Predict the Stock Market Using Daily News

CS 360 - Final Project
Tessa Pham

Prompt

Given:

top 25 daily news headlines from Reddit

Predict:

whether DJIA (market index) close value ↓

Data

- Time: **2008-06-08** to **2016-07-01**
- News: headlines from **Reddit WorldNews Channel** (/r/worldnews)
- Stock: **DJIA** from **Yahoo Finance**

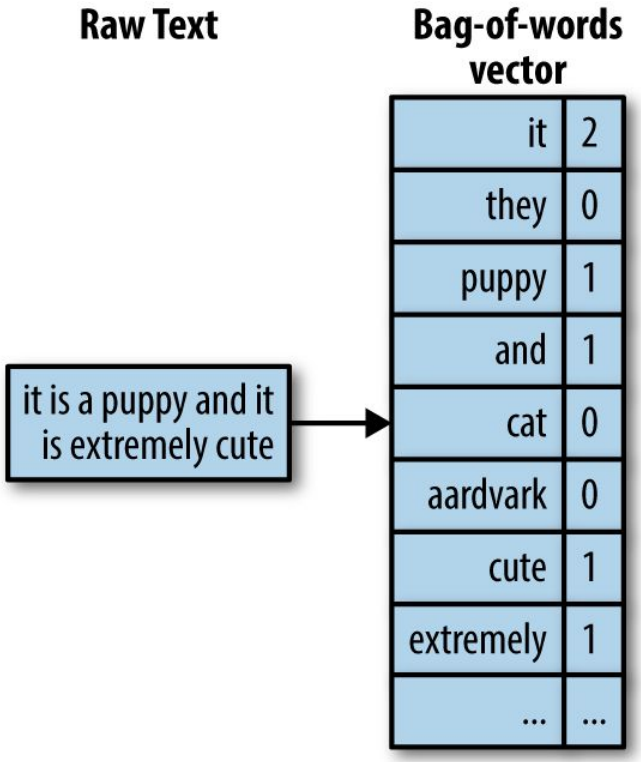
- **n** = 1989
- **p** = ?
- Split: **80/20** (by date)
  - **training:** < 2015-01-01
  - **test:** >= 2015-01-01

Task & Classifiers

- Binary classification: 0 for decrease, 1 for rise or same
- Models: **bag-of-words + TF-IDF**, **bag-of-bigrams**
- Classifiers:
  Logistic Regression, **k-NN**, **Naive Bayes**, **SVM**, **Random Forest**

Bag-of-Words

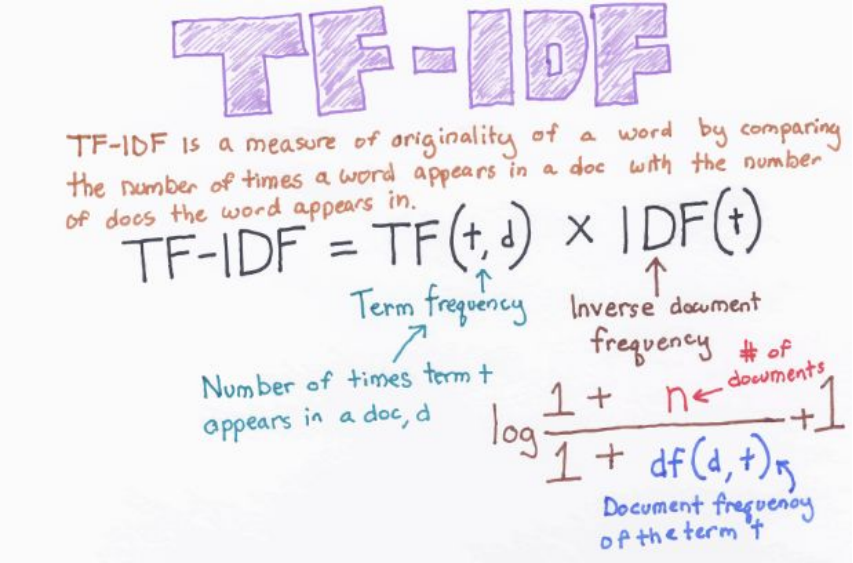
- gather a vocabulary
  (a **set** of words in all documents)
- each word = a feature
- each example/document
  = vector of word counts
- p = 31,675



