



Brianna Lytle - De'Varus May - Sibel Tanoglu

DSI-CC9-LA

Problem/Task:

- Create an alert system that utilizes social media activity to identify when an emergency first occurs

Focus: California wildfires



Data:



- Data were collected based on date and geotag
 - Tweepy (API)
 - GetOldTweets3 (web scraper)
- Pre-fire tweets (2019 September)
- Post-fire tweets (2019 October)
 - Specific Fire : Saddleridge, Tick, Kincade, Getty, Maria

Total Data: 24,410 Tweets

Related Tweets: 5,691 (23.3%)

Unrelated Tweets: 18,719 (76.6%)



Common Tri-grams

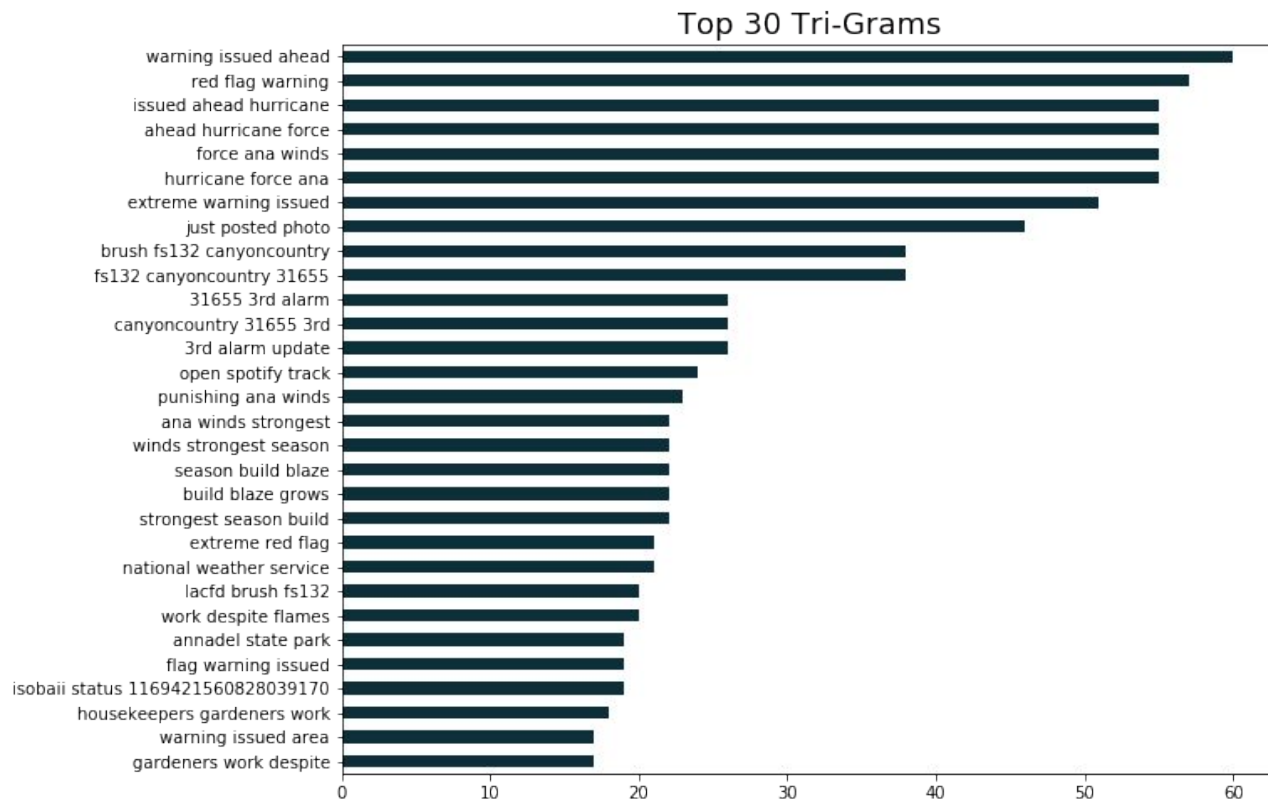
Weather related

“Warnings”

