Tweets Sentiment Analysis of 2016 U.S. Presidential Election

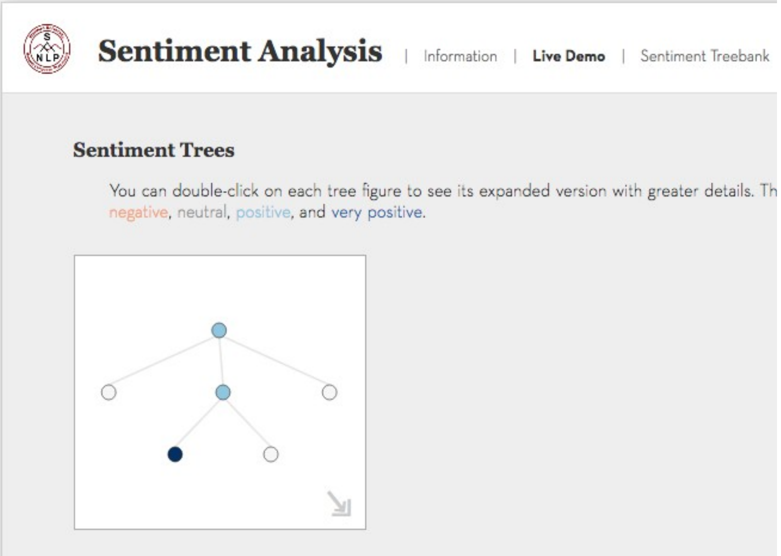
Bin Li
Computer Science Department
New York University

What is sentimental analysis?

- It is a special case of text mining generally focused on opinion polarity
- The goal is to determine if the text opinion is positive, negative or neutral

Why sentimental analysis?

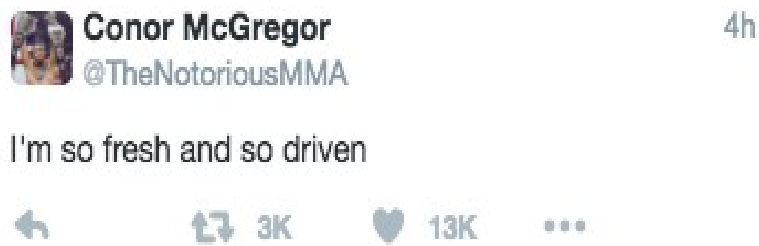
- Microblogging has been a central place for people to express their thoughts and opinions on political parties and candidates
- Provide investment guidance
- A company wants to know the reviews of their products. (restaurant reviews)



http://sentiment.vivekn.com/

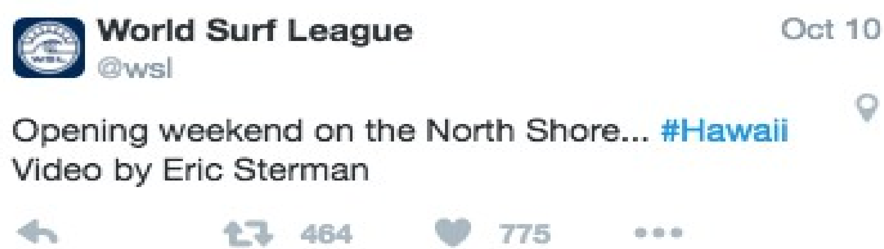
http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