http://uc-r.github.io/creating-text-features

TF-IDF

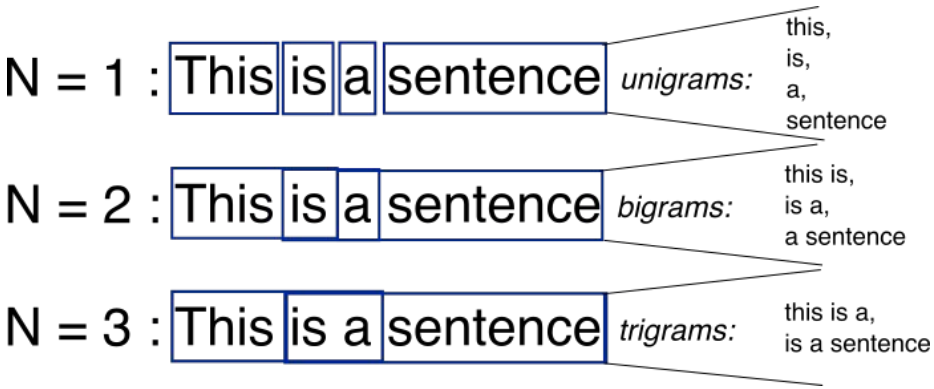
- Term Frequency - Inverse Document Frequency
- TF-IDF = TF x IDF
- assign weights to words based on its importance within a document



https://chrisalbon.com/machine_learning/preprocessing_text/tf-idf/

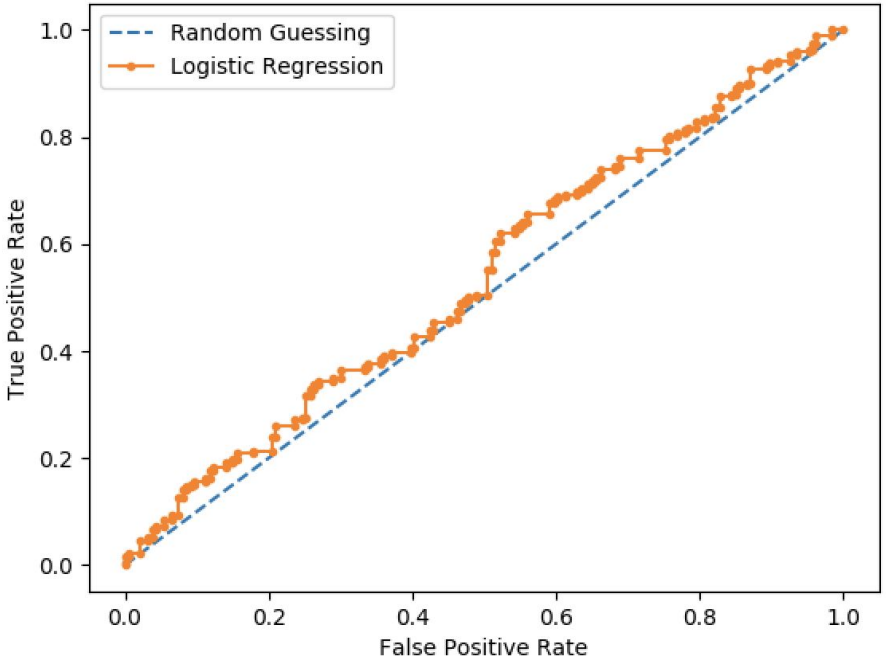
Bag-of-Bigrams

- n-gram = sequence of n consecutive words
- each bigram = a feature
- p = 366,721

