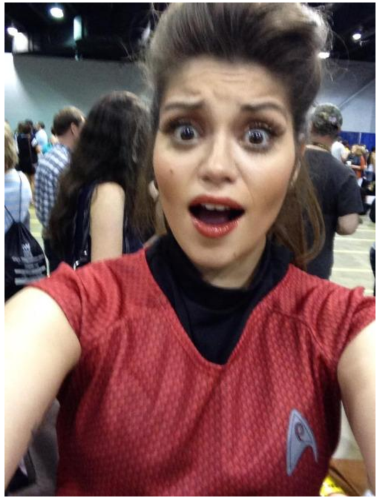
Is that spam in my ham?

A novice's inquiry into classification.

Lorena Mesa | EuroPython 2016
@loooorenanicole
bit.ly/europython2016-lmesa

Hi, I’m Lorena Mesa.



In reply to Đave Hoover
Lorena Mesa @loooorenanicole · 7h
@davehoover and we are super thankful!!!! Look at these Squirrels 2014!! #dbc @devbootcamp

SYSTERS
AN ANITA BORG INSTITUTE COMMUNITY

<write/speak/code>

django

python SOFTWARE FOUNDATION

sproutsocial

pyladies CHICAGO

Have you seen this before? *(You're not alone.)*

Subject:

De-junk And Speed Up Your Slow PC!!!

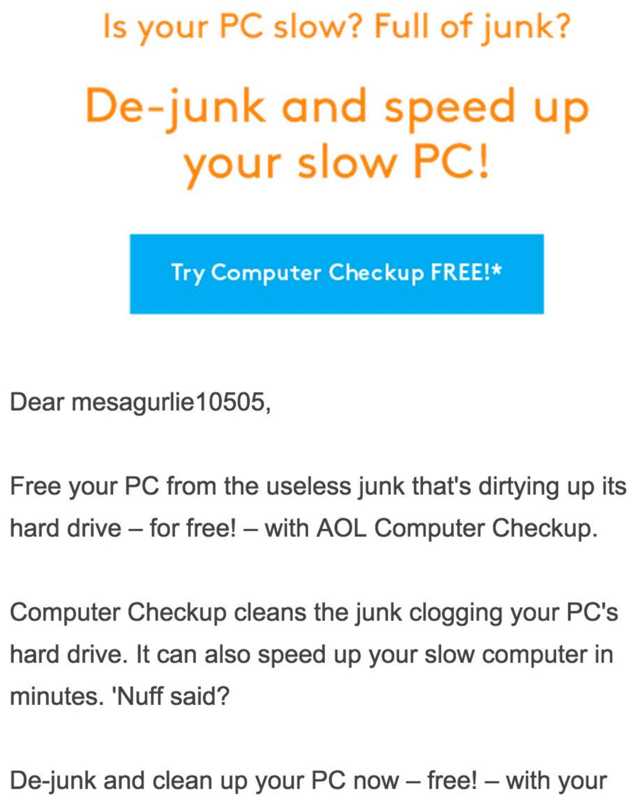
From:

AOL_MemberInfo@emailz.aol.com

Theme:

Promises of "free" item(s).

Several images in the email itself.


Is your PC slow? Full of junk?

De-junk and speed up your slow PC!

Try Computer Checkup FREE!*

Dear mesagurlie10505,

Free your PC from the useless junk that's dirtying up its hard drive – for free! – with AOL Computer Checkup.

Computer Checkup cleans the junk clogging your PC's hard drive. It can also speed up your slow computer in minutes. 'Nuff said?

De-junk and clean up your PC now – free! – with your

How I’ll approach today’s chat.

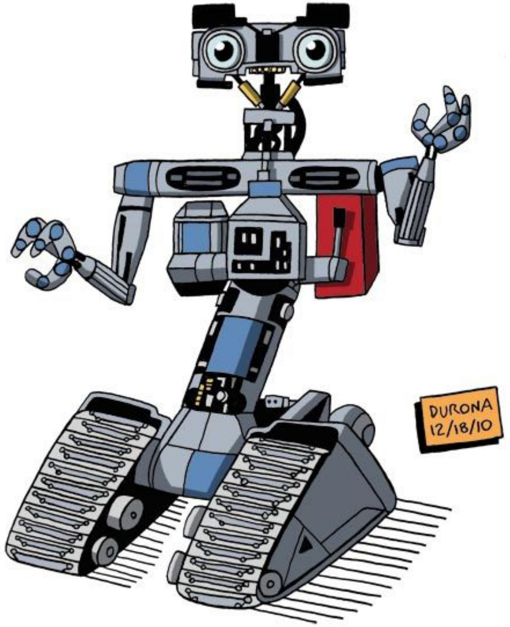
1. What is machine learning?
2. How is classification a part of this world?
3. How can I use Python to solve a classification problem like spam detection?

