Sentiment Classifier

Final Presentation

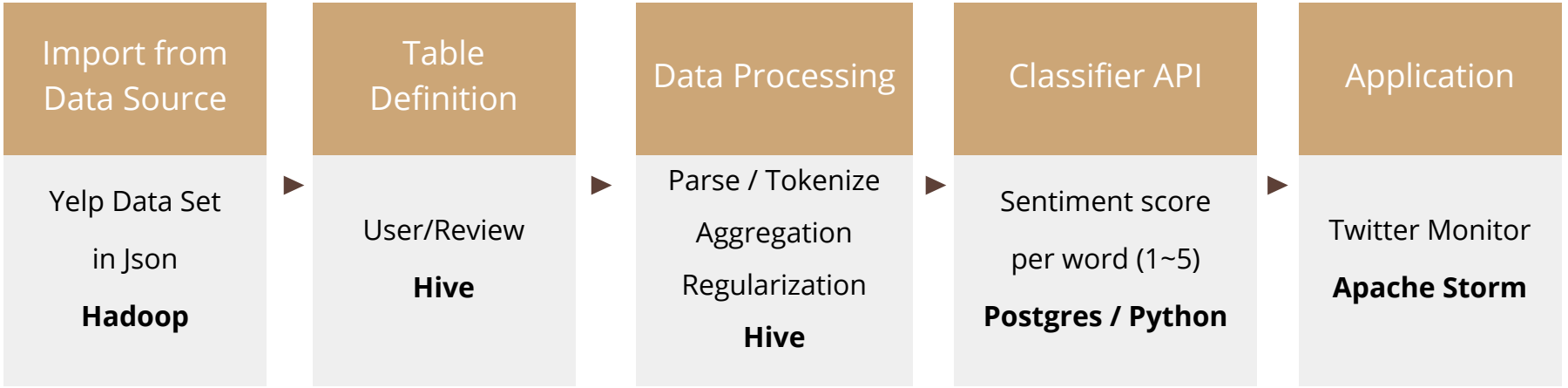
Justine . Michelle . Jae

December 14, 2017

Project Goal:

Use the Yelp Open Dataset to create a sentiment analysis system at scale.

Architecture



Data Processing

4.7m Reviews

user_id	string
text	string
stars	int
...	

1.1m Users

user_id	string
average_stars	int
...	

Clean up
Tokenization
Weighting
Aggregation
Scaling

review_words_avg

words	string
average_stars	double
weighted_stars	double
count	int

Result Dataset Overview

- Total occurrence : ***49,150,667***
- Distinct words : ***633,917***
- Average score : ***3.585***

word	average_stars	count
amazing	4.9048	21187
recommended	4.8892	1526
compliments	4.8748	1313
outstanding	4.8628	1184
best	4.8509	3200
highly	4.8450	16194
exceeded	4.8356	1083
wonderful	4.8210	1615
recommend	4.8180	3550
fantastic	4.8113	3530