Opinion Polarity



Positive

Neutral



Negative

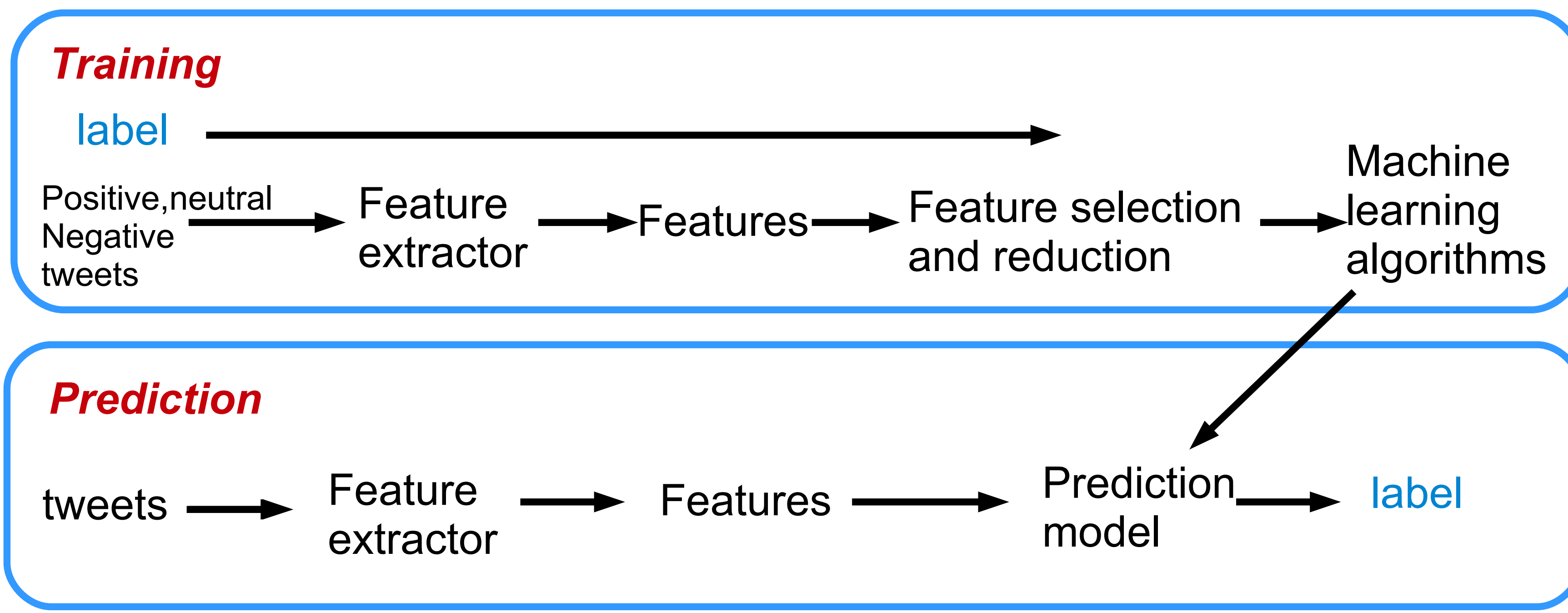
Problem Statement

- Given a tweet, determine if this tweet is positive, negative or neutral sentiment to a candidate.

Design Diagram

Data preparation

- Tweets Collection: tweets related to election through Twitter APIs
- Data Pre-processing: tokenization, feature reduction (remove non-english words, URLs, stop words and quotation, stemming, repeated characters etc.)
- Aggregate tweets to candidate: map tweets to the corresponding candidate



Data understanding

Tweets to be labeled

How? Keyword search using Twitter Search API and Stream API. On the second and last presidential debate date

Volume: 9.6 million tweets collected and hosted on HPC cluster

Type: highly unstructured text data

Data understanding

<u>Training dataset</u>

10,000 labeled tweets, Combination of the following two dataset:

- Stanford sentiment140

  Label classified based on emoji, no neutral labeled data

```
"4","2192559582","Tue Jun 16 07:12:59 PD 2009","NO_QUERY","yogilo","@EliciaKoay
you went overboard for the girl's birthday again "
```

- Sanders-twitter

  Manually labelled by human whose english is very well.
  Based on specific topics, IT company like apple, google

```
“apple”, "neutral","125828984293425152","Mon Oct 17 07:02:22 +0000 2011","ok it
is back @iphone @apple"
```

My training dataset includes positive and negative tweets from sentiment140 and neutral tweets from Sanders.

Data Preparation

- Remove irrelevant info such as URLs, quotes, citations and numbers
- Tokenization using tweet-preprocessor
- Remove punctuation
- Remove stop words
- Apply porter stemmer
- Finally, remove any word whose length is smaller than 3

For tweet,

```
rt @weneedfeminlsm: the way donald trump treats women is disgusting
and repulsive. nothing about this is ok. https://t.co/lmbjgopgnt
```

After preprocessing,

```
way donald trump treat women disgust repuls noth ok
```

Do all above steps for both training tweets and tweets to be labeled.

Data Preparation

Implement Naive Bayes algorithm

Text model: unigram or bigram

n*m

	f1	f2	f3	.	.	.	.	.	.	.	.	.	.	fm-1	fm
t1	2	4							7						
t2			6								6				
t3		4			6			3	9			e			
t4										4					
t5	2			7		2	13				6				65
.	3			25						45					
.								12		8			5	16	
.			65		2	9					4				
.				1			32	13				78			
.		87			45				4						
tn-1								7		23					65
tn					6								8		

Model training

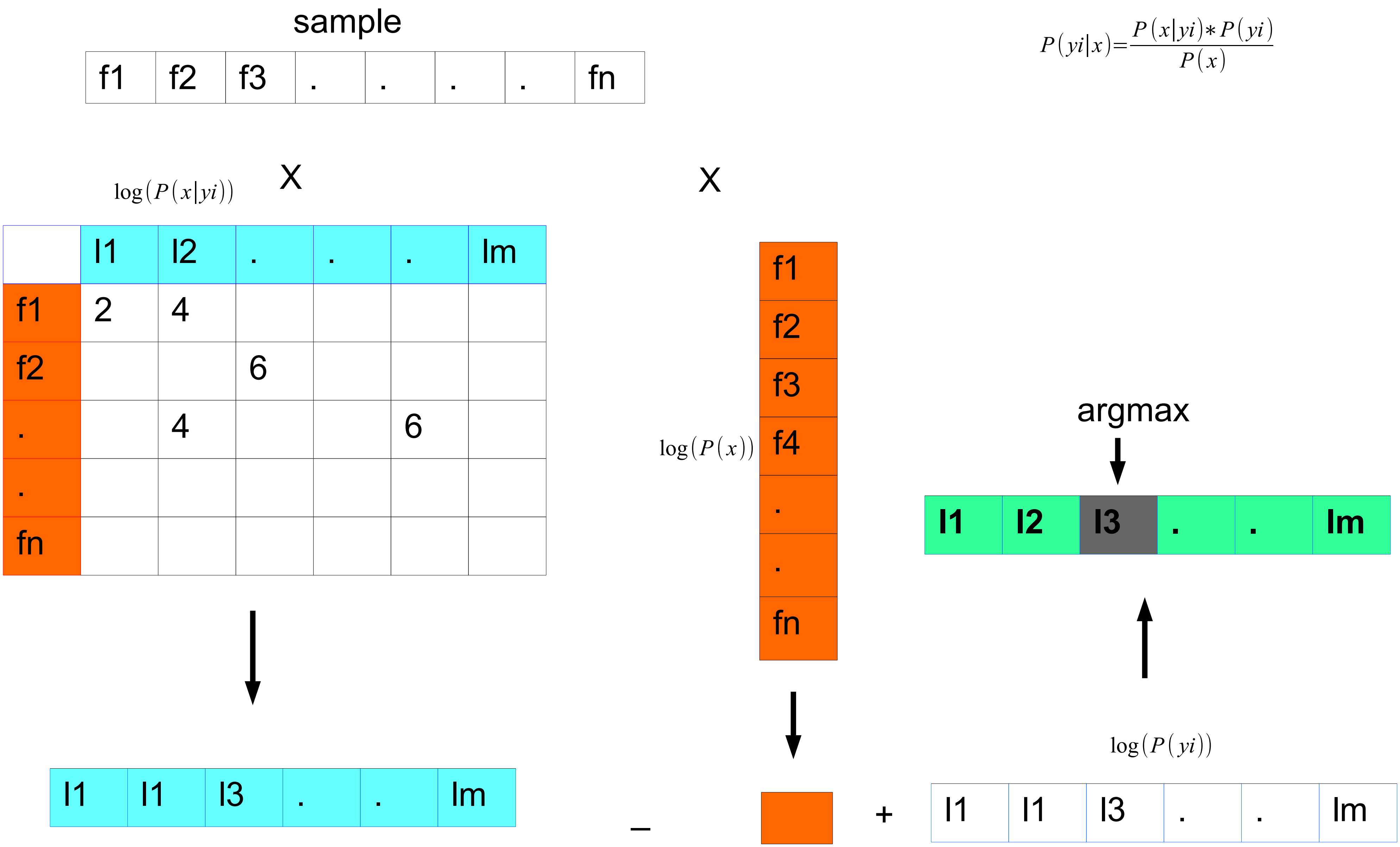
$$P(yi|x)=\frac{P(x|yi)*P(yi)}{P(x)}, x=a1, a2, \ldots, an$$