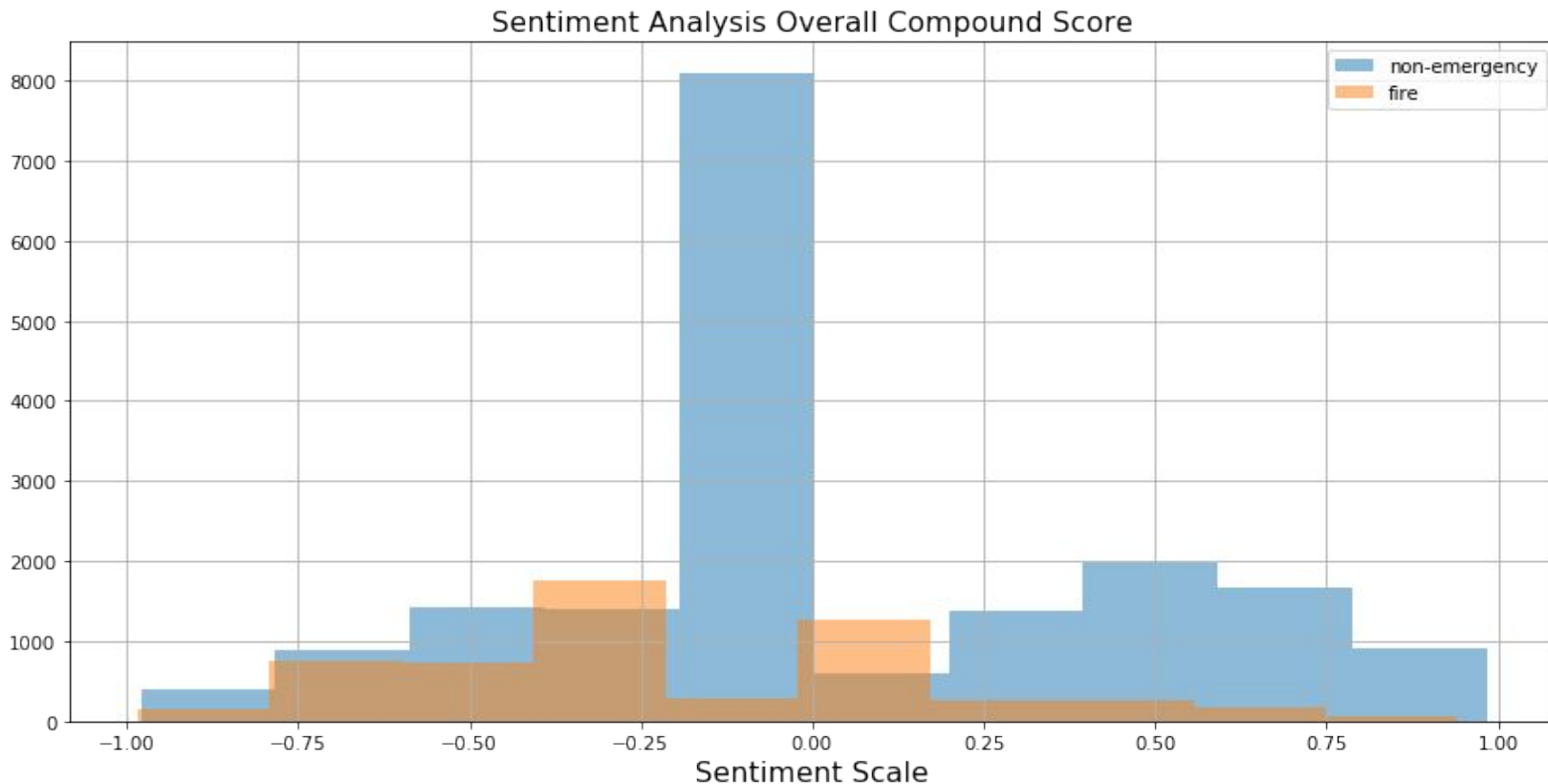
Los Angeles - Santa

Ana winds/

Hurricane winds

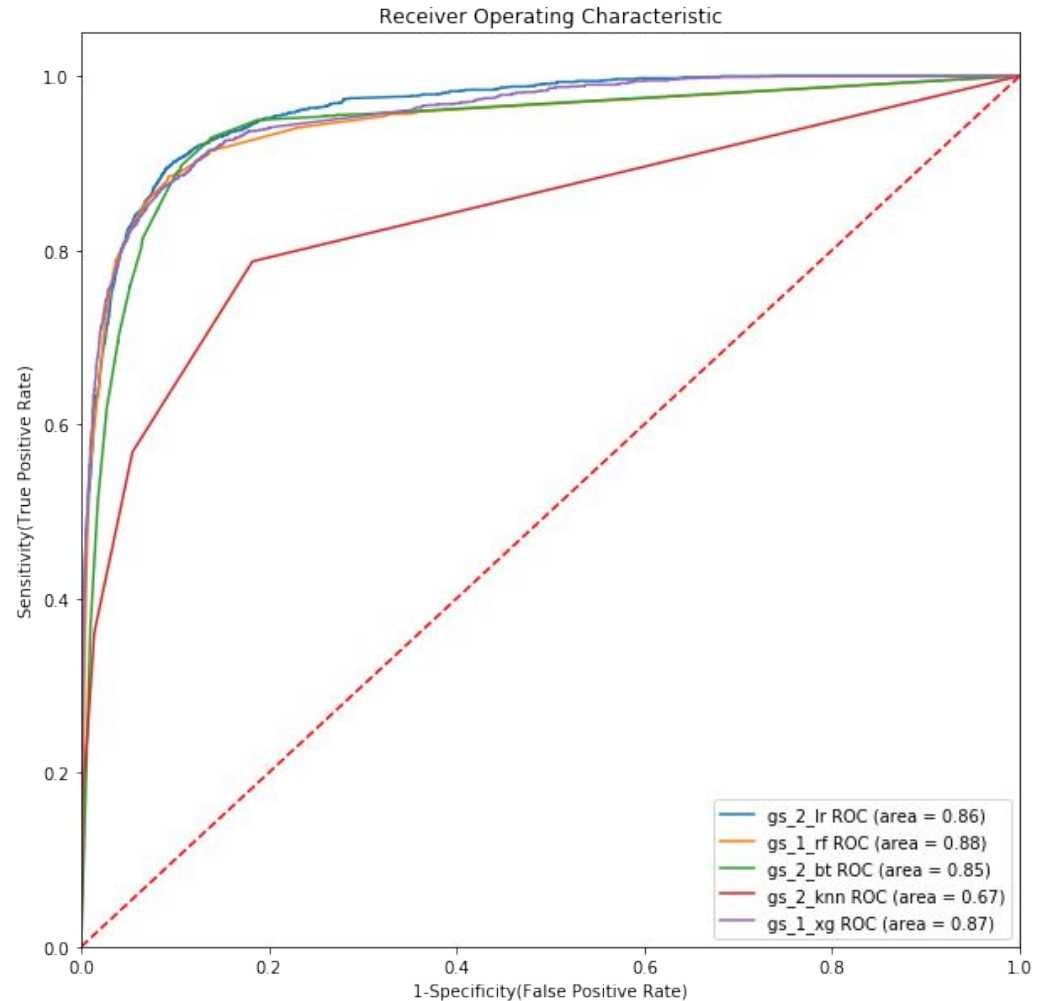