😕

🤖

📒 📚 🔤 🔟

⁉️


DURONA
12/18/10

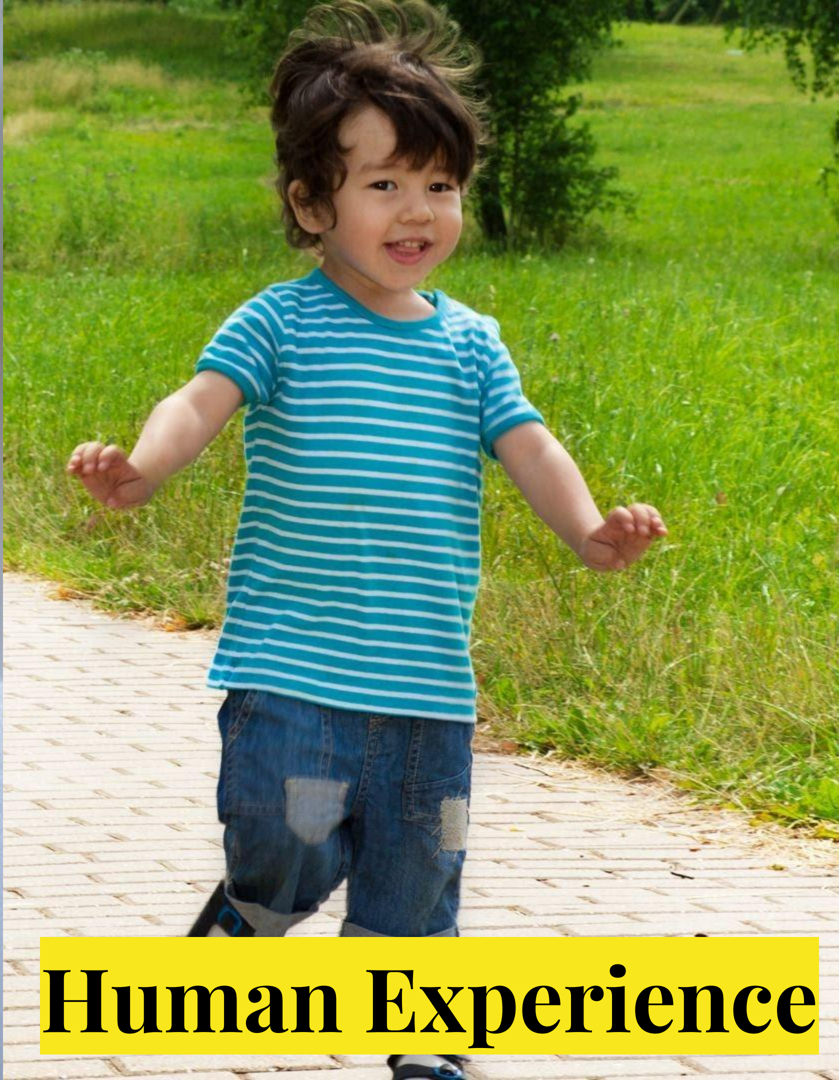
Machine Learning

is a subfield of computer science [that] stud[ies] pattern recognition and computational learning [in] artificial intelligence. [It] explores the construction and study of **algorithms** that can learn from and make **predictions on data**.

Put another way

A computer program is **said to learn** from **experience** (E) with respect to some **task** (T) and some performance **measure** (P), if its performance on T, as measured by P, improves with experience E.

(Ch. 1 - Machine Learning Tom Mitchell)



Human Experience

Recorded Experience

Classification in machine learning

Task: Classify a piece of data

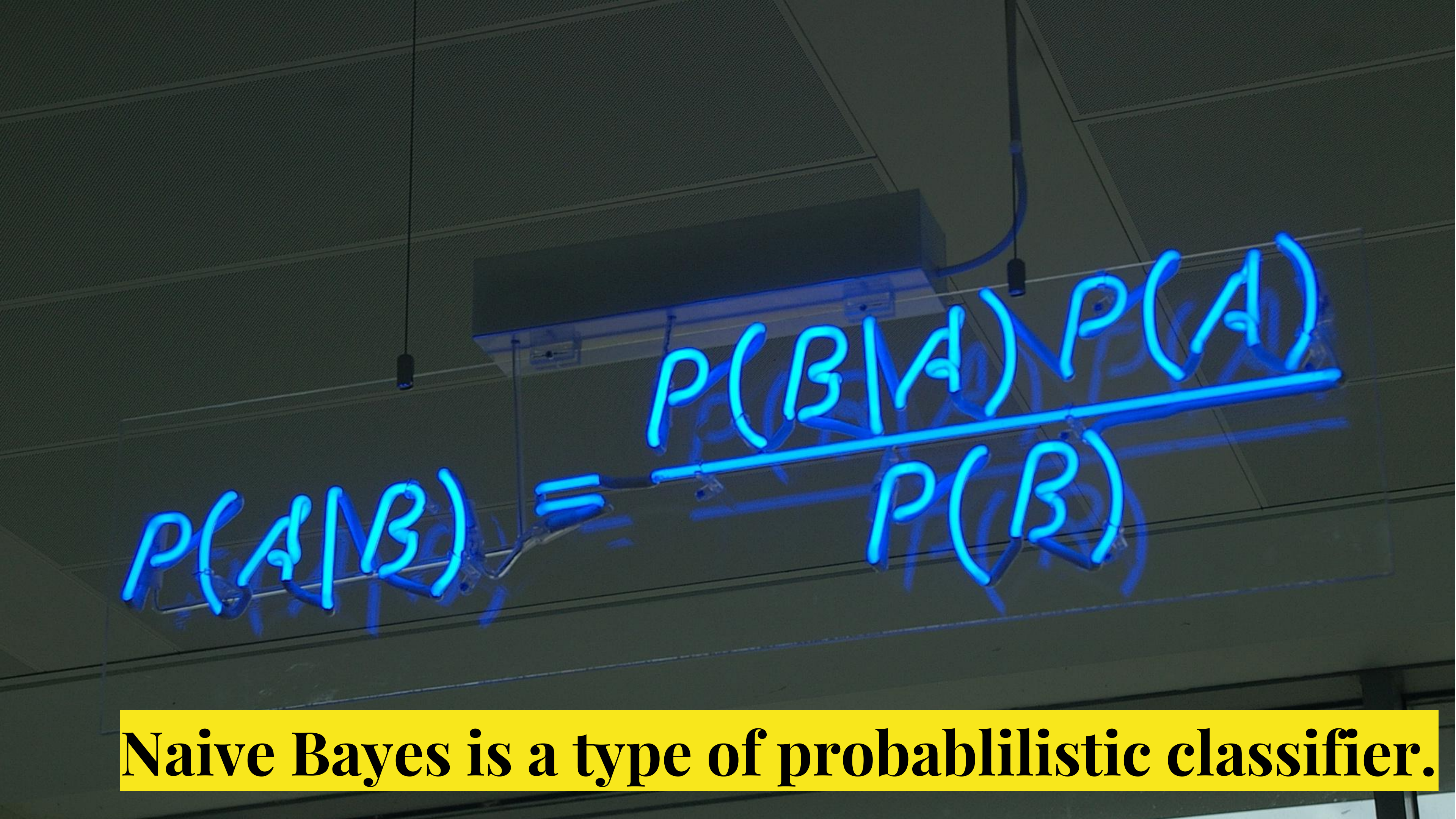
Is an email Spam or Ham?

Experience: Labeled training data

Email 1 | Ham
Email 2 | Spam

Performance Measurement: Is the label correct?