$$\log(P(yi|x))=\log(P(x|yi))+\log(P(yi))-\log(P(x))$$

The goal is to get three prior probabilities that will be used by classification.

- class_prior (P(yi))
  ----
  the probability of each label
- feature_prior (P(x))
  ----
  the probability of each feature in all feature space
- label_feature_matrix (P(x | yi))
  ----
  the probability of each feature under certain label

Model training



Stats

K-fold validation, 4216 features based on 500 training tweets using freq=2

```
************Feature Freq=2*********
```

```
unigram stats
------------label:neutral-------
tp: 86, tn: 301, fp: 4, fn: 4
accuracy:    0.980
precison:    0.956
recall:      0.956
--------------------------------
------------label:positive------
tp: 129, tn: 253, fp: 7, fn: 4
accuracy:    0.972
precison:    0.949
recall:      0.970
--------------------------------
------------label:negative------
tp: 121, tn: 280, fp: 2, fn: 5
accuracy:    0.983
precison:    0.984
recall:      0.960
----------------------------
```

```
bigram stats
------------label:neutral-------
tp: 110, tn: 157, fp: 25, fn: 0
accuracy:    0.914
precison:    0.815
recall:      1.000
--------------------------------
------------label:positive------
tp: 63, tn: 273, fp: 4, fn: 9
accuracy:    0.963
precison:    0.940
recall:      0.875
--------------------------------
------------label:negative------
tp: 59, tn: 300, fp: 0, fn: 20
accuracy:    0.947
precison:    1.000
recall:      0.747
--------------------------------
```

Stats

K-fold validation, 2612 features based on 500 training tweets using freq=3

```
************Feature Freq=3*********
```

```
unigram stats
------------label:neutral-------
tp: 91, tn: 290, fp: 12, fn: 3
accuracy:    0.962
precison:    0.883
recall:      0.968
--------------------------------
------------label:positive------
tp: 128, tn: 266, fp: 2, fn: 11
accuracy:    0.968
precison:    0.985
recall:      0.921
--------------------------------
------------label:negative------
tp: 124, tn: 285, fp: 3, fn: 3
accuracy:    0.986
precison:    0.976
recall:      0.976
--------------------------------
```

```
bigram stats
----------label:neutral--------
tp: 116, tn: 106, fp: 22, fn: 3
accuracy:    0.899
precison:    0.841
recall:      0.975
-------------------------------
---------label:positive--------
tp: 46, tn: 287, fp: 5, fn: 9
accuracy:    0.960
precison:    0.902
recall:      0.836
-------------------------------
-----------label:negative------
tp: 43, tn: 310, fp: 1, fn: 16
accuracy:    0.954
precison:    0.977
recall:      0.729
-------------------------------
```

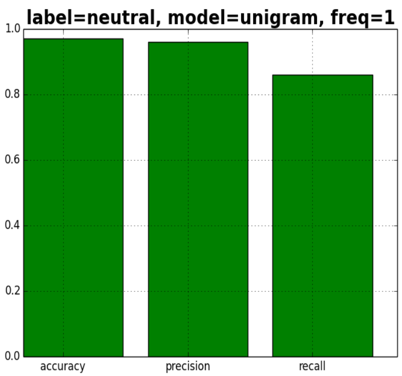
Performance: freq(2) > freq(3), unigram > bigram