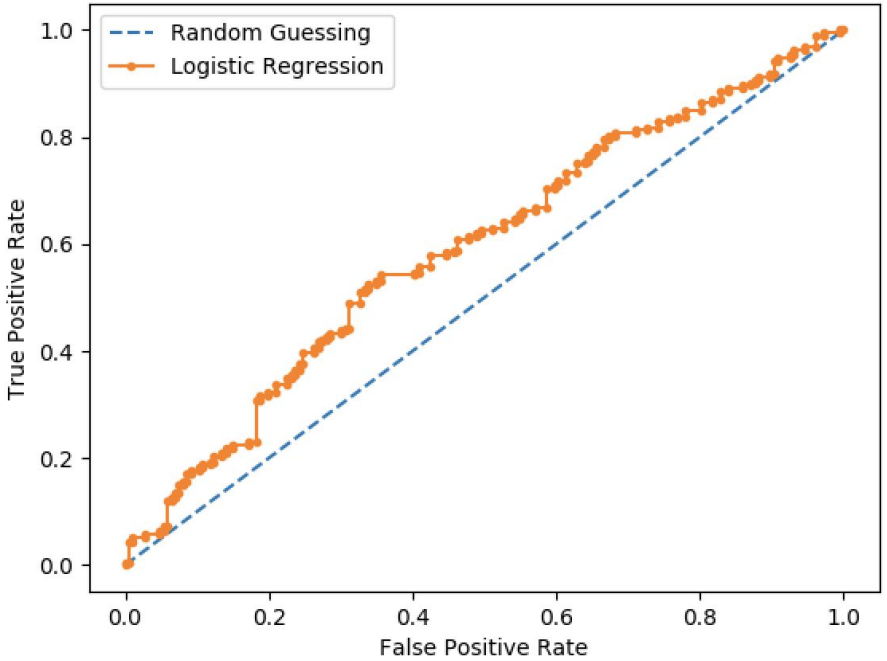


https://stackoverflow.com/questions/18193253/what-exactly-is-an-n-gram

Logistic Regression

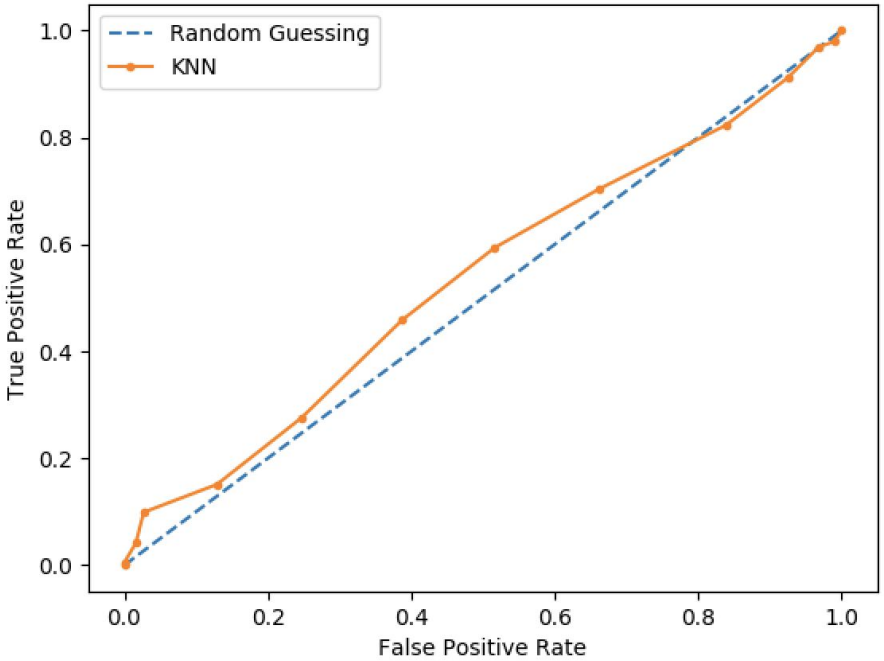


Unigram + TDIF
(Accuracy = **51%**, AUC = **0.468**)