Verify if the email is Spam or Ham

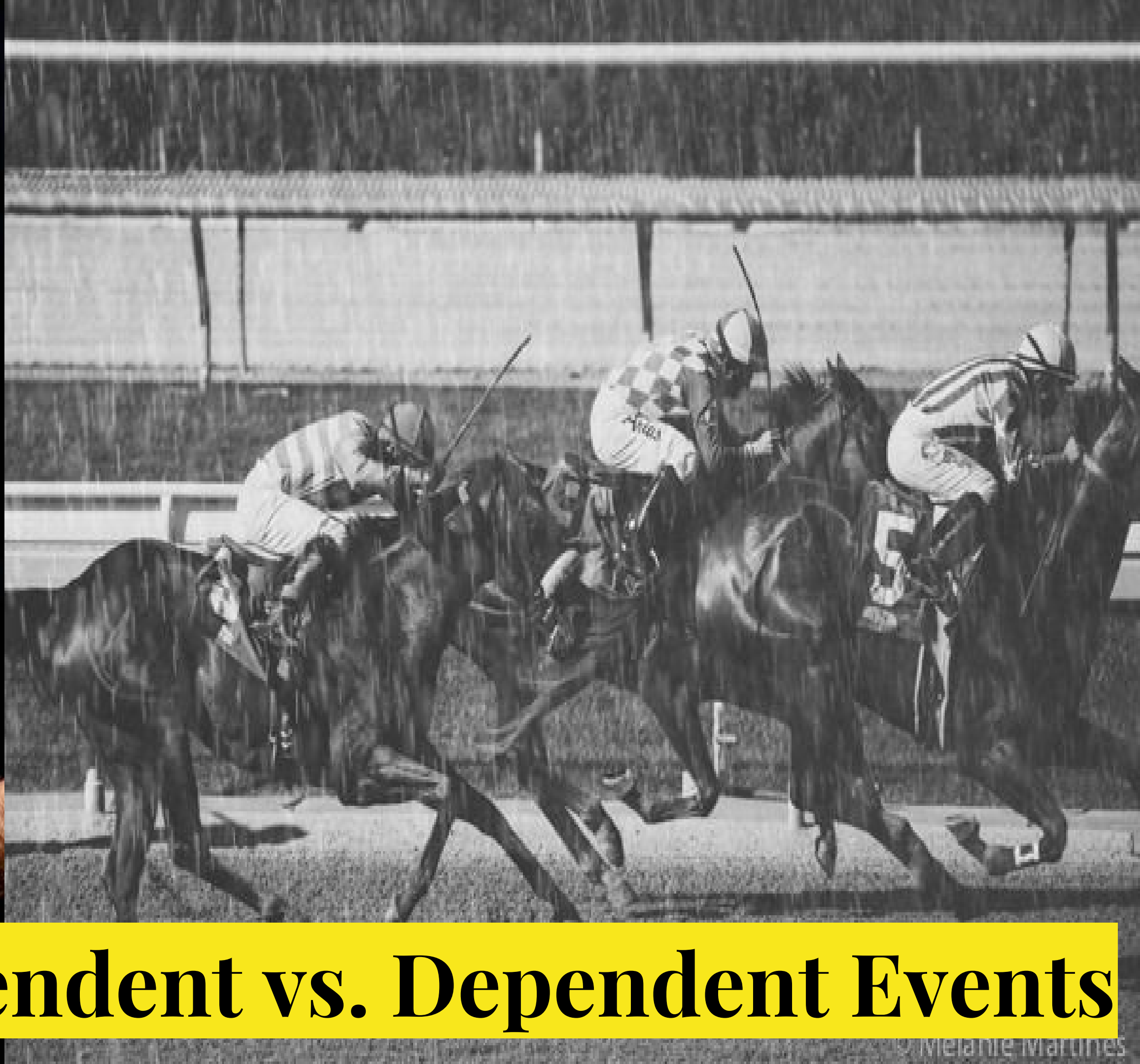


Naive Bayes is a type of probablilistic classifier.

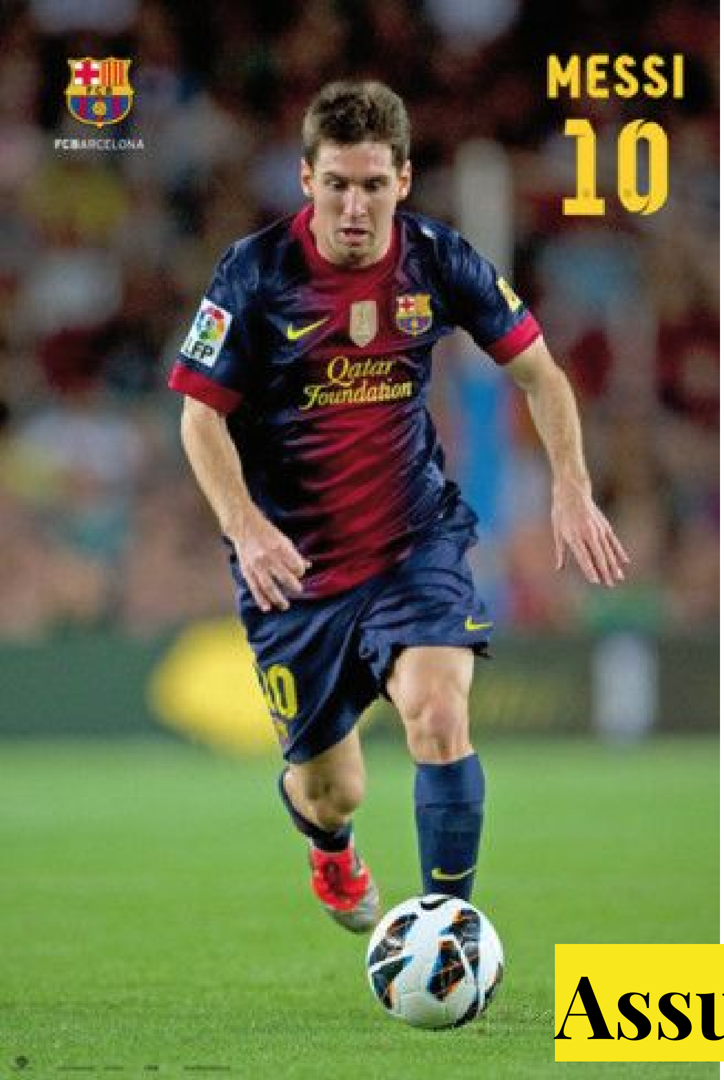
Naive Bayes in stats theory

The math for Naive Bayes is based on Bayes theorem. It states that the likelihood of one event is independent of the likelihood of another event.

Naive Bayes classifiers make use of this “naive” assumption.