Stats

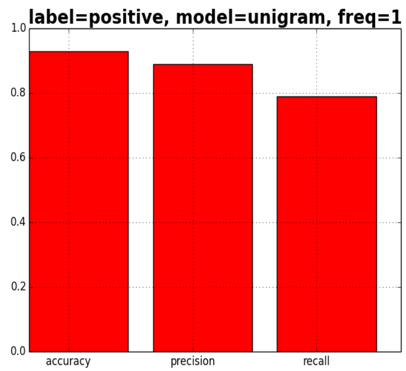
Feature selection

Based on 10,000 training tweets, compare feature frequency

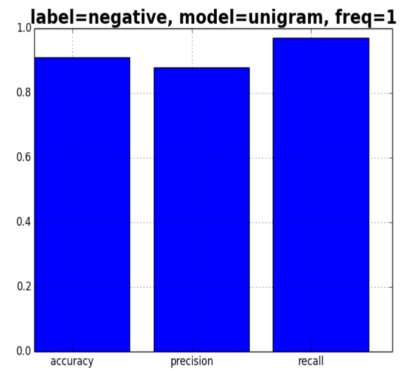
neutral



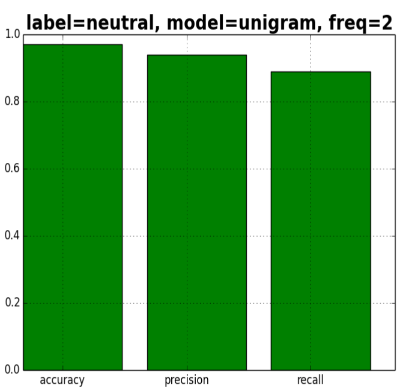
positive

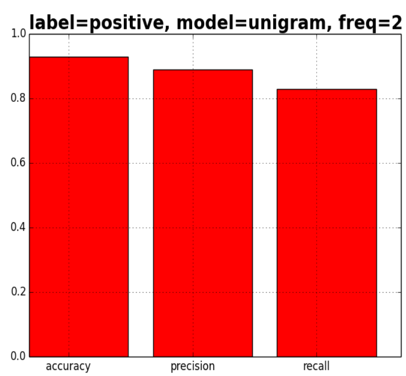


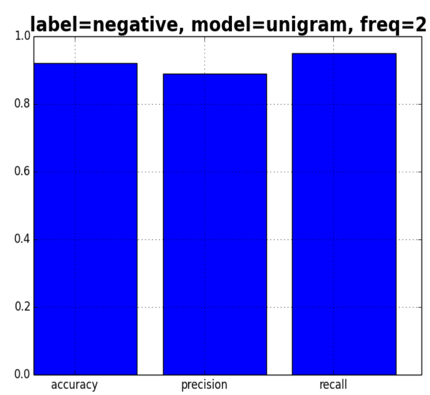
negative



Performance:
Freq=1 and freq=2 very close
so we choose freq=2







PCA

	f1	f2	f3	.	.	.	.	.	fm
t1	2	4							7
t2			6						
t3		4			6			3	9
t4									
.	2			7		2	13		
.	3			25					
tn								12	

	f1	f2	.	fk
t1				
t2				
t3				
t4				
.				
.				
tn				

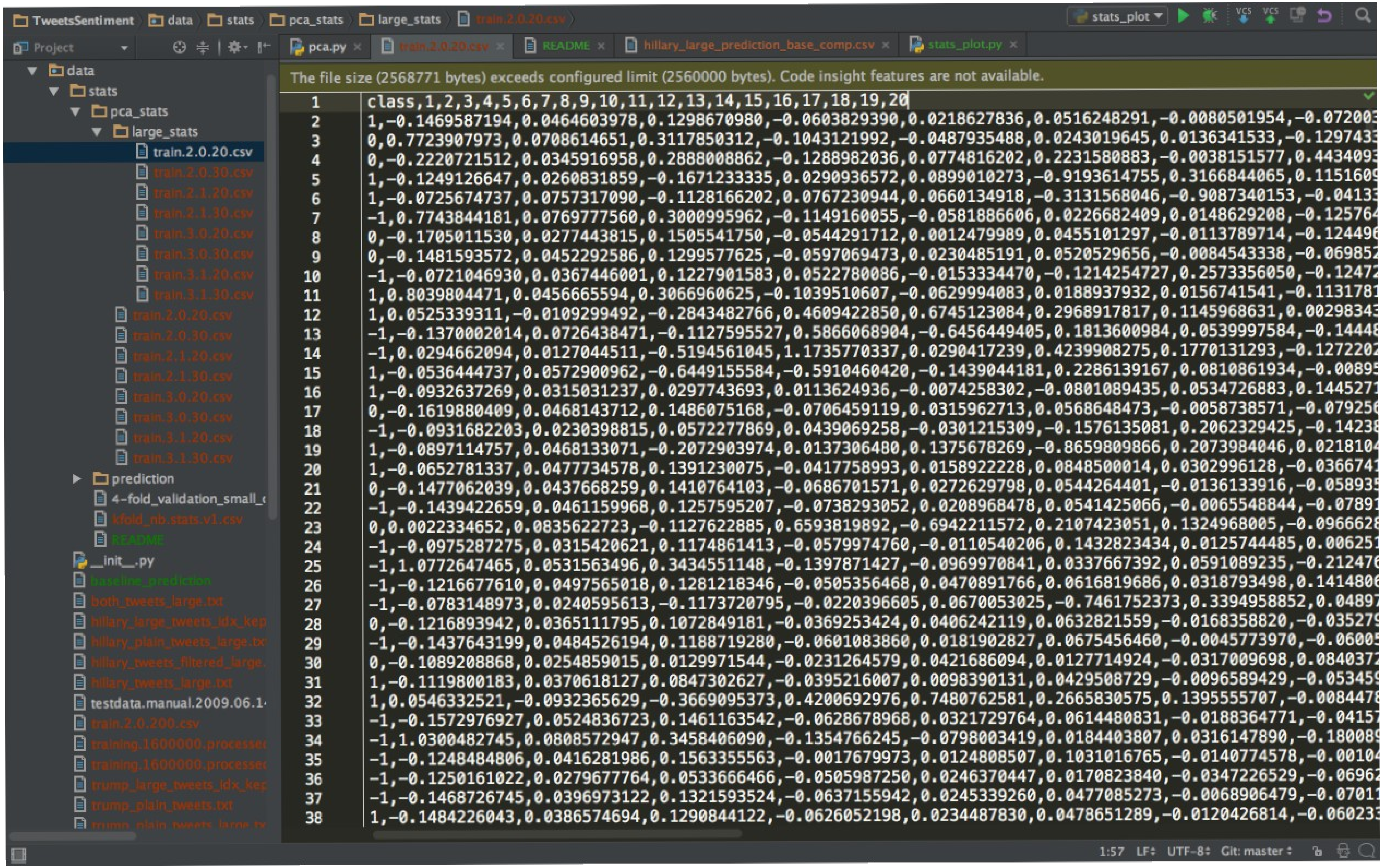
n*m (m = 5000)

