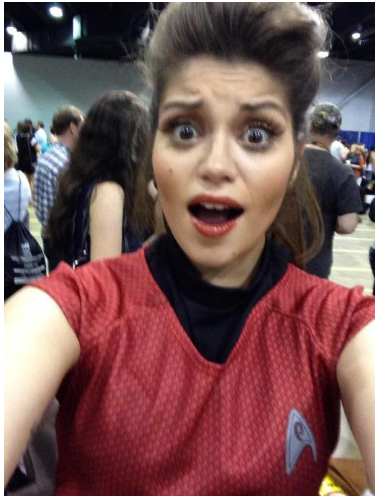
Predicting free pizza with Python. Cowabunga dude!

Lorena Mesa @loooorenanicole

GOTO Conference - Chicago 2017

http://bit.ly/2qoU7Pp

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>

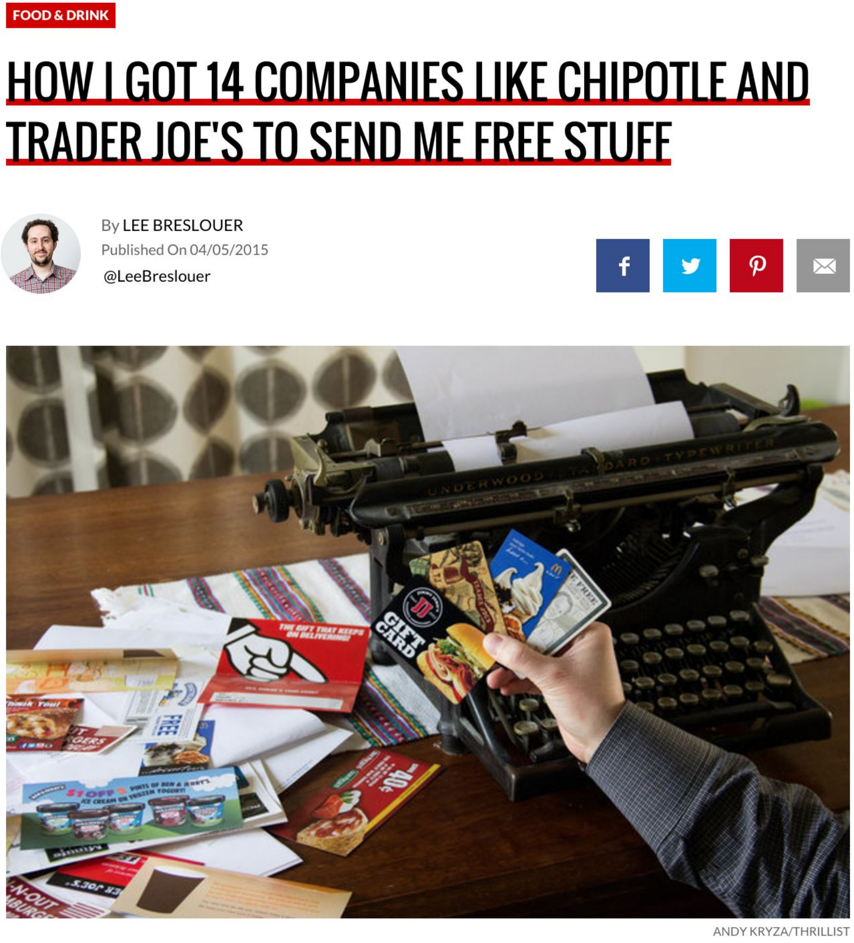
django

pyladies CHICAGO

python SOFTWARE FOUNDATION

sproutsocial

http://bit.ly/2qoU7Pp



FOOD & DRINK

HOW I GOT 14 COMPANIES LIKE CHIPOTLE AND TRADER JOE'S TO SEND ME FREE STUFF

By LEE BRESLOUER
Published On 04/05/2015
@LeeBreslouer

ANDY KRYZA/THRILLIST

Flattery Project

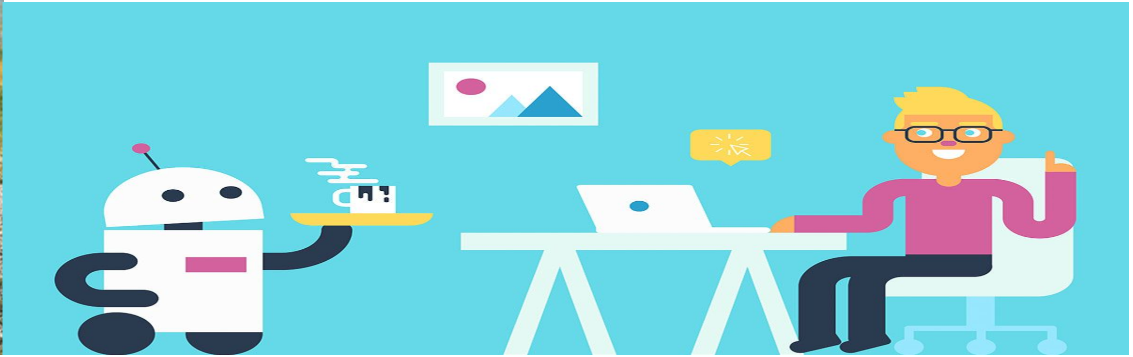
I thanked 42 companies, see what they sent me!

wallethacks.com

http://bit.ly/2qoU7Pp

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?
4. Example of Python in action – classifying if a request will garner free pizza!



http://bit.ly/2qoU7Pp

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**.