Independent vs. Dependent Events



BUY NOW

Assumption: Independent Events

Naive Bayes in Spam Classifiers

Q: What is the probability of an email being Spam and Ham?

$$P(c|x) = P(x|c)P(c) \;/\; P(x)$$

likelihood of predictor in the class
e.g. 28 out of 50 spam emails have the word "free"

prior probability of class
e.g. 50 of all 150 emails are spam

prior probability of predictor
e.g. 72 of 150 emails have word free

Picks category with MAP

MAP: maximum a posterori probability

label = argmax P(x|c)P(c)

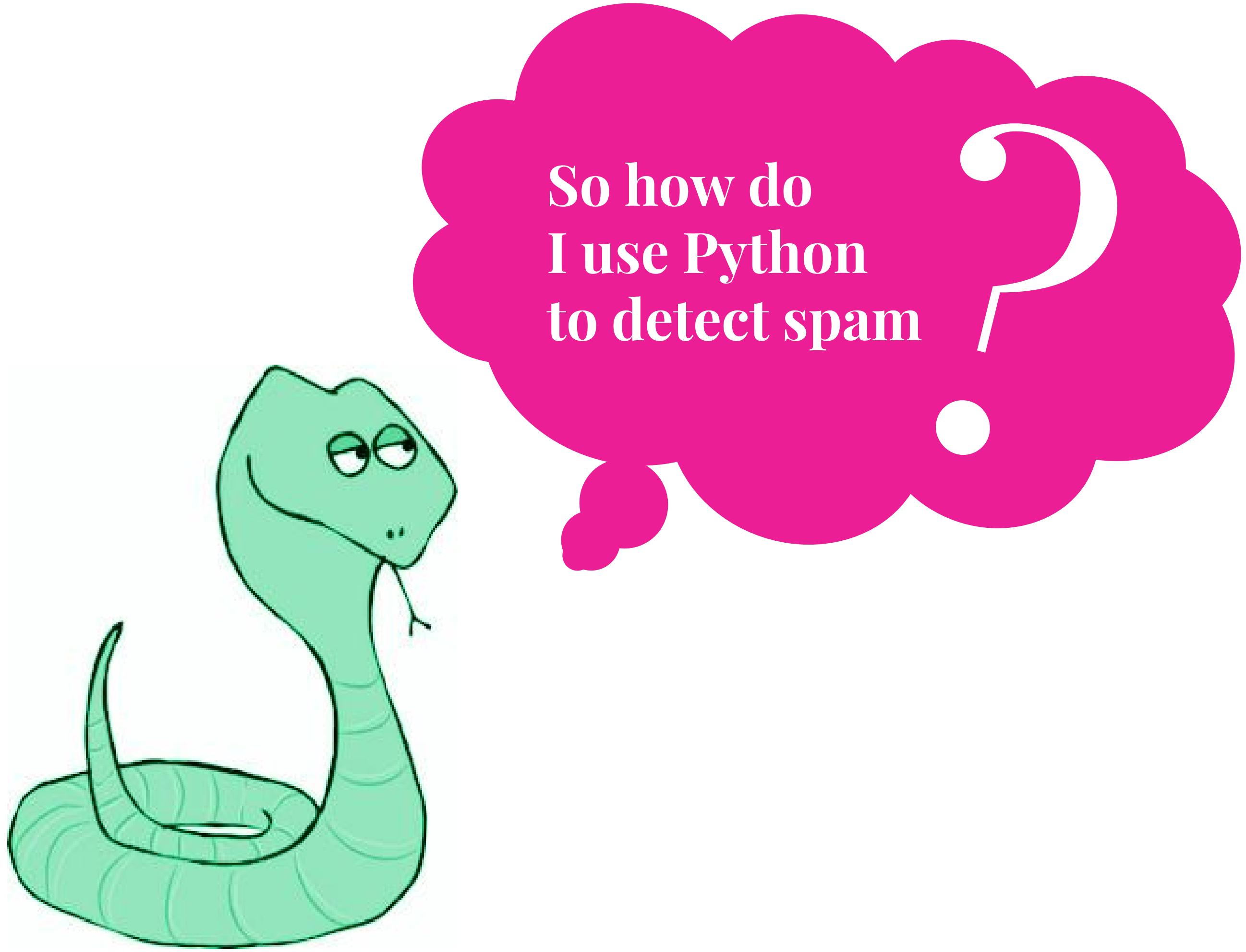
P(x) identical for all classes; don't use it

Q: Is P(c|x) bigger for ham or spam?

A: Pick the MAP!

Why Naive Bayes?

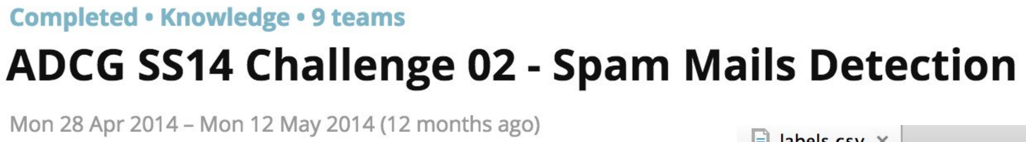
There are other classifier algorithms you could explore but the math behind Naive Bayes is much simpler and suites what we need to do just fine.


So how do I use Python to detect spam ?

Task: Spam Detection

kaggle in Class

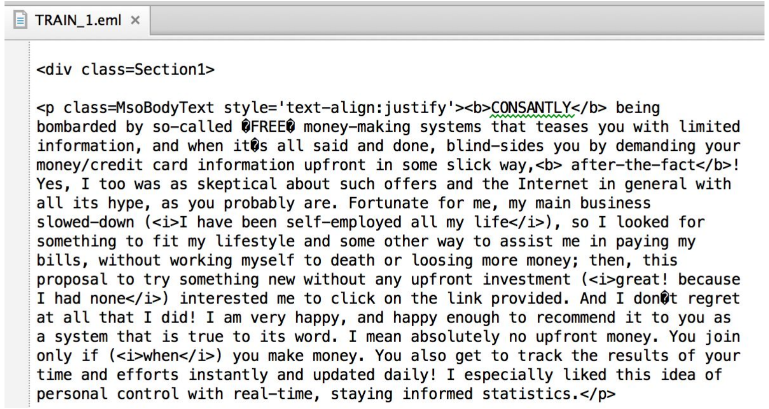
Training data contains 2500 mails both in Ham (1721) labelled as 1 and Spam(779) labelled as 0.



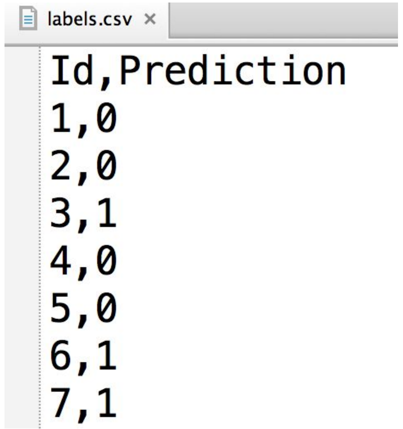
Completed • Knowledge • 9 teams

ADCG SS14 Challenge 02 - Spam Mails Detection

Mon 28 Apr 2014 – Mon 12 May 2014 (12 months ago)



```
<div class=Section1>

<p class=MsoBodyText style='text-align:justify'><b>CONSANTLY</b> being
bombarded by so-called �FREE� money-making systems that teases you with limited
information, and when it�s all said and done, blind-sides you by demanding your
money/credit card information upfront in some slick way,<b> after-the-fact</b>!
Yes, I too was as skeptical about such offers and the Internet in general with
all its hype, as you probably are. Fortunate for me, my main business
slowed-down (<i>I have been self-employed all my life</i>), so I looked for
something to fit my lifestyle and some other way to assist me in paying my
bills, without working myself to death or loosing more money; then, this
proposal to try something new without any upfront investment (<i>great! because
I had none</i>) interested me to click on the link provided. And I don�t regret
at all that I did! I am very happy, and happy enough to recommend it to you as
a system that is true to its word. I mean absolutely no upfront money. You join
only if (<i>when</i>) you make money. You also get to track the results of your
time and efforts instantly and updated daily! I especially liked this idea of
personal control with real-time, staying informed statistics.</p>
```