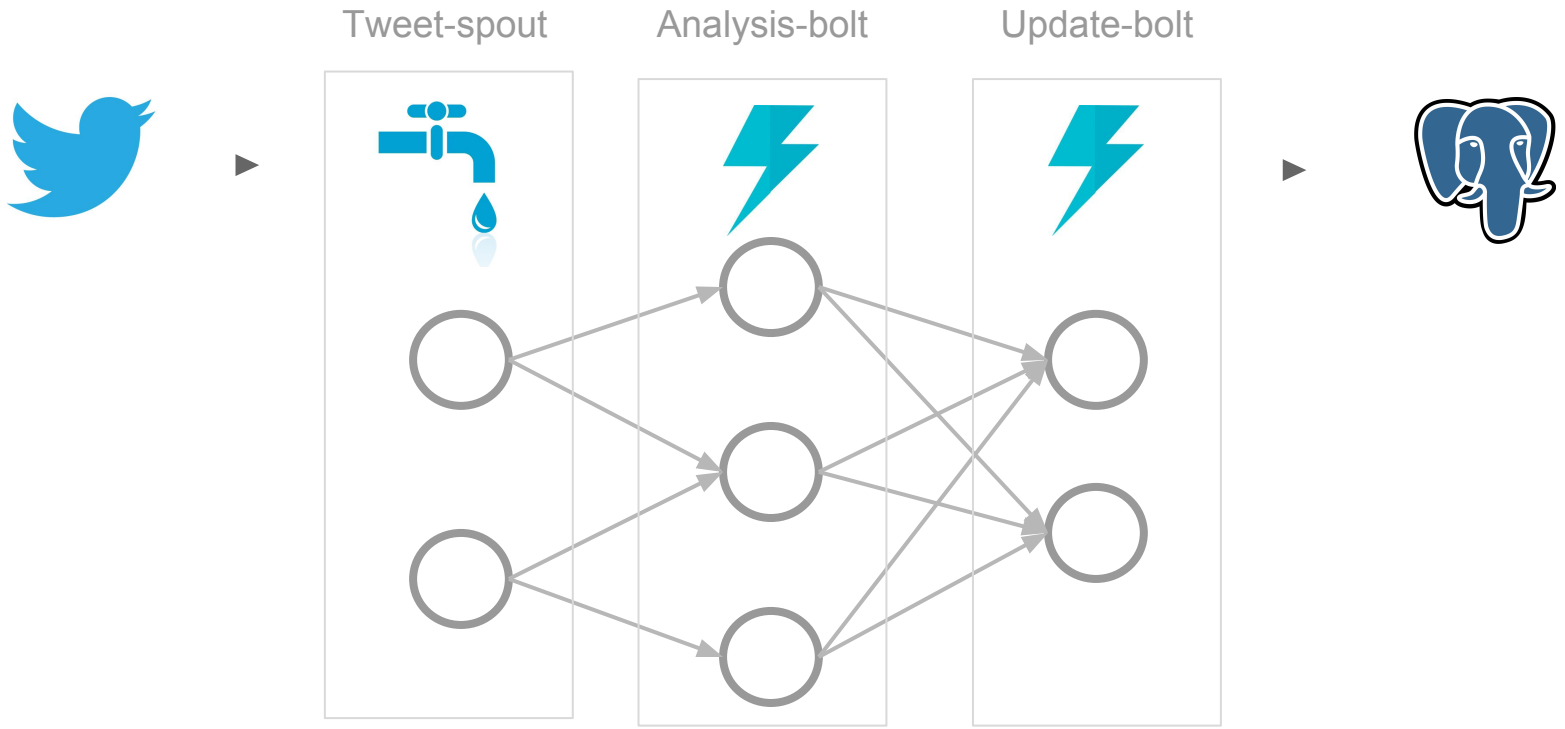
word	average_stars	count
worst	1.0816	6123
horrible	1.0990	4786
terrible	1.1564	3248
rude	1.1808	1106
unprofessional	1.2274	1987
0	1.2326	1625
worst	1.2864	12478
refused	1.3206	2227
disgusting	1.3750	1248
poor	1.3772	1694

Sentiment Classification

- Input: Results table (as csv)
- Split into training and test data
- Trained a logistic regression model
  - Features: word, count
  - Target: avg. rating
- Takes average of predicted sentiment per word in a phrase
- Output:

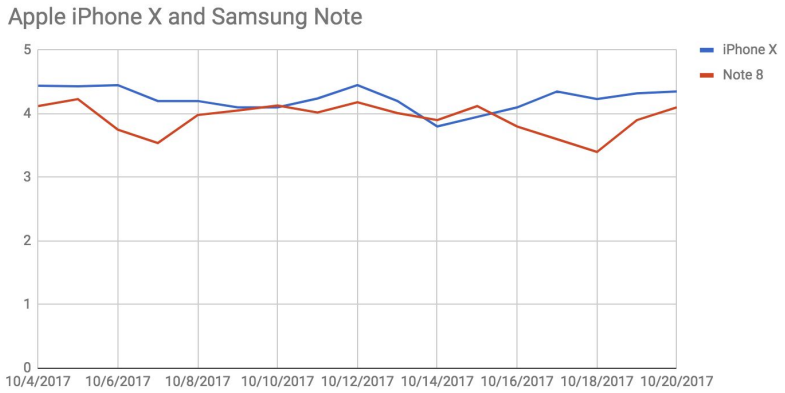
```
$ python yelp_classifier.py “Their cold brew was so good.”

   Sentiment prediction: 4.46
```

Application - Twitter Monitor



Next Steps

- Stress test system for scale
- Further refinements of classifier
- Applications
  - Sentiment trend Visualization
  - News Sentiment Analysis / Stock prediction
  - Personality analysis



Thank you!