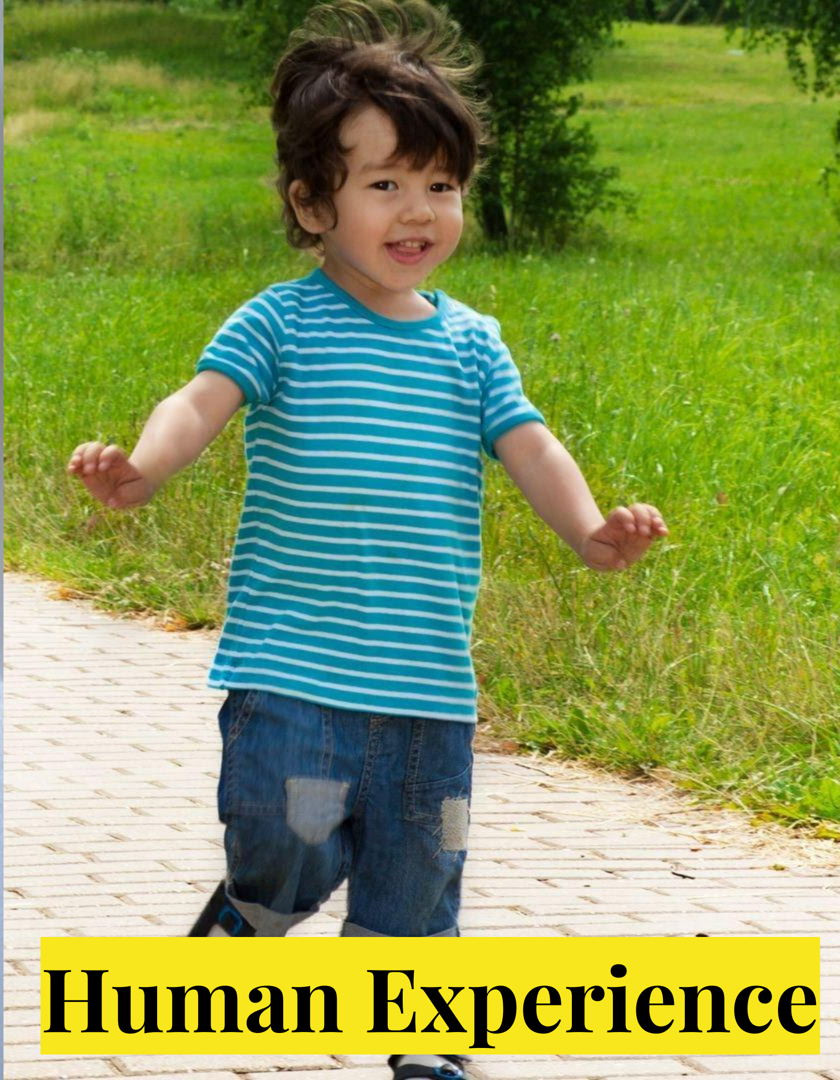
http://bit.ly/2qoU7Pp

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)

http://bit.ly/2qoU7Pp



Human Experience

Recorded Experience

Classification in machine learning

Task: Classify a piece of data

Is a pizza request successful? Is it altruistic or not?

http://bit.ly/2qoU7Pp

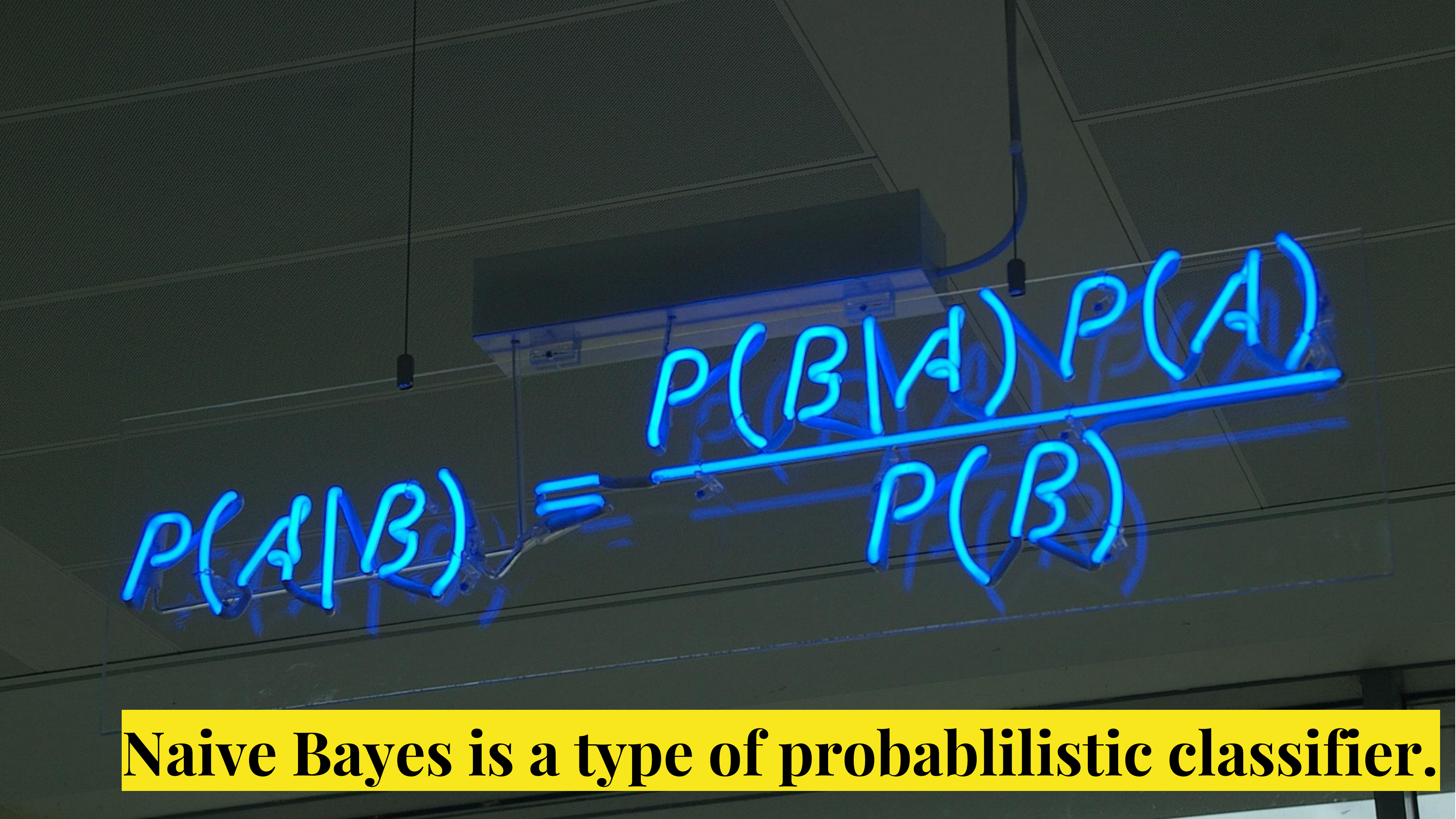
Experience: Labeled training data

Request_ id	*No*
Request_ id	*Yes*

Performance Measurement: Is the label correct?

Verify if the request is successful or not

http://bit.ly/2qoU7Pp



Naive Bayes is a type of probablilistic classifier.

First, why Naive Bayes?