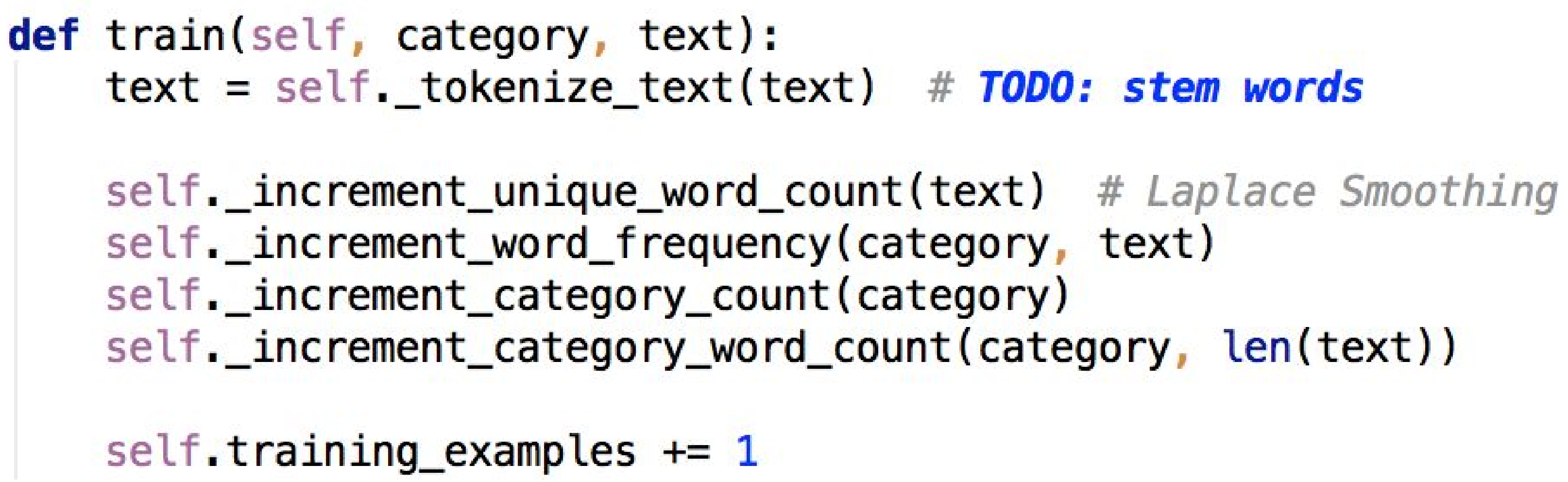


```
Id,Prediction
1,0
2,0
3,1
4,0
5,0
6,1
7,1
```

Tools: What we'll use.

email	email package to parse emails into Message objects
lxml	to transform email messages into plain text
nltk	filter out "stop" words

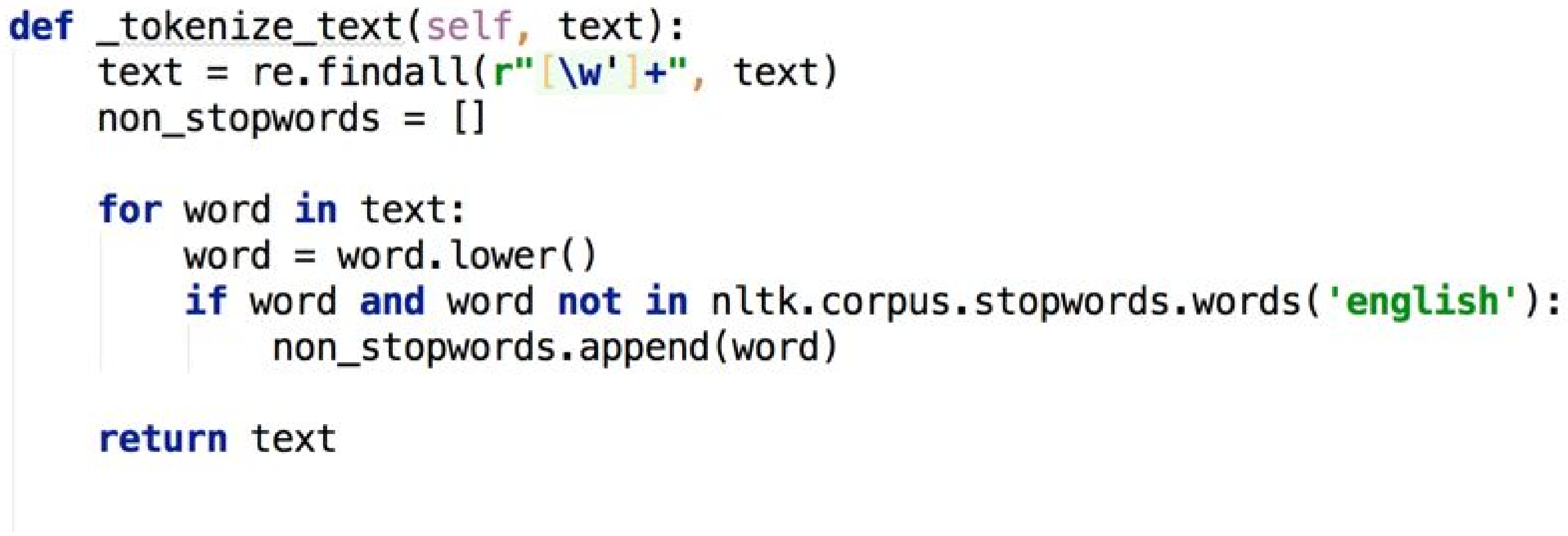
Task: Training the spam filter



```
def train(self, category, text):
    text = self._tokenize_text(text)  # TODO: stem words

    self._increment_unique_word_count(text)  # Laplace Smoothing
    self._increment_word_frequency(category, text)
    self._increment_category_count(category)
    self._increment_category_word_count(category, len(text))

    self.training_examples += 1
```

Stemming words - treat words like "shop" and "shopping" alike.