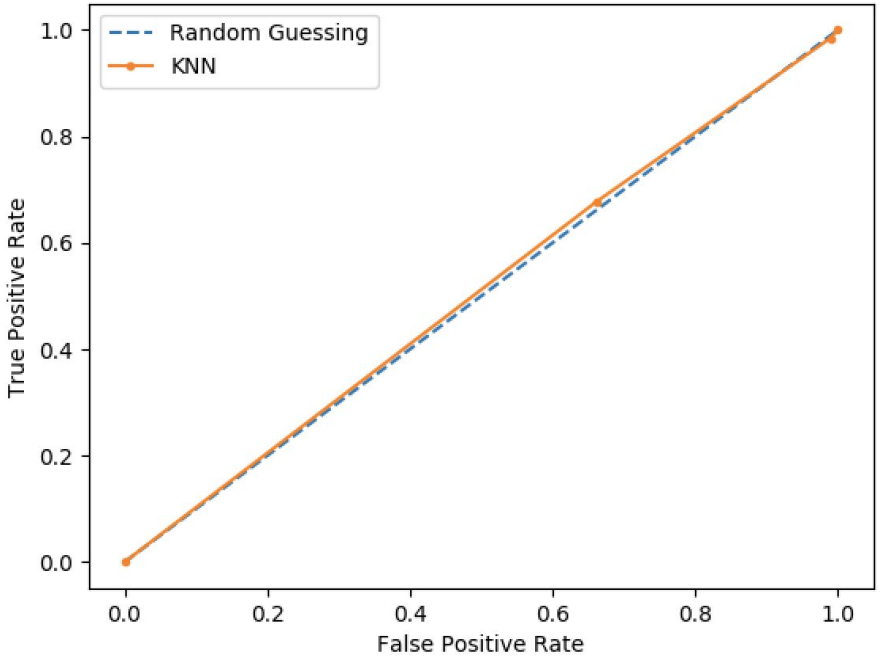


Bigram
(Accuracy = **57%**, AUC = **0.591**)

k-NN

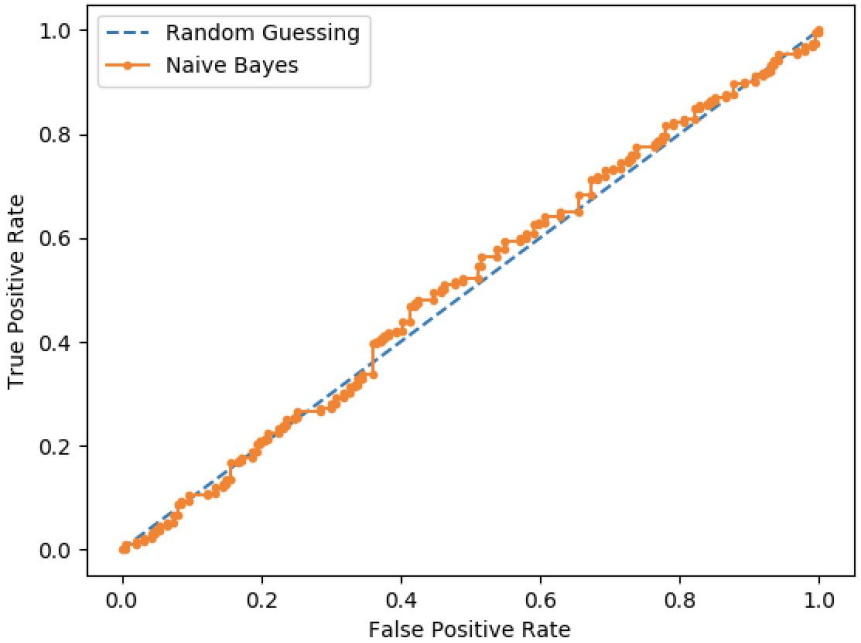


Unigram + TDIF
(Accuracy = **53%**, AUC = **0.534**)