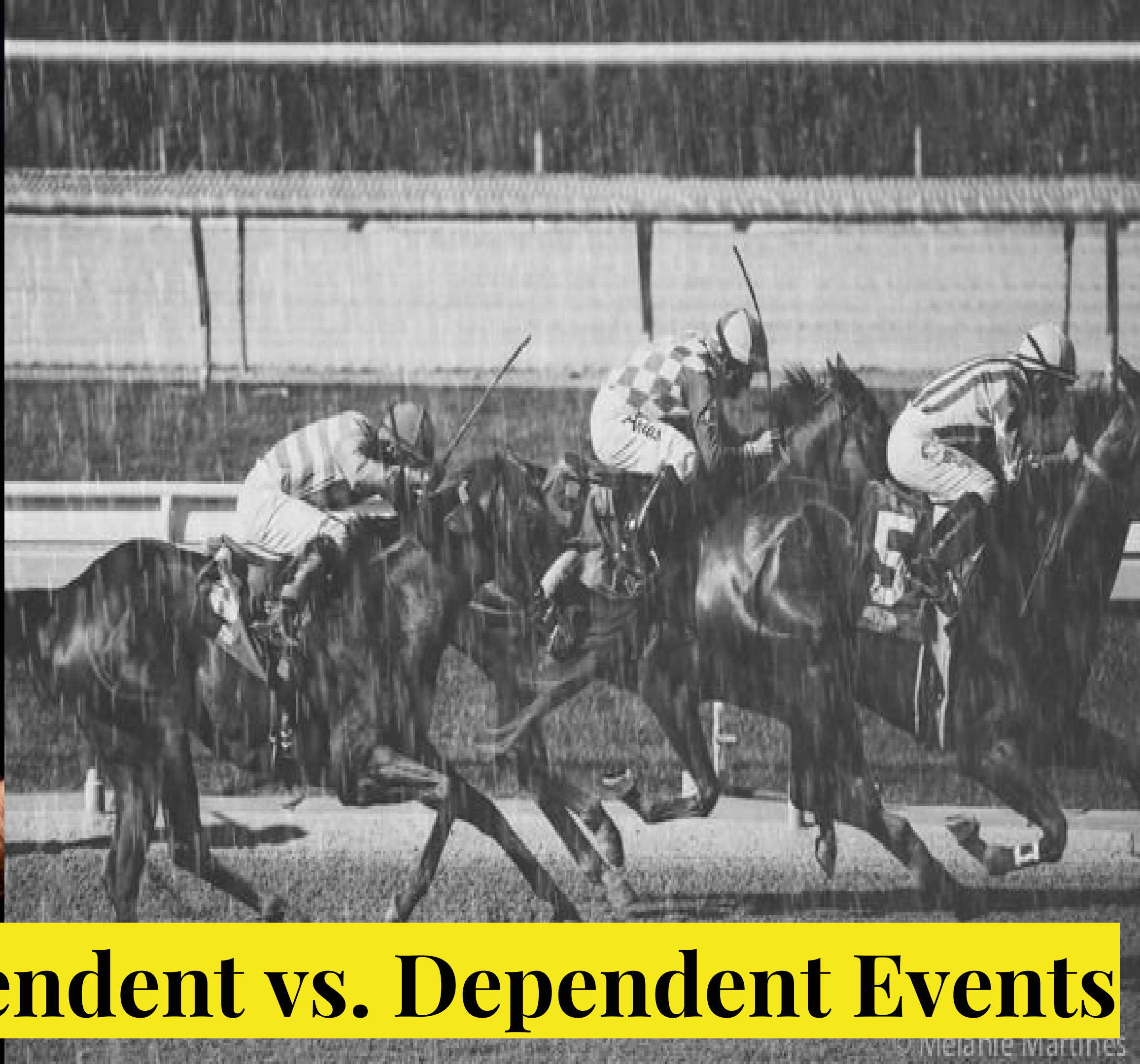
1. Requires a small amount of training data to start making predictions!
2. Useful if only need to know what is most likely, not the actual percentage of likelihood
3. Can work with missing data!

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.

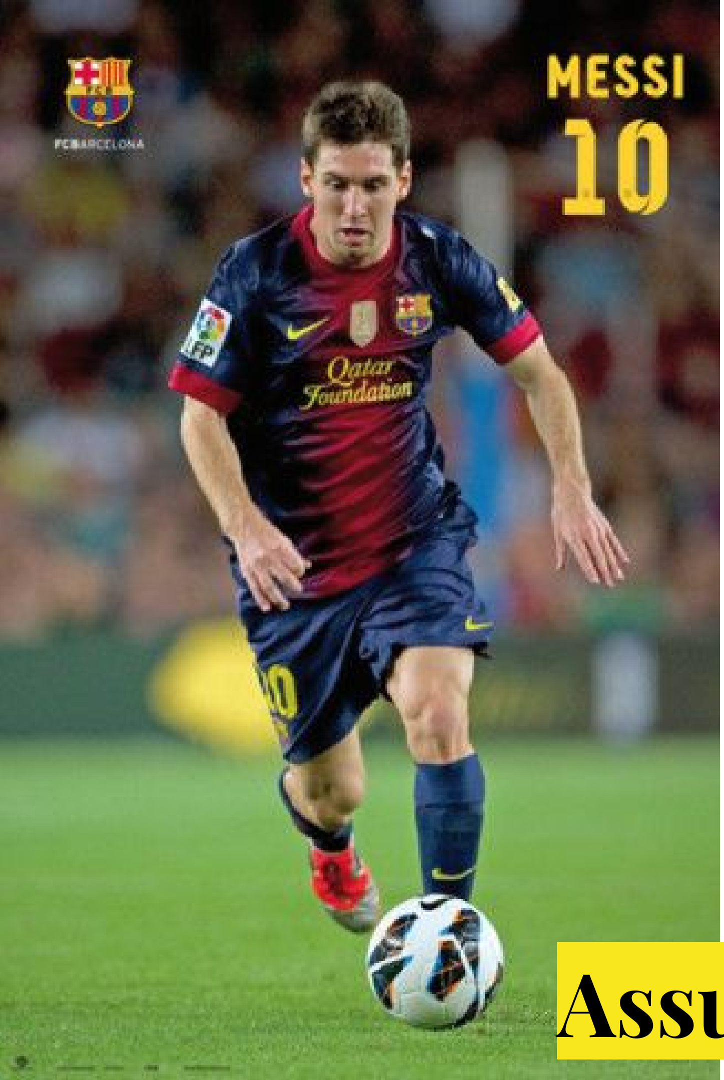
Naive Bayes classifiers make use of this “naive” assumption.

http://bit.ly/2qoU7Pp



© Melanie Martines

Independent vs. Dependent Events



Assumption: Independent Events

Naive Bayes in Classifying Altruism

Q: What is the probability of an pizza request being successful or not?