Training the Python Naive Bayes classifier



```
def _tokenize_text(self, text):
    text = re.findall(r"[\w']+", text)
    non_stopwords = []

    for word in text:
        word = word.lower()
        if word and word not in nltk.corpus.stopwords.words('english'):
            non_stopwords.append(word)

    return text
```

Tokenize text into a bag of words

Zero-Word Frequency

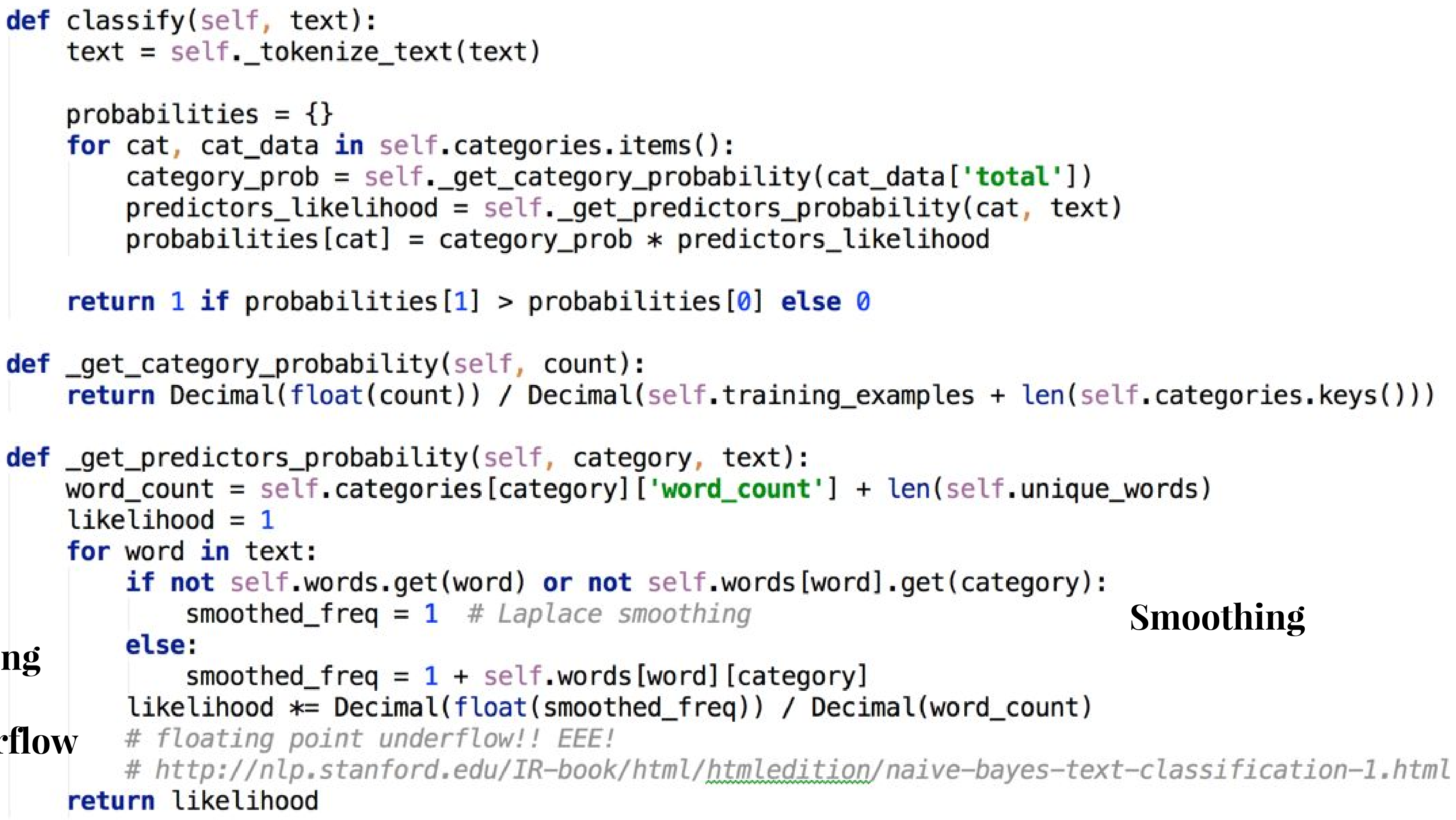
What happens if have a new word in an email that was not yet seen by training data?

$$P(free|spam) * P(your|spam) * .... * P(junk|spam)$$

$$0/150 * \quad 50/150 * \quad .... * \quad 25 / 150$$

Laplace smoothing allows you to add a small positive (e.g. 1) to all counts to prevent this.

Task: Classifying emails

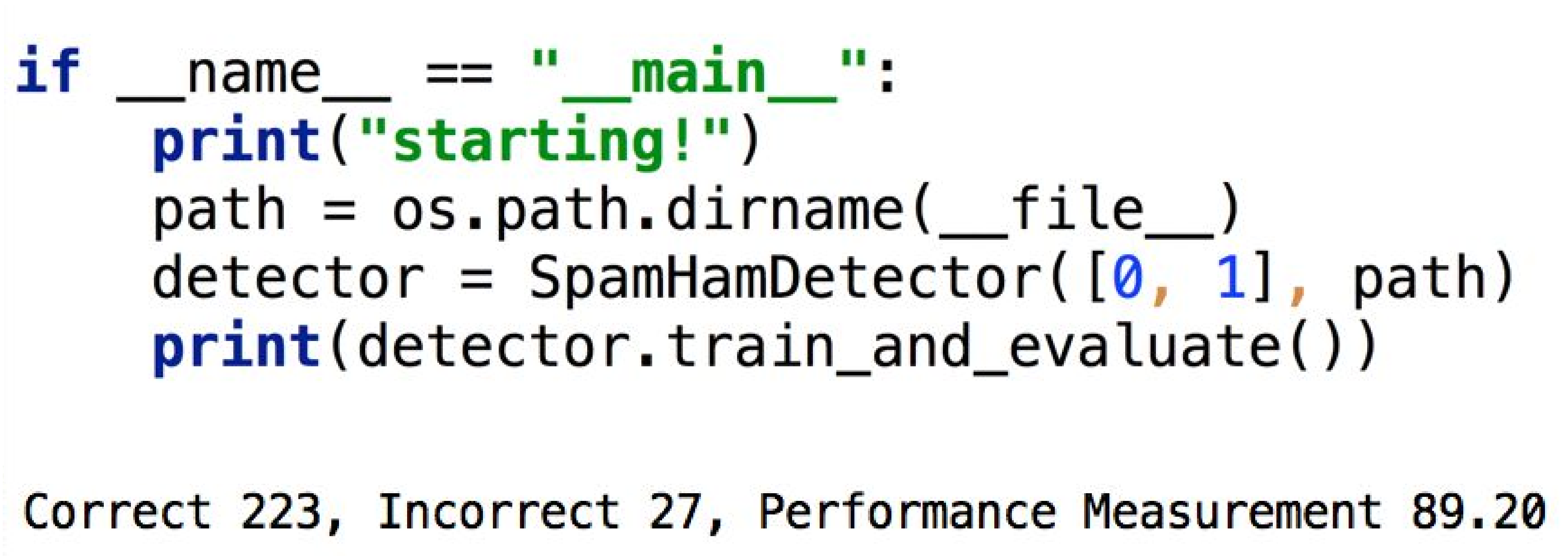


```
def classify(self, text):
    text = self._tokenize_text(text)

    probabilities = {}
    for cat, cat_data in self.categories.items():
        category_prob = self._get_category_probability(cat_data['total'])
        predictors_likelihood = self._get_predictors_probability(cat, text)
        probabilities[cat] = category_prob * predictors_likelihood

    return 1 if probabilities[1] > probabilities[0] else 0

def _get_category_probability(self, count):
    return Decimal(float(count)) / Decimal(self.training_examples + len(self.categories.keys()))

def _get_predictors_probability(self, category, text):
    word_count = self.categories[category]['word_count'] + len(self.unique_words)
    likelihood = 1
    for word in text:
        if not self.words.get(word) or not self.words[word].get(category):
            smoothed_freq = 1  # Laplace smoothing
        else:
            smoothed_freq = 1 + self.words[word][category]
        likelihood *= Decimal(float(smoothed_freq)) / Decimal(word_count)
        # floating point underflow!! EEE!
        # http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html
    return likelihood
```