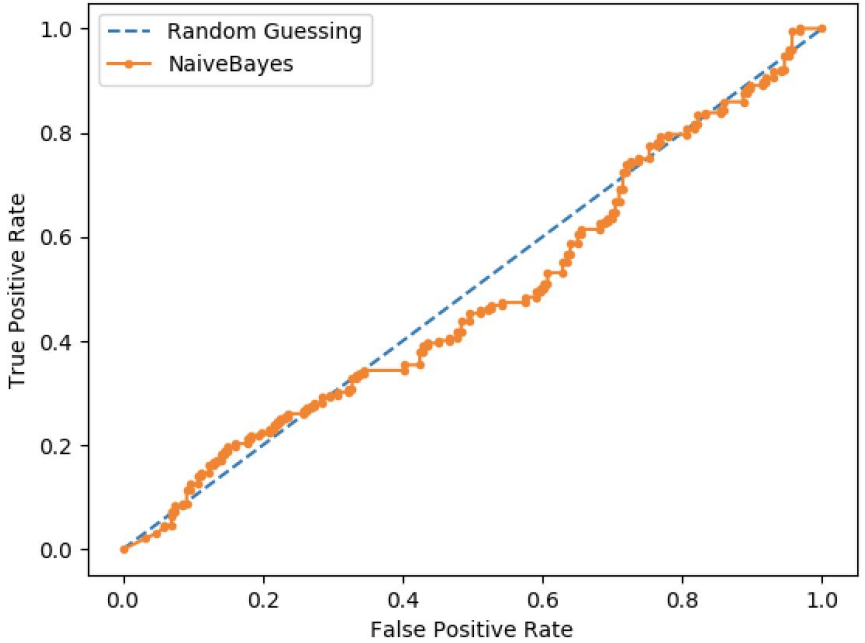


Bigram
(Accuracy = **51%**, AUC = **0.507**)

Naive Bayes

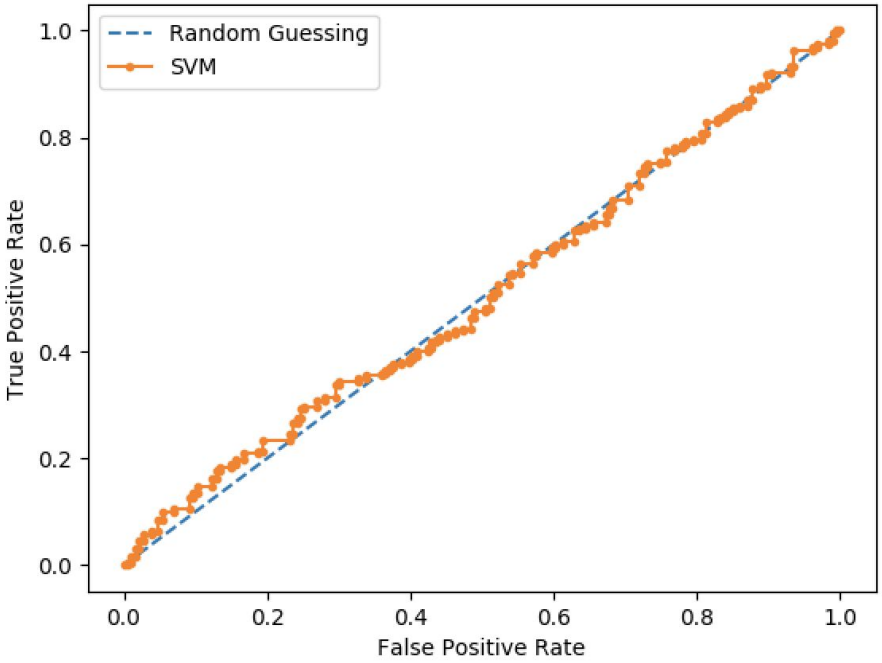


Unigram + TDIF
(Accuracy = **52%**, AUC = **0.511**)