Overall Sentiment Score of Tweets



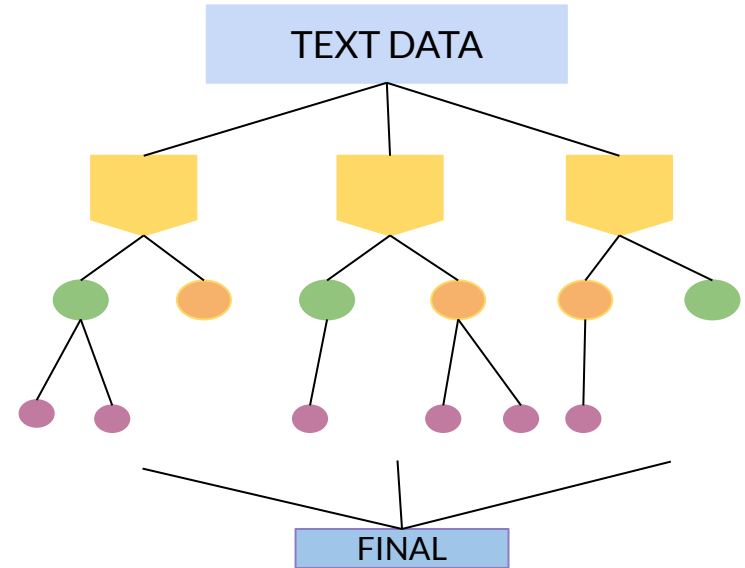