Smoothing

Floating Point Underflow

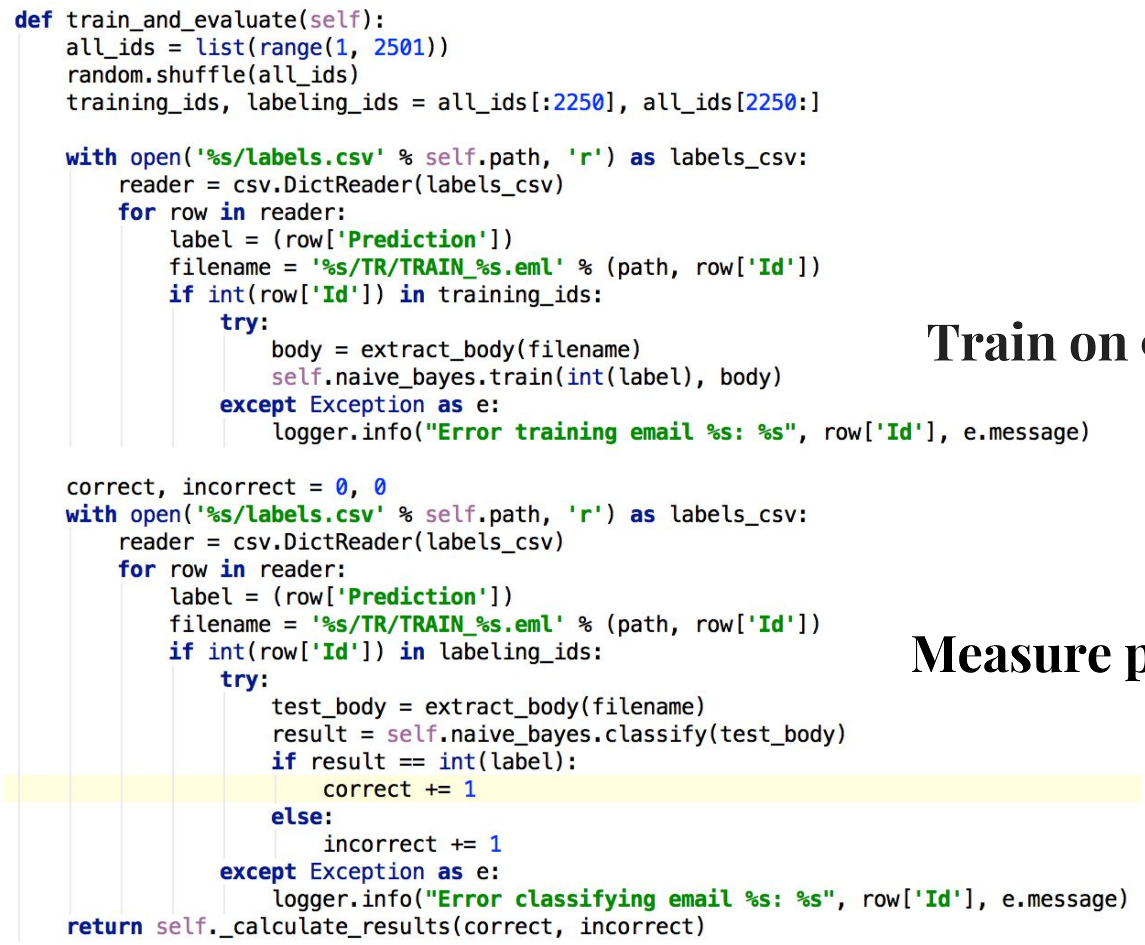
Performance Measurement: 90/10 Split



```python
if __name__ == "__main__":
    print("starting!")
    path = os.path.dirname(__file__)
    detector = SpamHamDetector([0, 1], path)
    print(detector.train_and_evaluate())
```

```
Correct 223, Incorrect 27, Performance Measurement 89.20
```

Classify the unseen examples.



```python
def train_and_evaluate(self):
    all_ids = list(range(1, 2501))
    random.shuffle(all_ids)
    training_ids, labeling_ids = all_ids[:2250], all_ids[2250:]

    with open('%s/labels.csv' % self.path, 'r') as labels_csv:
        reader = csv.DictReader(labels_csv)
        for row in reader:
            label = (row['Prediction'])
            filename = '%s/TR/TRAIN_%s.eml' % (path, row['Id'])
            if int(row['Id']) in training_ids:
                try:
                    body = extract_body(filename)
                    self.naive_bayes.train(int(label), body)
                except Exception as e:
                    logger.info("Error training email %s: %s", row['Id'], e.message)

    correct, incorrect = 0, 0
    with open('%s/labels.csv' % self.path, 'r') as labels_csv:
        reader = csv.DictReader(labels_csv)
        for row in reader:
            label = (row['Prediction'])
            filename = '%s/TR/TRAIN_%s.eml' % (path, row['Id'])
            if int(row['Id']) in labeling_ids:
                try:
                    test_body = extract_body(filename)
                    result = self.naive_bayes.classify(test_body)
                    if result == int(label):
                        correct += 1
                    else:
                        incorrect += 1
                except Exception as e:
                    logger.info("Error classifying email %s: %s", row['Id'], e.message)
    return self._calculate_results(correct, incorrect)
```