n*k (k = 20, 30, 40)

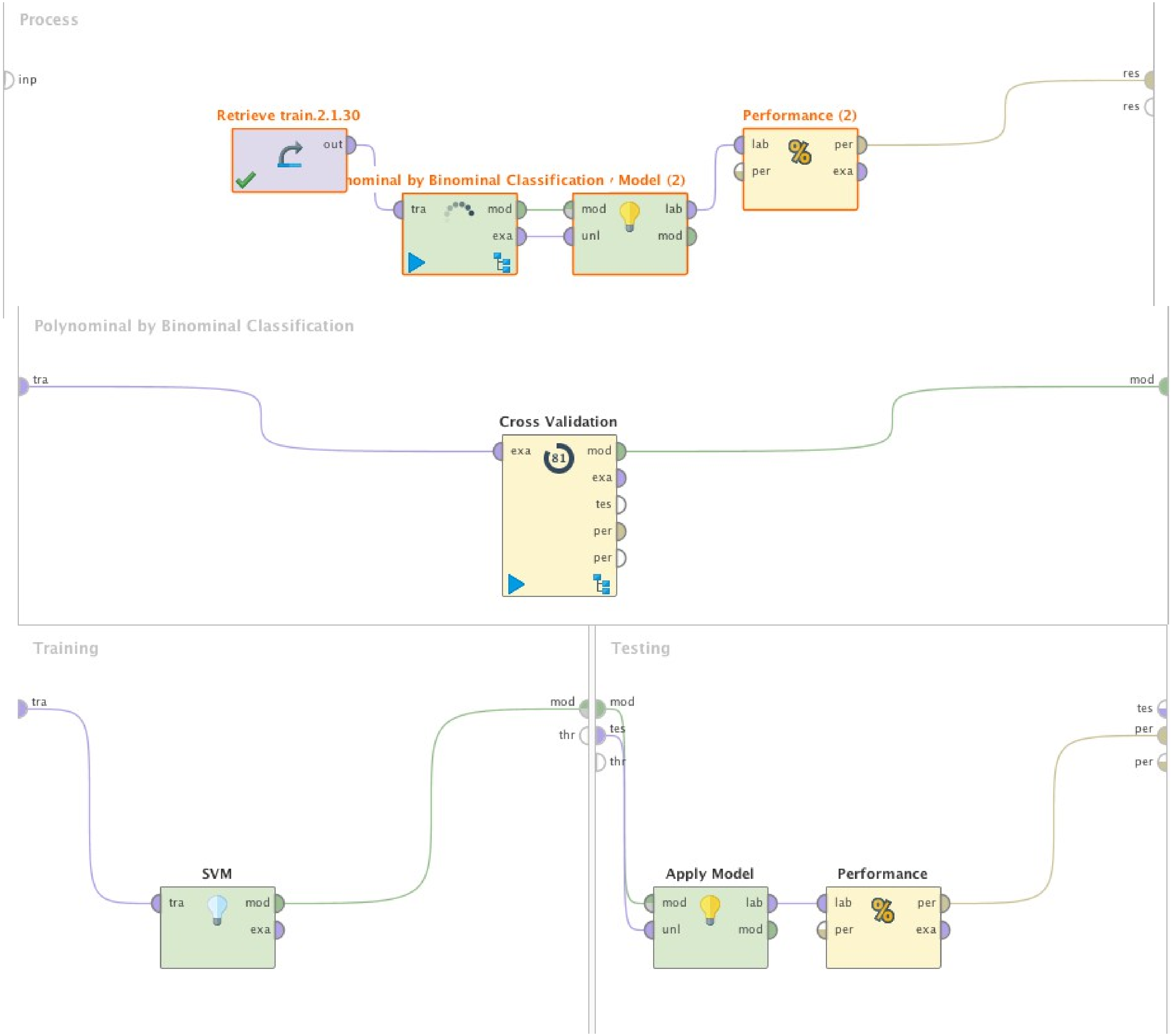
First k PCs

normalization → Covariance matrix → Eigen vectors

PCA


TweetsSentiment
data
stats
pca_stats
large_stats
train.2.0.20.csv
stats_plot
Project
pca.py
README
hillary_large_prediction_base_comp.csv
stats_plot.py
The file size (2568771 bytes) exceeds configured limit (2560000 bytes). Code insight features are not available.
class,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
1,-0.1469587194,0.0464603978,0.1298670980,-0.0603829390,0.0218627836,0.0516248291,-0.0080501954,-0.072003
0,0.7723907973,0.0708614651,0.3117850312,-0.1043121992,-0.0487935488,0.0243019645,0.0136341533,-0.1297433
0,-0.2220721512,0.0345916958,0.2888008862,-0.1288982036,0.0774816202,0.2231580883,-0.0038151577,0.4434093
1,-0.1249126647,0.0260831859,-0.1671233335,0.0290936572,0.0899010273,-0.9193614755,0.3166844065,0.1151609
1,-0.0725674737,0.0757317090,-0.1128166202,0.0767230944,0.0660134918,-0.3131568046,-0.9087340153,-0.04133
-1,0.7743844181,0.0769777560,0.3000995962,-0.1149160055,-0.0581886606,0.0226682409,0.0148629208,-0.125764
0,-0.1705011530,0.0277443815,0.1505541750,-0.0544291712,0.0012479989,0.0455101297,-0.0113789714,-0.124496
0,-0.1481593572,0.0452292586,0.1299577625,-0.0597069473,0.0230485191,0.0520529656,-0.0084543338,-0.069852
-1,-0.0721046930,0.0367446001,0.1227901583,0.0522780086,-0.0153334470,-0.1214254727,0.2573356050,-0.12472
1,0.8039804471,0.0456665594,0.3066960625,-0.1039510607,-0.0629994083,0.0188937932,0.0156741541,-0.1131781
1,0.0525339311,-0.0109299492,-0.2843482766,0.4609422850,0.6745123084,0.2968917817,0.1145968631,0.00298343
-1,-0.1370002014,0.0726438471,-0.1127595527,0.5866068904,-0.6456449405,0.1813600984,0.0539997584,-0.14448
-1,0.0294662094,0.0127044511,-0.5194561045,1.1735770337,0.0290417239,0.4239908275,0.1770131293,-0.1272202
1,-0.0536444737,0.0572900962,-0.6449155584,-0.5910460420,-0.1439044181,0.2286139167,0.0810861934,-0.00895
1,-0.0932637269,0.0315031237,0.0297743693,0.0113624936,-0.0074258302,-0.0801089435,0.0534726883,0.1445271
0,-0.1619880409,0.0468143712,0.1486075168,-0.0706459119,0.0315962713,0.0568648473,-0.0058738571,-0.079256
-1,-0.0931682203,0.0230398815,0.0572277869,0.0439069258,-0.0301215309,-0.1576135081,0.2062329425,-0.14238