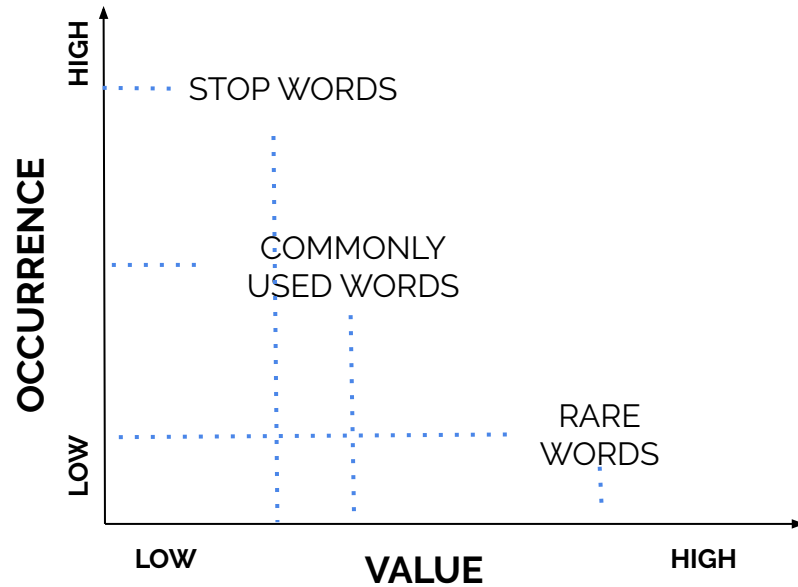
Model Estimators:
Logistic Regression
Random Forest
Bagging
K-nearest neighbors
XGBoost