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

likelihood of predictor in the class
e.g. 28 out of 50 requests have the word “hungry”

prior probability of class
e.g. 50 of all 150 requests are are unsuccessful

prior probability of predictor
e.g. 72 of 150 requests have “hungry”

http://bit.ly/2qoU7Pp

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 class one (success) or two (not)?

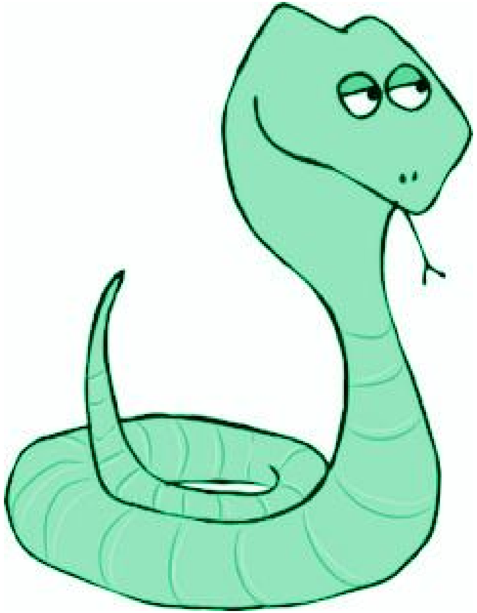
A: Pick the MAP!

http://bit.ly/2qoU7Pp

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 ?



http://bit.ly/2qoU7Pp

Task: Altruism Classification

Random Acts of Pizza

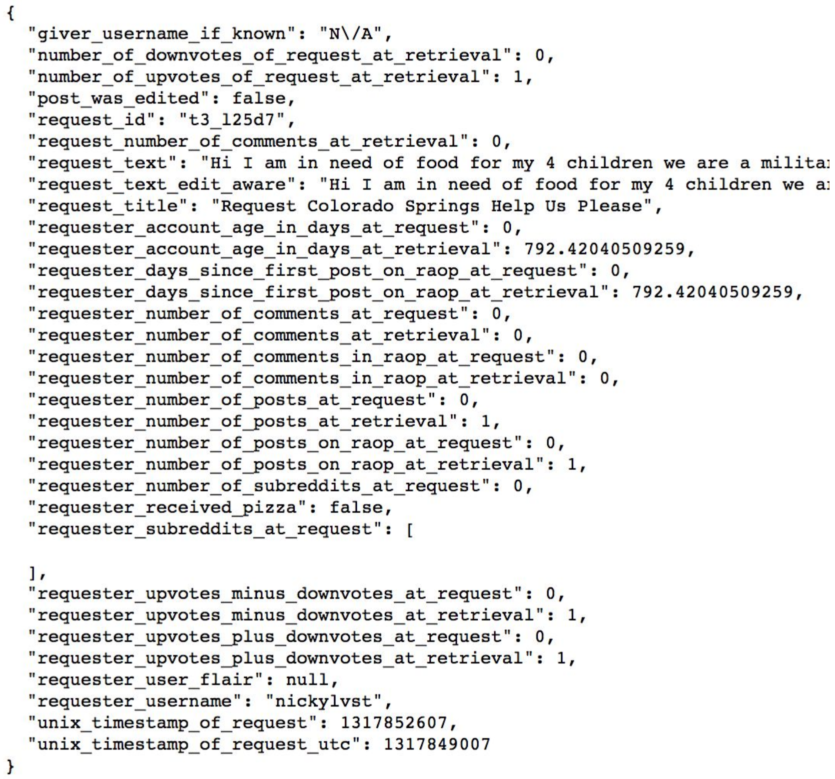
Predicting altruism through free pizza

464 teams · 2 years ago

Training data contains:

- 5671 requests
- Successful (994) labelled as True
- Unsuccessful (3046) labelled as False.

Unlabeled data has 1631 requests.



```
{
  "giver_username_if_known": "N\/A",
  "number_of_downvotes_of_request_at_retrieval": 0,
  "number_of_upvotes_of_request_at_retrieval": 1,
  "post_was_edited": false,
  "request_id": "t3_l25d7",
  "request_number_of_comments_at_retrieval": 0,
  "request_text": "Hi I am in need of food for my 4 children we are a milita
  "request_text_edit_aware": "Hi I am in need of food for my 4 children we a
  "request_title": "Request Colorado Springs Help Us Please",
  "requester_account_age_in_days_at_request": 0,
  "requester_account_age_in_days_at_retrieval": 792.42040509259,
  "requester_days_since_first_post_on_raop_at_request": 0,
  "requester_days_since_first_post_on_raop_at_retrieval": 792.42040509259,
  "requester_number_of_comments_at_request": 0,
  "requester_number_of_comments_at_retrieval": 0,
  "requester_number_of_comments_in_raop_at_request": 0,
  "requester_number_of_comments_in_raop_at_retrieval": 0,
  "requester_number_of_posts_at_request": 0,
  "requester_number_of_posts_at_retrieval": 1,
  "requester_number_of_posts_on_raop_at_request": 0,
  "requester_number_of_posts_on_raop_at_retrieval": 1,
  "requester_number_of_subreddits_at_request": 0,
  "requester_received_pizza": false,
  "requester_subreddits_at_request": [

  ],
  "requester_upvotes_minus_downvotes_at_request": 0,
  "requester_upvotes_minus_downvotes_at_retrieval": 1,
  "requester_upvotes_plus_downvotes_at_request": 0,
  "requester_upvotes_plus_downvotes_at_retrieval": 1,
  "requester_user_flair": null,
  "requester_username": "nickylvst",
  "unix_timestamp_of_request": 1317852607,
  "unix_timestamp_of_request_utc": 1317849007
}
```

http://bit.ly/2dwYIbp

Example Full Text of Requests

[REQUEST] Florida Haven't worked in a couple weeks and won't have money for another 2 or 3 weeks. Really looking to have some pizza to share with my family

[Request] Would love a pizza tonight Been a lurker for some time, figured I'd give it a shot. Nothing special about me. Just moved to San Francisco and don't know many people, so I figured I'd just stay in tonight and hope for some cheesy goodness. :)

[Request] Hungry Hungry Hoosier broke college student with -8.00 dollars to my name and between work checks(subway) would greatly appreciate a pizza to offset a 8th day of Peanut butter and jelly!