1,-0.0897114757,0.0468133071,-0.2072903974,0.0137306480,0.1375678269,-0.8659809866,0.2073984046,0.0218104
1,-0.0652781337,0.0477734578,0.1391230075,-0.0417758993,0.0158922228,0.0848500014,0.0302996128,-0.0366741
0,-0.1477062039,0.0437668259,0.1410764103,-0.0686701571,0.0272629798,0.0544264401,-0.0136133916,-0.058935
-1,-0.1439422659,0.0461159968,0.1257595207,-0.0738293052,0.0208968478,0.0541425066,-0.0065548844,-0.07891
0,0.0022334652,0.0835622723,-0.1127622885,0.6593819892,-0.6942211572,0.2107423051,0.1324968005,-0.0966628
-1,-0.0975287275,0.0315420621,0.1174861413,-0.0579974760,-0.0110540206,0.1432823434,0.0125744485,0.006251
-1,1.0772647465,0.0531563496,0.3434551148,-0.1397871427,-0.0969970841,0.0337667392,0.0591089235,-0.212476
-1,-0.1216677610,0.0497565018,0.1281218346,-0.0505356468,0.0470891766,0.0616819686,0.0318793498,0.1414806
-1,-0.0783148973,0.0240595613,-0.1173720795,-0.0220396605,0.0670053025,-0.7461752373,0.3394958852,0.04897
0,-0.1216893942,0.0365111795,0.1072849181,-0.0369253424,0.0406242119,0.0632821559,-0.0168358820,-0.035279
-1,-0.1437643199,0.0484526194,0.1188719280,-0.0601083860,0.0181902827,0.0675456460,-0.0045773970,-0.06005
0,-0.1089208868,0.0254859015,0.0129971544,-0.0231264579,0.0421686094,0.0127714924,-0.0317009698,0.0840372
1,-0.1119800183,0.0370618127,0.0847302627,-0.0395216007,0.0098390131,0.0429508729,-0.0096589429,-0.053459
1,0.0546332521,-0.0932365629,-0.3669095373,0.4200692976,0.7480762581,0.2665830575,0.1395555707,-0.0084478
-1,-0.1572976927,0.0524836723,0.1461163542,-0.0628678968,0.0321729764,0.0614480831,-0.0188364771,-0.04157
-1,1.0300482745,0.0808572947,0.3458406090,-0.1354766245,-0.0798003419,0.0184403807,0.0316147890,-0.180089
-1,-0.1248484806,0.0416281986,0.1563355563,-0.0017679973,0.0124808507,0.1031016765,-0.0140774578,-0.00104
-1,-0.1250161022,0.0279677764,0.0533666466,-0.0505987250,0.0246370447,0.0170823840,-0.0347226529,-0.06962
-1,-0.1468726745,0.0396973122,0.1321593524,-0.0637155942,0.0245339260,0.0477085273,-0.0068906479,-0.07011
1,-0.1484226043,0.0386574694,0.1290844122,-0.0626052198,0.0234487830,0.0478651289,-0.0120426814,-0.060233
prediction
__init__.py
testdata.manual.2009.06.1
1:57 LF UTF-8 Git: master