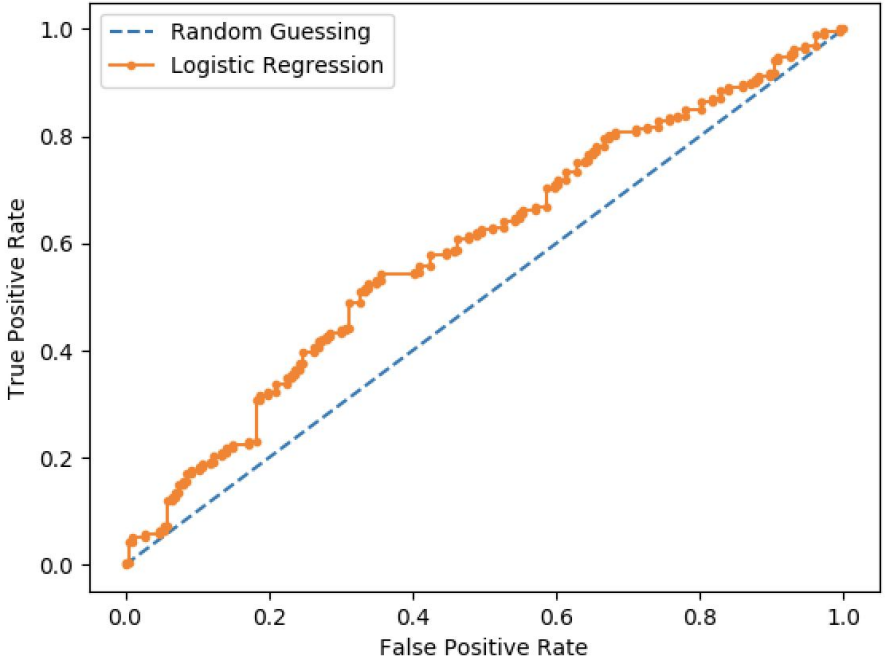


Bigram
(Accuracy = **50%**, AUC = **0.481**)

SVM

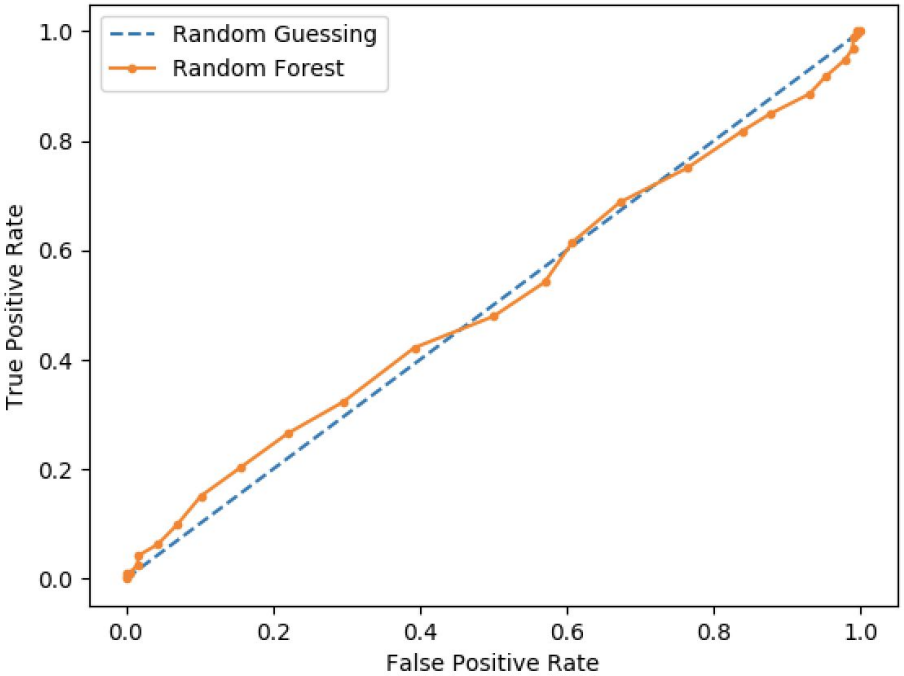


Unigram + TDIF
(Accuracy = **51%**, AUC = **0.506**)