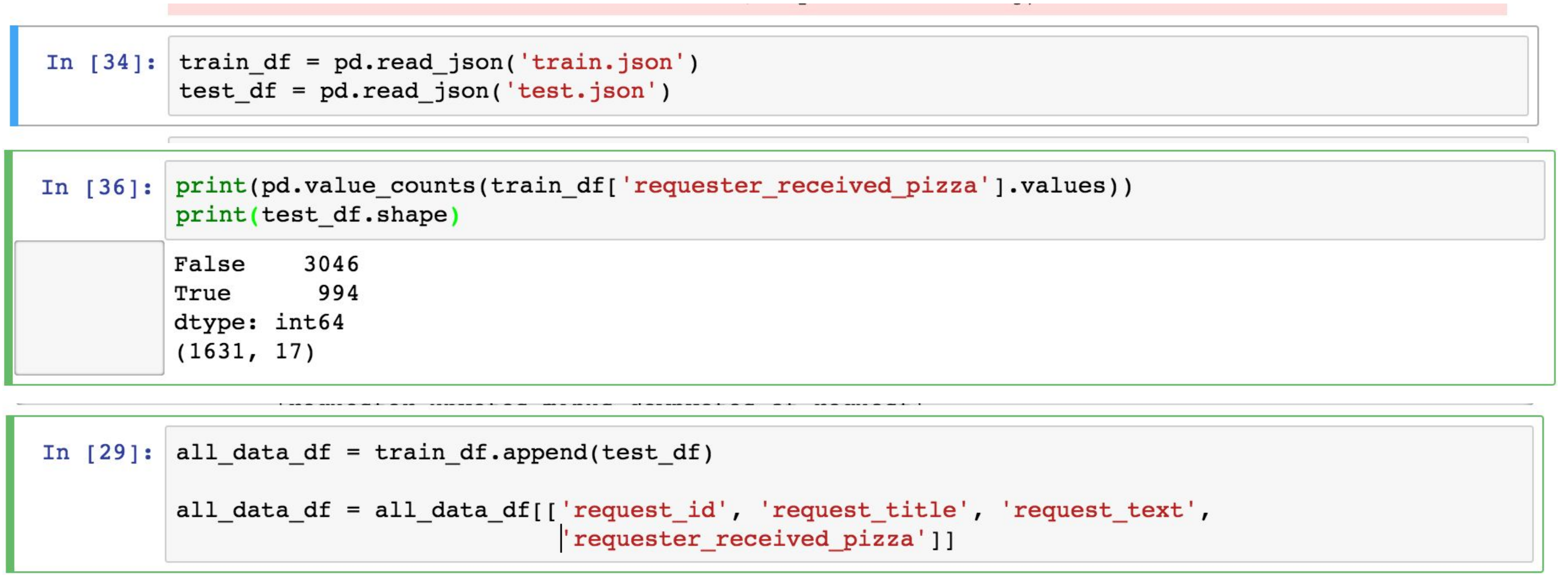
http://bit.ly/2dwYIbp

Tools: What we’ll use.

sklearn	Open source Python machine learning library including classification, SVM, regression algorithms!
pandas	Open source Python data analysis tool with “expressive data structures”.
nltk	Natural language toolkit for Python, use to filter out stop words!
jupyter	Open source web app to create and share code, visualizations, explanatory text.

http://bit.ly/2qoU7Pp

Quickly, Pandas (see notebook!)



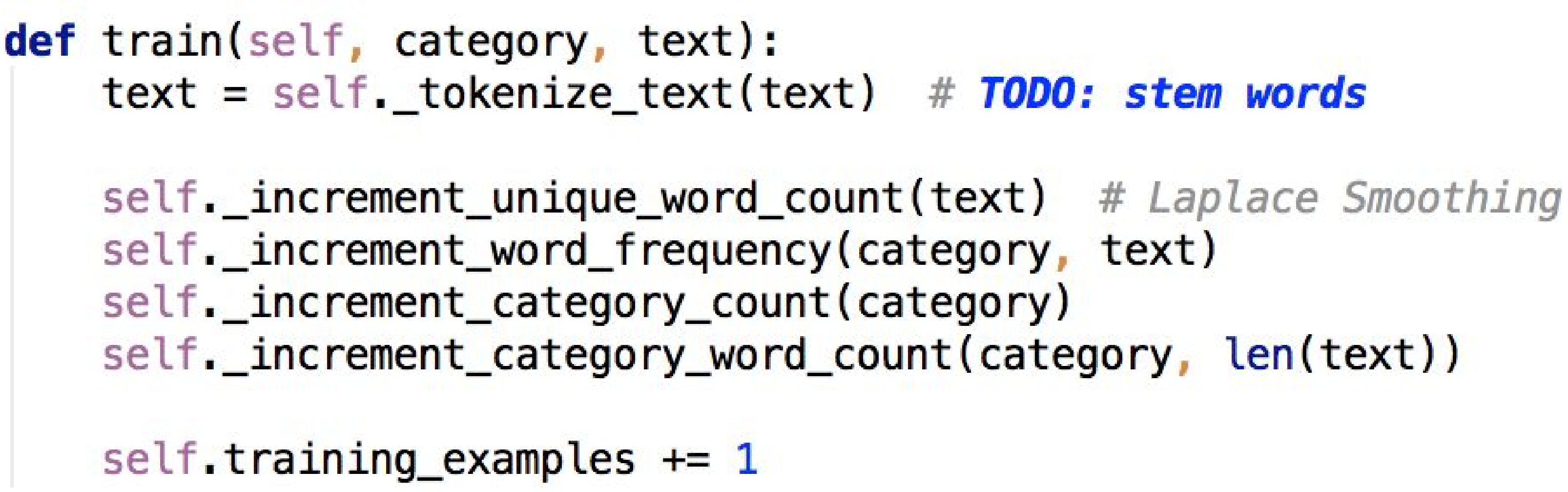
```
In [34]: train_df = pd.read_json('train.json')
         test_df = pd.read_json('test.json')
```

```
In [36]: print(pd.value_counts(train_df['requester_received_pizza'].values))
         print(test_df.shape)

         False    3046
         True      994
         dtype: int64
         (1631, 17)
```

```
In [29]: all_data_df = train_df.append(test_df)

         all_data_df = all_data_df[['request_id', 'request_title', 'request_text',
                                    'requester_received_pizza']]
```