PCA



Process
inp
Retrieve train.2.1.30
out
nominal by Binominal Classification
Model (2)
tra
mod
exa
mod
unl
lab
mod
Performance (2)
lab
per
per
exa
res
res
Polynominal by Binominal Classification
tra
mod
Cross Validation
exa
mod
exa
tes
per
per
Training
tra
mod
thr
SVM
tra
mod
exa
Testing
mod
tes
thr
tes
per
per
Apply Model
mod
unl
lab
mod
Performance
lab
per
per
exa

PCA

Train(freq=2, model=unigram, PCs=20), Based on 10,000 training data

accuracy: 55.21%

	true 1	true 0	true -1	class precision
pred. 1	539	148	190	61.46%
pred. 0	76	723	60	84.17%
pred. -1	2383	1371	3949	51.27%
class recall	17.98%	32.25%	94.05%	

Train(freq=2, model=unigram, PCs=30), based on 10,000 training data

accuracy: 55.44%

	true -1	true 1	true 0	class precision
pred. -1	3914	2339	1355	51.45%
pred. 1	201	584	152	62.33%
pred. 0	84	75	735	82.21%
class recall	93.21%	19.48%	32.78%	

PCA

Train(freq=2, model=bigram, PCs=20), Based on 10,000 training data

accuracy: 48.83%

	true 0	true -1	true 1	class precision
pred. 0	389	14	17	92.62%
pred. -1	1792	4099	2860	46.84%
pred. 1	61	86	121	45.15%
class recall	17.35%	97.62%	4.04%	

Train(freq=2, model=bigram, PCs=30), Based on 10,000 training data

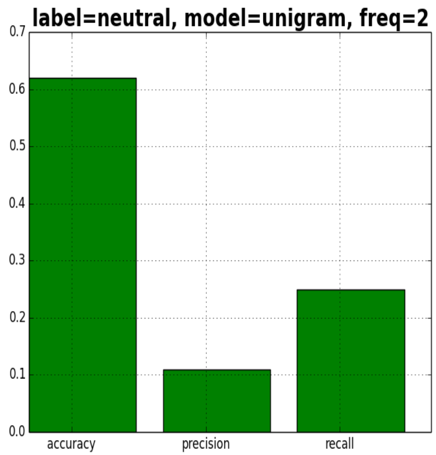
accuracy: 49.25%

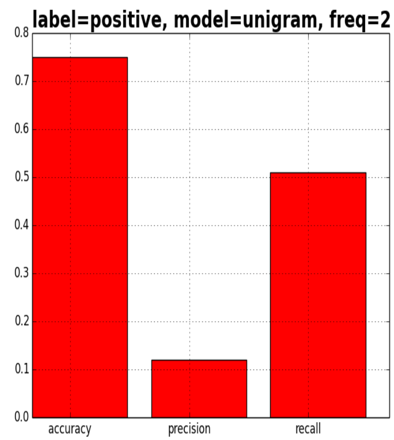
	true 1	true -1	true 0	class precision
pred. 1	111	20	3	82.84%
pred. -1	2869	4162	1863	46.80%
pred. 0	18	17	376	91.48%
class recall	3.70%	99.12%	16.77%	

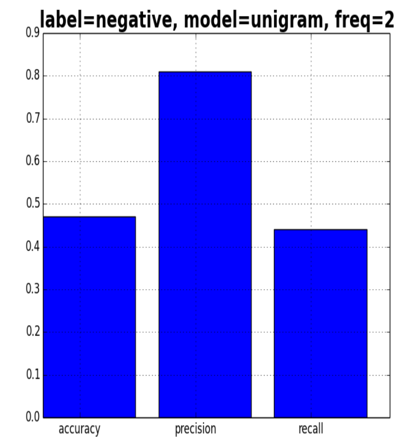
Evaluation

Model=unigram, feature frequency=2

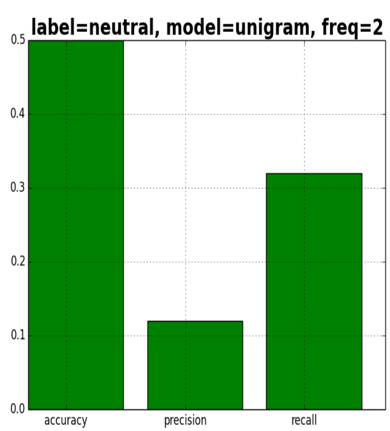
Trump, compare with baseline, based on 10972 tweets