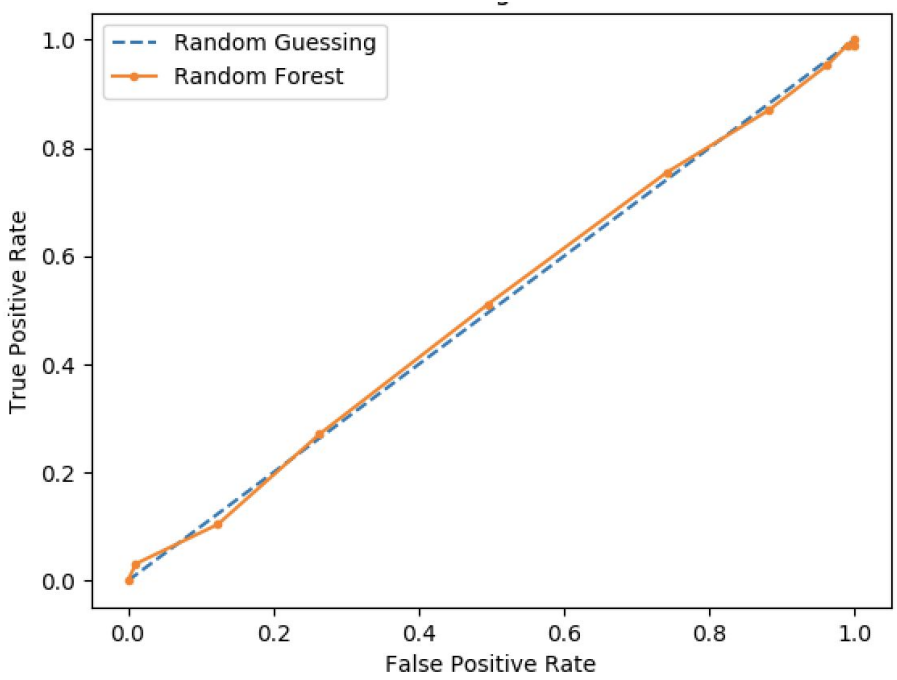


Bigram
(Accuracy = **56%**, AUC = **0.554**)