Train on 90% of training data

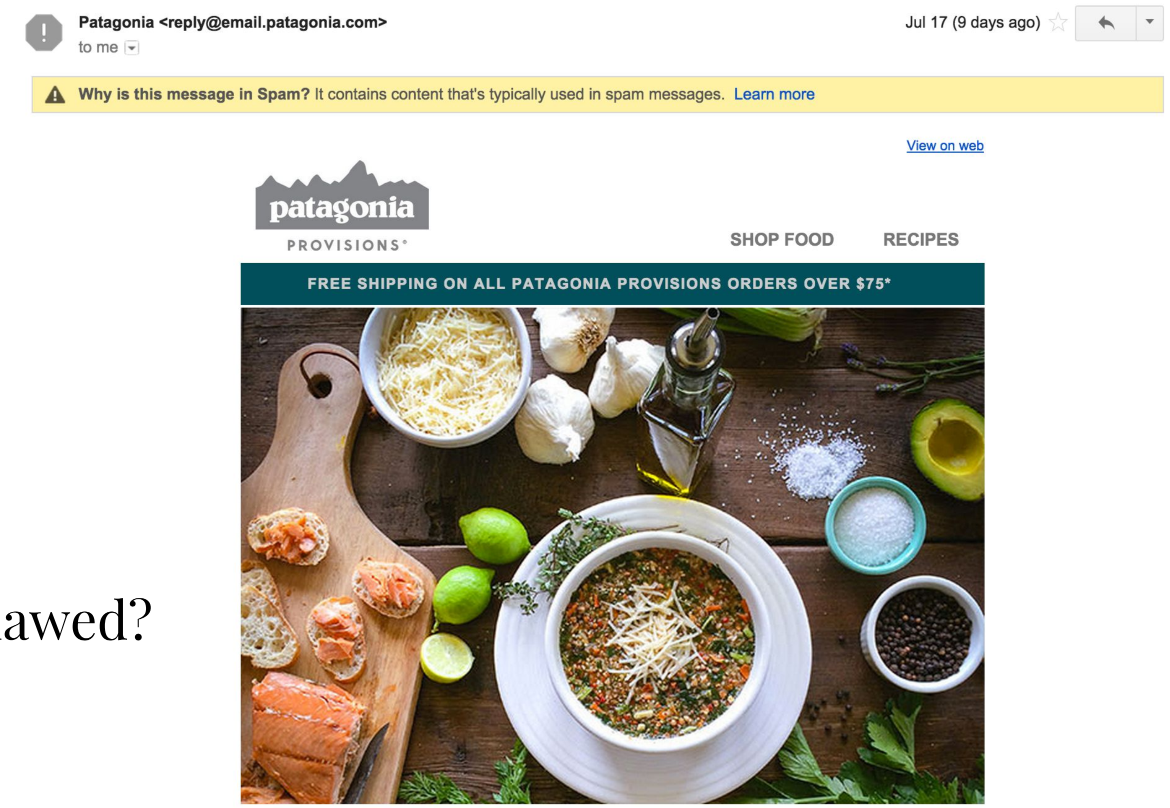
Measure performance on 10% of data

False Positives

I signed up to receive promotional deals from Patagonia.

"Typically used in spam" implementation may be flawed? (e.g. too naive?).

Google spam ⤜ report as spam (or not!)


Patagonia <reply@email.patagonia.com>
to me
Jul 17 (9 days ago)

Why is this message in Spam? It contains content that's typically used in spam messages. Learn more

View on web

patagonia PROVISIONS

SHOP FOOD RECIPES

FREE SHIPPING ON ALL PATAGONIA PROVISIONS ORDERS OVER $75*

Naive Bayes limitations & challenges

- Independence assumption is a simplistic model of the world
- Overestimates the probability of the label ultimately selected
- Inconsistent labeling of data (e.g. same email has both spam label and ham label)

Improve Performance

More & better feature extraction

Other possible features:

- Subject
- Images
- Sender

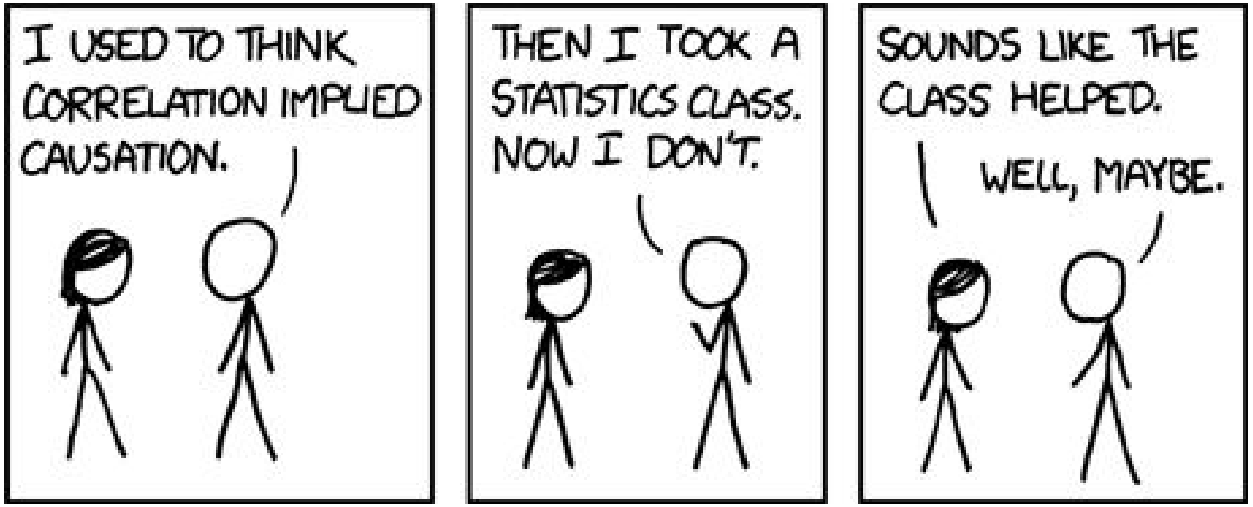
<u>**MORE DATA**</u>**!**

Want to learn more?

Kaggle for toy machine learning problems!

Introduction to Machine Learning With Python by Sarah Guido

Your local Python user group!


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Thank you!

bit.ly/europython2016-lmesa | @looooorenanicole