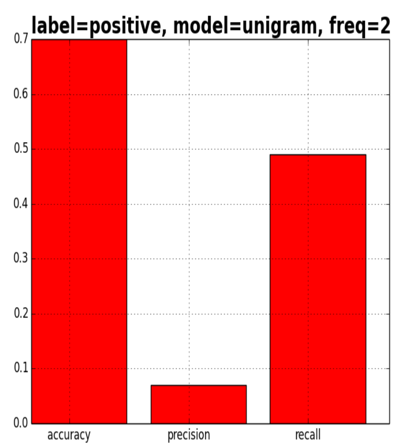


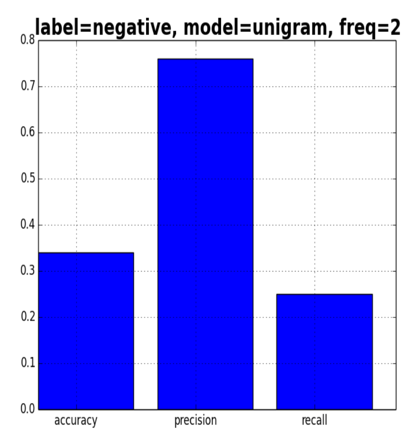


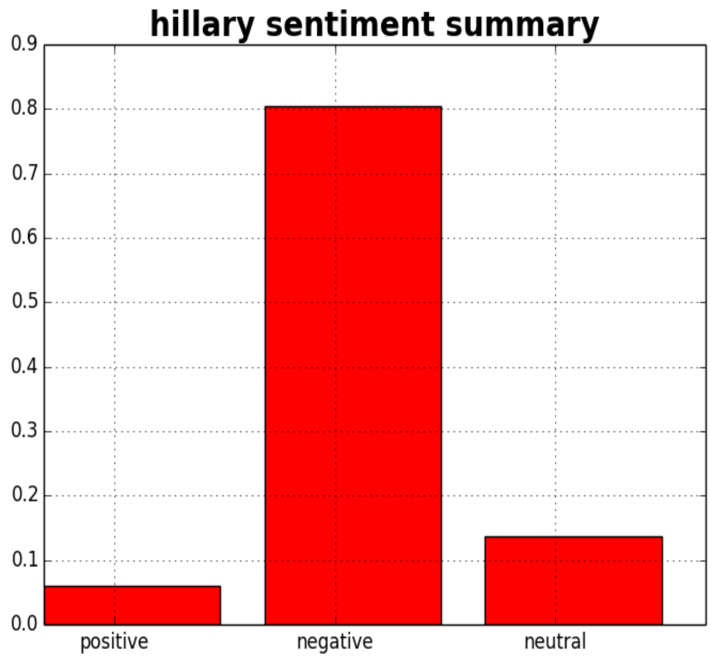


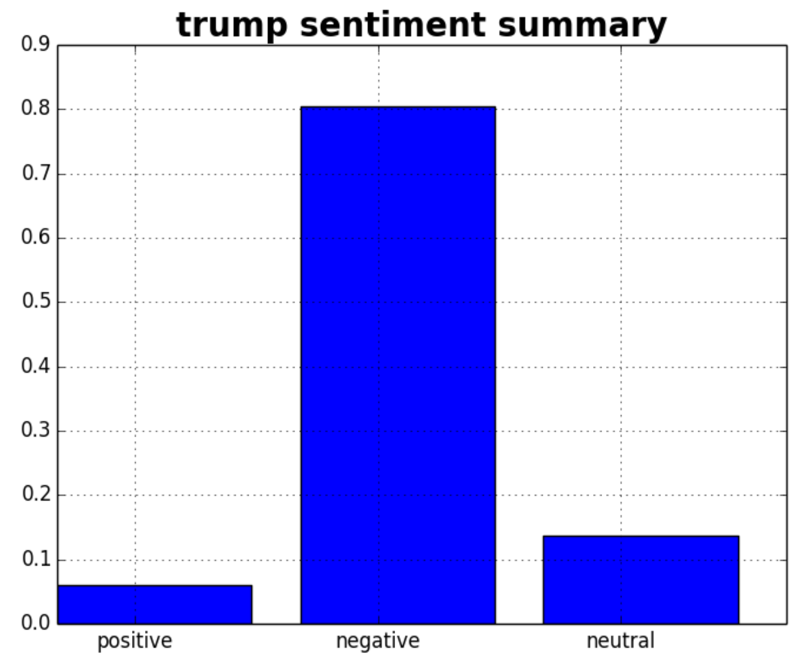
Hillary, compare with baseline, based on 8941 tweets











Sentiment summary computed by Hadoop Map/Reduce, based on 9.6 million tweets

Deployment

- Write tweets and their predicted sentiment to HBase
- Front end UI to use JavaScript to make REST call to HBase and virtualize tweet sentiment

hillari:2016-11-25 16:36:50_1657   column=data:sentiment, timestamp=1480136553022, value=positive

hillari:2016-11-25 16:36:50_1657   column=data:tweet, timestamp=1480136553022, value=rt @foxnews: .@judgenap: new fbi docs show 'bribe offer to agents in #hillaryclinton email probe https://t.co/49cj9rgdun https://t.co/svfd\x0A

Current standing

Data preparation

- Collected 4,000,000 tweets using keywords search. Starting from the second debase. Data is hosted on NYU HPC clusters.
- Downloaded labelled tweets (1.6 million) as training data from Stanford Sentiment140
- Performed tokenization, removed stop words, URLs, repeated characters and quotation, stemming

Acknowledgement

- Stanford sentiment140
- Sanders-twitter
- NYU HPC

<u>helpful links:</u>

http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/

http://adilmoujahid.com/posts/2014/07/twitter-analytics/