Random Forest

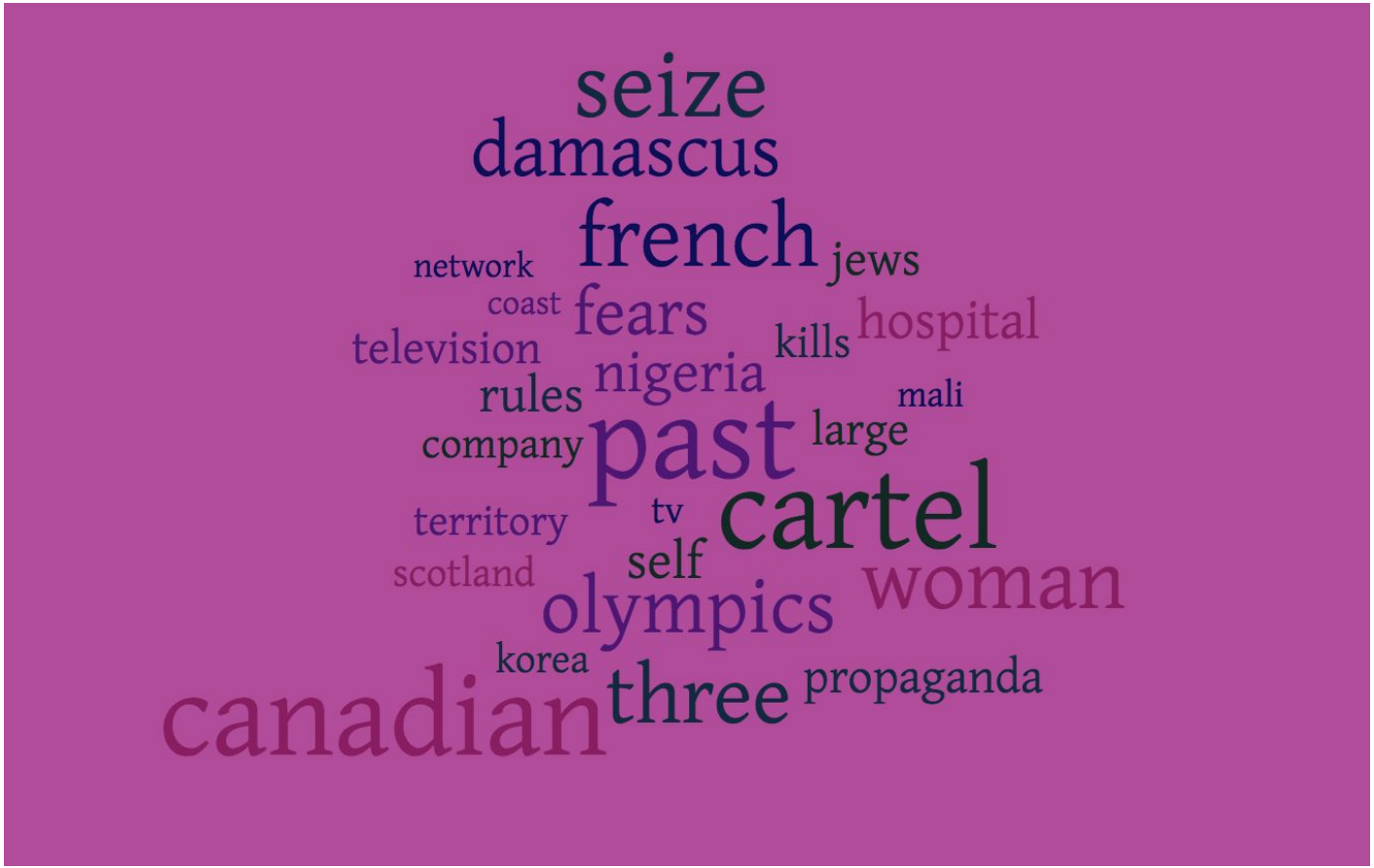


Unigram + TDIF
(Accuracy = **50-52%**, AUC = **0.523**)



Bigram
(Accuracy = **51-55%**, AUC = **0.531**)

Negative Unigrams (lowest coefs in Log Reg)



Positive Unigrams (highest coefs in Log Reg)

Negative Bigrams

Word	Coefficient
in iran	-0.165284933
wall street	-0.166076214
if it	-0.166630495
to take	-0.173657266
sex with	-0.17569238
us and	-0.175703342
phone hacking	-0.175723773
nuclear weapons	-0.180257798
been arrested	-0.183213618
the german	-0.183467924
sexual abuse	-0.18349989
threatens to	-0.185189163
south korean	-0.18597363
children in	-0.186817292
in gaza	-0.188524334
that is	-0.188919959
bin laden	-0.188931567
10 000	-0.190050694

Positive Bigrams

Word	Coefficient
in china	0.227942799
in south	0.224184337
found in	0.219129965
forced to	0.216726182
new zealand	0.208993401
after the	0.205478571
the pope	0.201760936
time in	0.194867776
embassy in	0.191785523
that they	0.191704002
are the	0.190300156
what the	0.189657878
if they	0.185001155
in nigeria	0.183183153
the french	0.17900413
israel and	0.178629376
during the	0.17620462
security council	0.175774605
dozens of	0.17427957
to the	0.174113232

Conclusions & Future Work

- Unigram + TFIDF: **k-NN**
- Bigram: **Logistic Regression, SVM**

- Natural Language Processing (NLP) techniques

- Future directions:
  - get more data
  - rely on previous dates to predict better
  - reduce biases (e.g., vocab usage changes)
  - better performance?