http://bit.ly/2qoU7Pp

Task: Training the spam filter

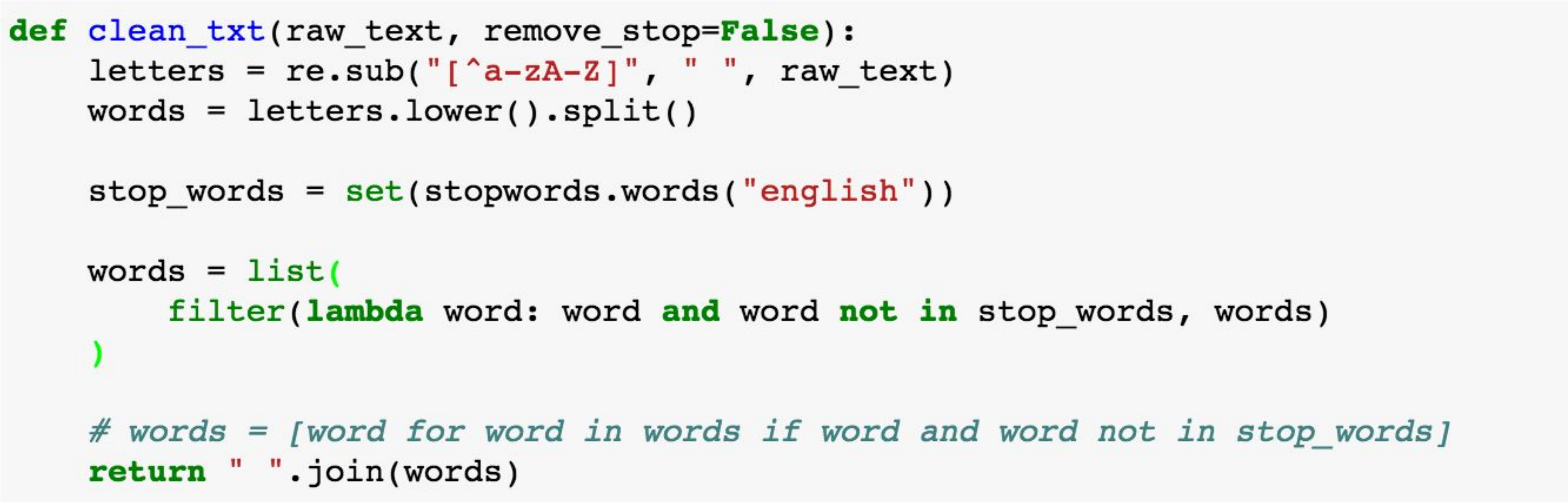


```
def train(self, category, text):
    text = self._tokenize_text(text)  # TODO: stem words

    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

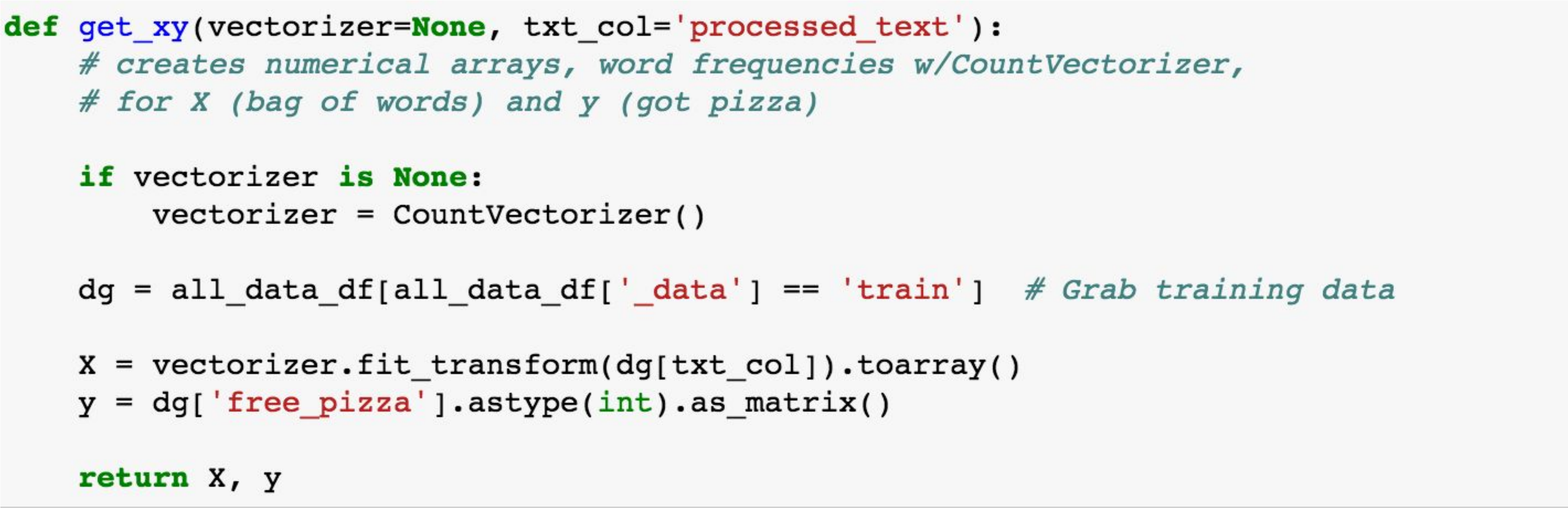
http://bit.ly/2qoU7Pp



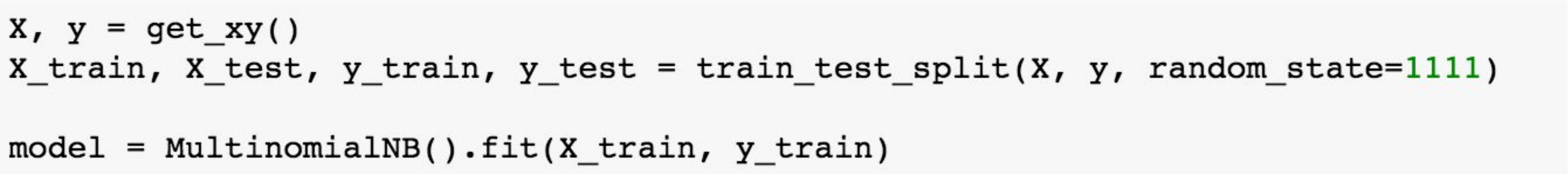
```
def clean_txt(raw_text, remove_stop=False):
    letters = re.sub("[^a-zA-Z]", " ", raw_text)
    words = letters.lower().split()

    stop_words = set(stopwords.words("english"))

    words = list(
        filter(lambda word: word and word not in stop_words, words)
    )