Best Model:
TF-idf + Random Forest
X = text data



Model Transformer: Tf-IDF

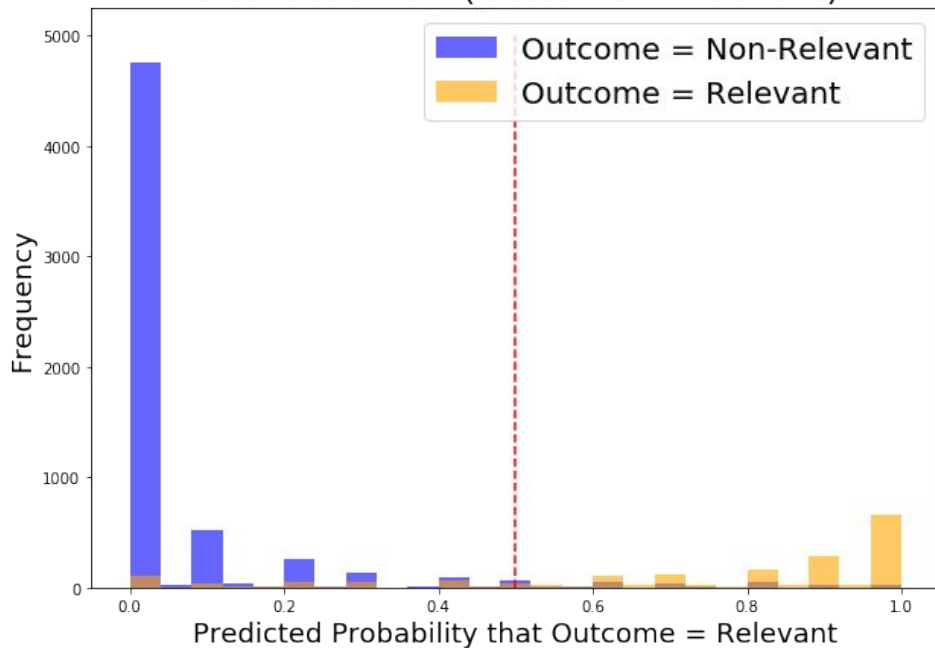
Model Estimator: Random Forest



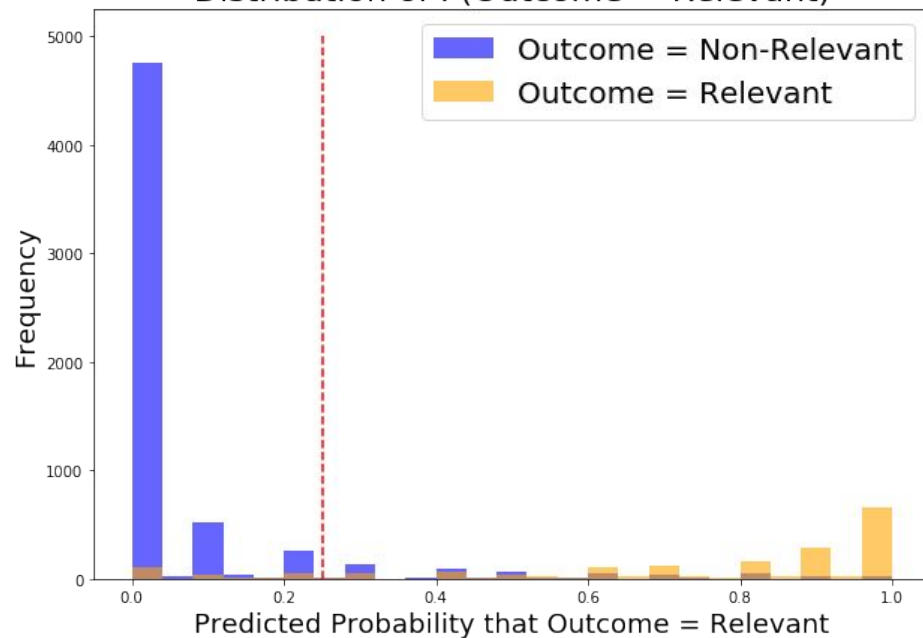
Model Accuracy: .92217