    # words = [word for word in words if word and word not in stop_words]
    return " ".join(words)
```

Tokenize text into a bag of words

http://bit.ly/2qoU7Pp



```python
def get_xy(vectorizer=None, txt_col='processed_text'):
    # creates numerical arrays, word frequencies w/CountVectorizer,
    # for X (bag of words) and y (got pizza)

    if vectorizer is None:
        vectorizer = CountVectorizer()

    dg = all_data_df[all_data_df['_data'] == 'train']  # Grab training data

    X = vectorizer.fit_transform(dg[txt_col]).toarray()
    y = dg['free_pizza'].astype(int).as_matrix()

    return X, y
```

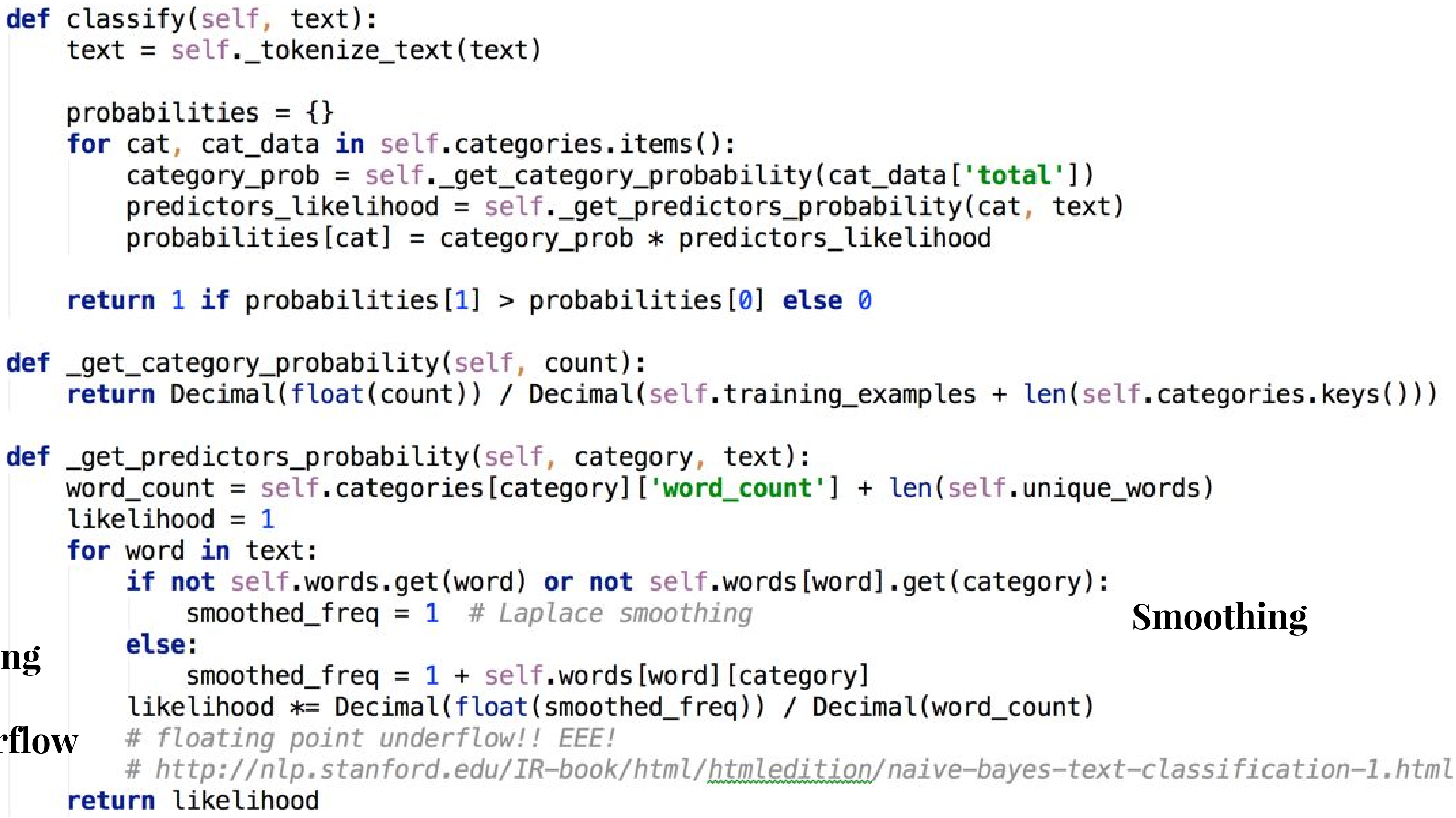


```python
X, y = get_xy()
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1111)

model = MultinomialNB().fit(X_train, y_train)
```

Training the Python Naive Bayes classifier

http://bit.ly/2qoU7Pp

Task: Classifying emails

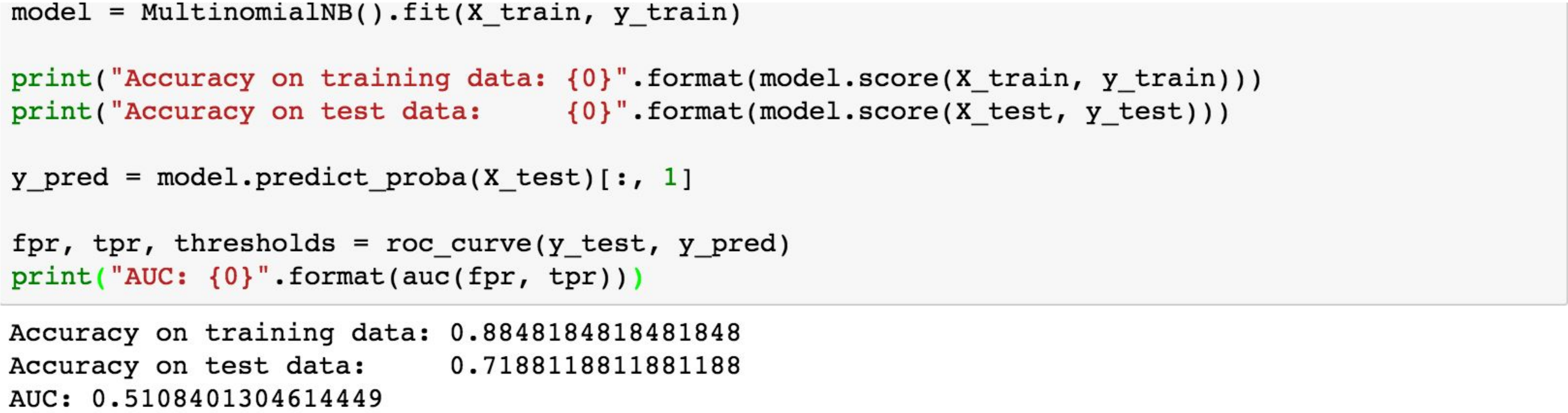


```
def classify(self, text):
    text = self._tokenize_text(text)

    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

http://bit.ly/2qoU7Pp



```
model = MultinomialNB().fit(X_train, y_train)

print("Accuracy on training data: {0}".format(model.score(X_train, y_train)))
print("Accuracy on test data:     {0}".format(model.score(X_test, y_test)))

y_pred = model.predict_proba(X_test)[:, 1]

fpr, tpr, thresholds = roc_curve(y_test, y_pred)
print("AUC: {0}".format(auc(fpr, tpr)))
```

```
Accuracy on training data: 0.8848184818481848
Accuracy on test data:     0.7188118811881188
AUC: 0.5108401304614449
```

Predicting!

http://bit.ly/2qoU7Pp

Performance Measurement Explained

AUC Score

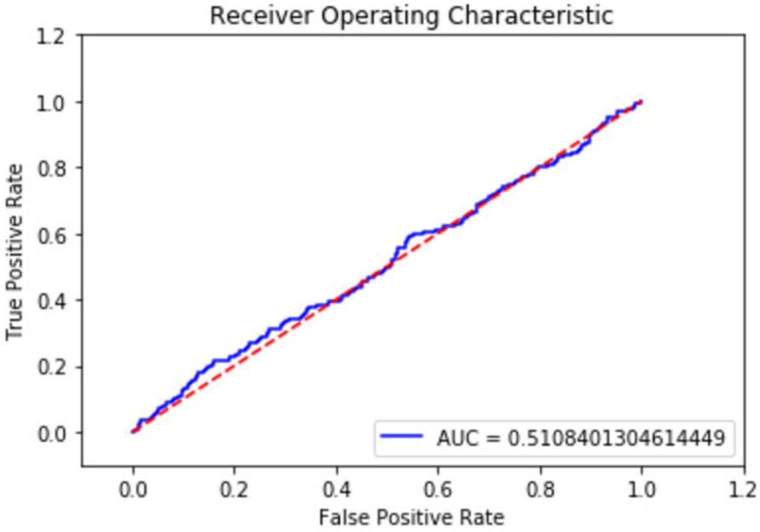
General function that computes the area under the ROC curve; tells us the general accuracy of the classifier over all thresholds

(x) - false positive

(y) - true positive rate

General rule of thumb:

- .90-1 = very good (A)
- .80-.90 = good (B)
- .70-.80 = not so good (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)



http://bit.ly/2qoU7Pp

False Positives

Incorrectly labelled training data:

Requesting Pizza... Cibolo, TX 78108
*Just closed on our new house, no food or money until the 1st. =\\ S**t sucks. Wife doesn't know I'm posting this or she would tell me not to. Cheese or Pepperoni. Message me for address/phone. Thank you reddit and RAoP*

“Too naive”?

http://bit.ly/2qoU7Pp

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:

- Emoji
- Time of day sent
- Information about the requester

MORE DATA!

http://bit.ly/2qoU7Pp

Types of Naive Bayes Alogrithms

- Multinomial: *Can we use samples to represent the frequencies of classes?*
- Bernoulli: *Are the features representable as booleans?*
- Gaussian: *Are we working with continuous data?*

http://bit.ly/2qoU7Pp

Want to learn more?

Kaggle for toy machine learning problems!

Introduction to Machine Learning With Python by Sarah Guido

Your local Python user group!

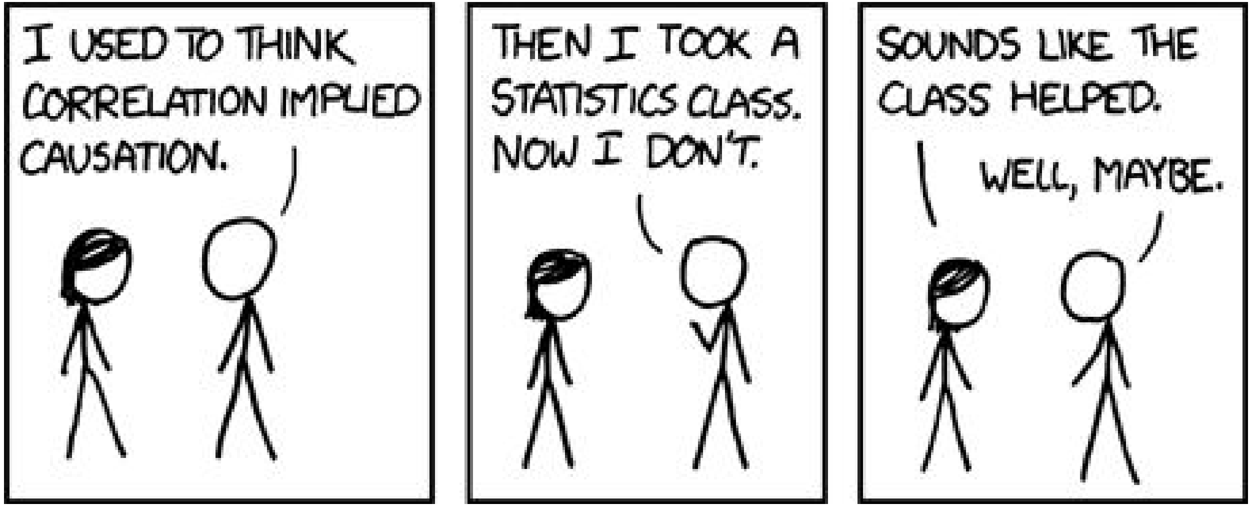
Tim Althoff et al, How to Ask for a Favor: A Case Study on the Success of Altruistic Requests

Harry Zhang's 2005 "The Optimality of Naive Bayes"

Jake Vanderplas PyCon 2016, "Statistics for Hackers"

Brian Lange's PyData Chicago 2016, "It's Not Magic: Explaining Classification Algorithms"

http://bit.ly/2qoU7Pp


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!

http://bit.ly/2qoU7Pp | @looooorenanicole