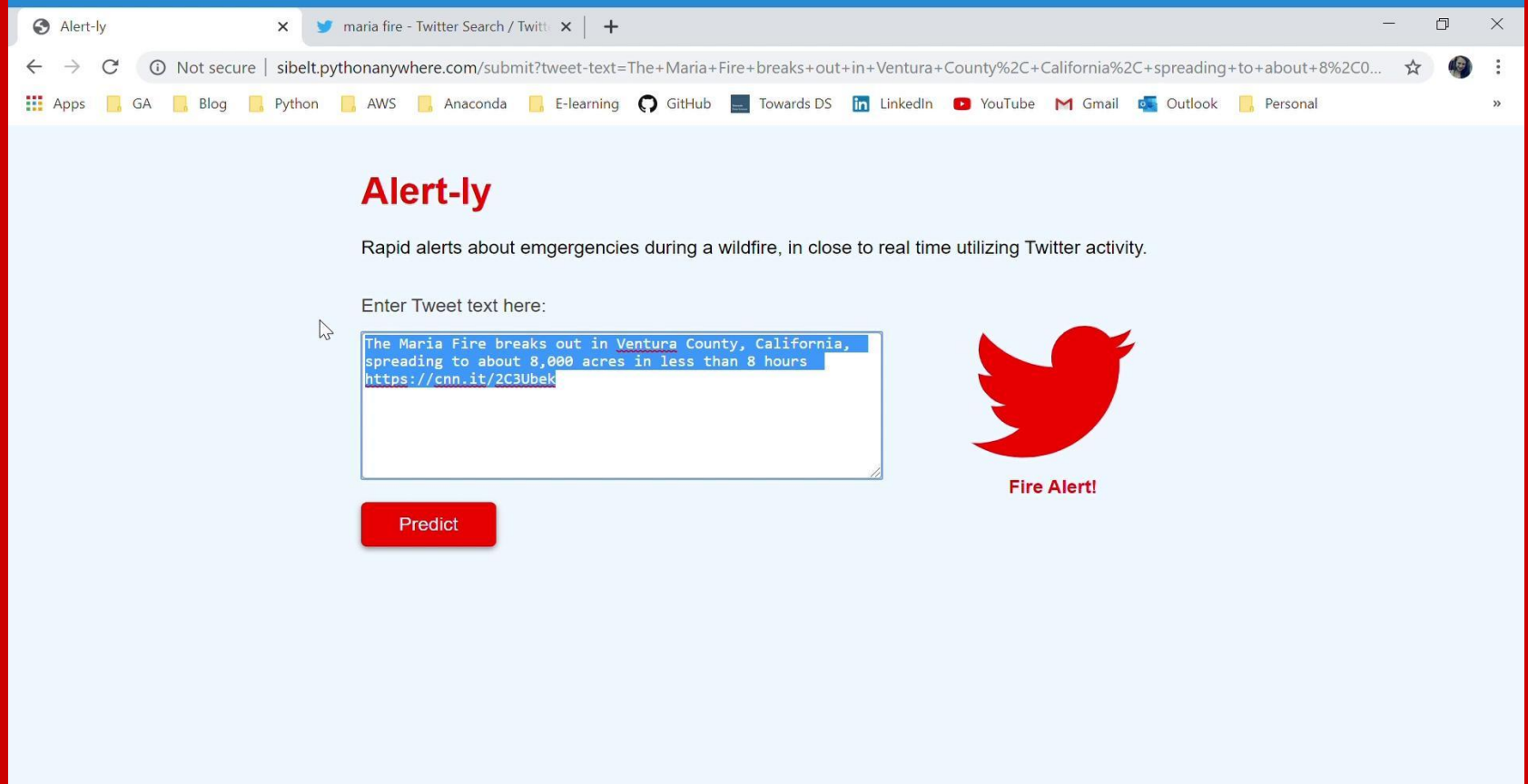
Threshold : 0.5 \rightarrow 0.25
Sensitivity : 0.8728 \rightarrow 0.907

Distribution of P(Outcome = Relevant)



Distribution of P(Outcome = Relevant)



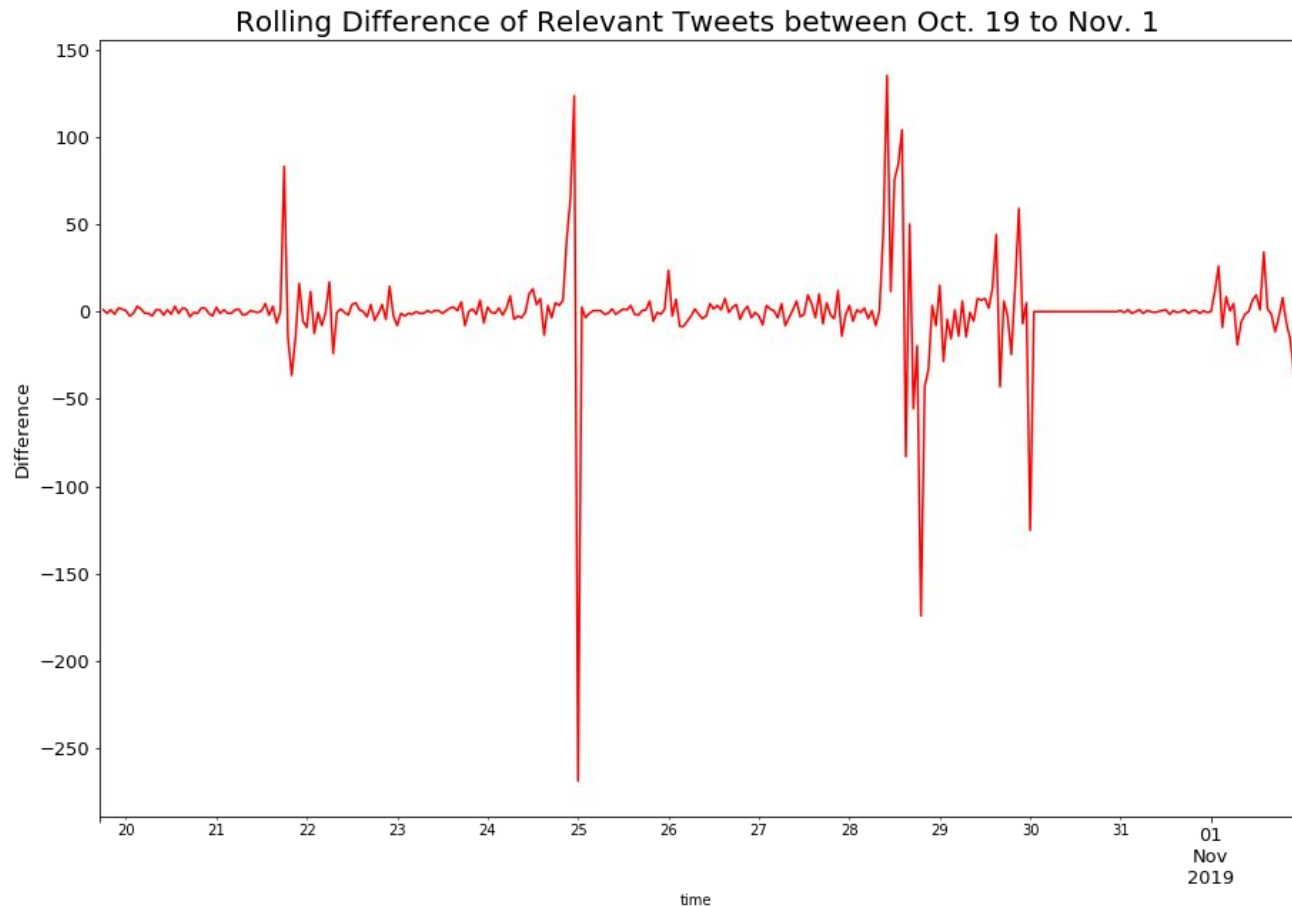


ALERT.LY 1.0 Demo

Model: Pickle | App: Flask | HTML, CSS | Hosting: PythonAnywhere.com

Time Series Analysis

- Difference between “Relevant Tweets” vs “Rolling Avg. of last 2 hours”
- Spikes show significant increase in language related to emergency




Alert.ly 2.0

127.0.0.1:5000

Apps GA Blog Python AWS Anaconda E-learning GitHub Towards DS LinkedIn YouTube Gmail Outlook Personal

Alert.ly 2.0

Rapid alerts about emergencies during a wildfire, in close to real time utilizing Twitter activity.



Get Tweets

No Emergency At 10/31/2019 0:00
No Emergency At 10/30/2019 23:00
No Emergency At 10/30/2019 22:00
No Emergency At 10/30/2019 21:00
No Emergency At 10/30/2019 17:00
No Emergency At 10/30/2019 16:00
No Emergency At 10/30/2019 15:00
No Emergency At 10/30/2019 14:00
No Emergency At 10/30/2019 13:00
No Emergency At 10/30/2019 12:00
No Emergency At 10/30/2019 11:00
No Emergency At 10/30/2019 10:00
No Emergency At 10/30/2019 9:00
No Emergency At 10/30/2019 5:00
No Emergency At 10/30/2019 4:00
No Emergency At 10/30/2019 1:00
No Emergency At 10/30/2019 0:00
No Emergency At 10/29/2019 23:00
No Emergency At 10/29/2019 22:00
ALERT! Fire Emergency At 10/29/2019 21:00
No Emergency At 10/29/2019 20:00

No Emergency At 10/31/2019 0:00

ALERT.LY 2.0 Demo

Data: Time Series Analysis | App: Flask | HTML, CSS, JQuery | Hosting: PythonAnywhere.com

Goal:

- Create an alert system that utilizes social media activity to identify when an alert first occurs



Next Steps:

- Gather more data
- Implement other disasters into our model
- Further improve/optimize our Type II Errors
- Create a model to detect anomalies of emergency related tweets
- Implement a continuous live scrapper to monitor tweets
- Mobile Application for ALERT.LY

ALERT.LY



Questions?

Brianna Lytle - De'Varus May - Sibel Tanoglu

DSI-CC9-LA

ALERT.LY