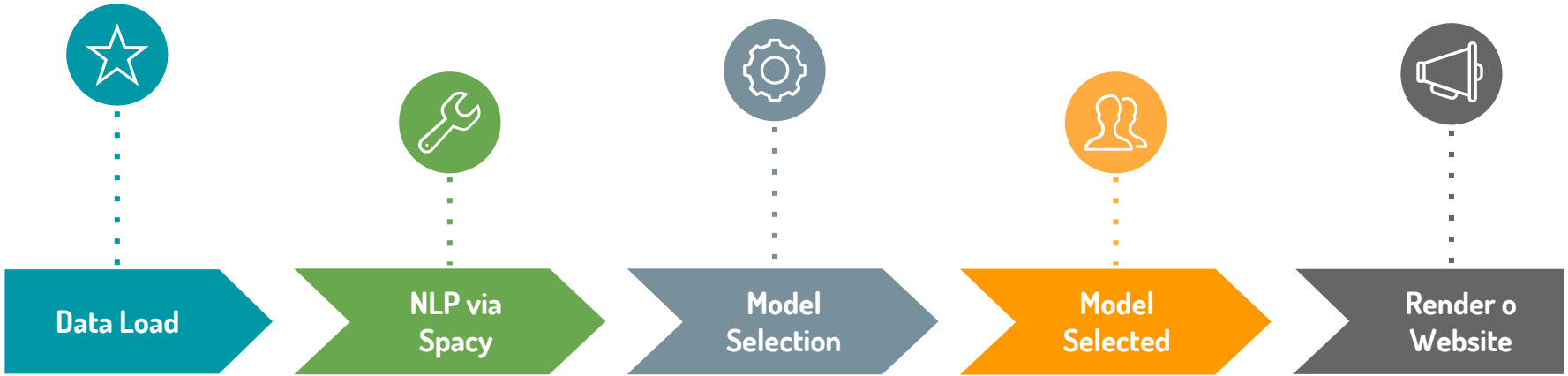
Building a Better Sentiment Detector using Yelp dataset

Ryan Chae

Goal

ANALYSIS

- To create a sentiment analysis tool based on Yelp Reviews
- Create a website that outputs sentiment based on journal entry that the user types



Background

1. Yelp is a crowdsourced review website about local businesses. Each review contains a text and a star rating (1-5).
2. Yelp reviews consist of natural language
3. 1.8 GB file consisting of 2.2M reviews and stars
4. Sentiment can be built via analyzing Yelp review texts and their stars

Text Cleaning

Original:

I was **pleasantly surprised** by the **kalbi** and **purple rice**...it **reminded me** of **my favorite Korean place** when I **lived** in **Hawaii**. **This place used** to have **wraps** and **sandwiches**, but have since **changed menu** and **maybe ownership**. The **waitress** who **serves** me here is always **super nice** and makes me **feel welcome**. When I'm done here I **like** to **cap** it off with **Yogurt Park**...**super convenience**.

After Lemmetization and N-gram:

pleasantly_surprise kalbi purple rice remind_me my_favorite korean place live hawaii this_place use wrap sandwich change menu maybe ownership waitress serve super nice feel welcome like cap yogurt_park super convenience

Modeling

1% model: Bigram over Trigram

Yelp trigram may contain trigrams that are too speific to Yelp (ex. burger_was_delicious)

1% model: CountVectorizer over TF-IDF

CV outperformed TF-IDF on all three models tested: Naive Bayes, NB Boosting, SVM, and Random Forest

1% model: Gridsearch

Performed Gridsearch to optimize parameters

1% model: CountVectorizer Parameter Tuning

Tuned min_df and max_features

100% model: Model Recreation

After text cleaning, recreated CountVectorizer, Naive-Bayes model using 100% of data

100% model: Joblib Pickle

Saved the model as a joblib pickle file for production

Result

Algorithm	Accuracy	F1
Naive Bayse on CV	0.66 (+/- 0.04)	0.66 (+/- 0.04)
Naive Bayse on TF-IDF	0.60 (+/- 0.00)	0.60 (+/- 0.00)
SVM on CV	0.68 (+/- 0.02)	0.68 (+/- 0.02)
RF on CV	0.66 (+/- 0.01)	0.66 (+/- 0.02)
NB Boosting on CV	0.61 (+/- 0.11)	0.61 (+/- 0.11)
NB Boosting on TF-IDF	0.60 (+/- 0.00)	0.60 (+/- 0.00)
NBCV min_df=.1	0.63 (+/- 0.01)	0.63 (+/- 0.01)
NBCV max_features=10000	0.63 (+/- 0.05)	0.63 (+/- 0.05)

NB: Good result with much, much faster computation

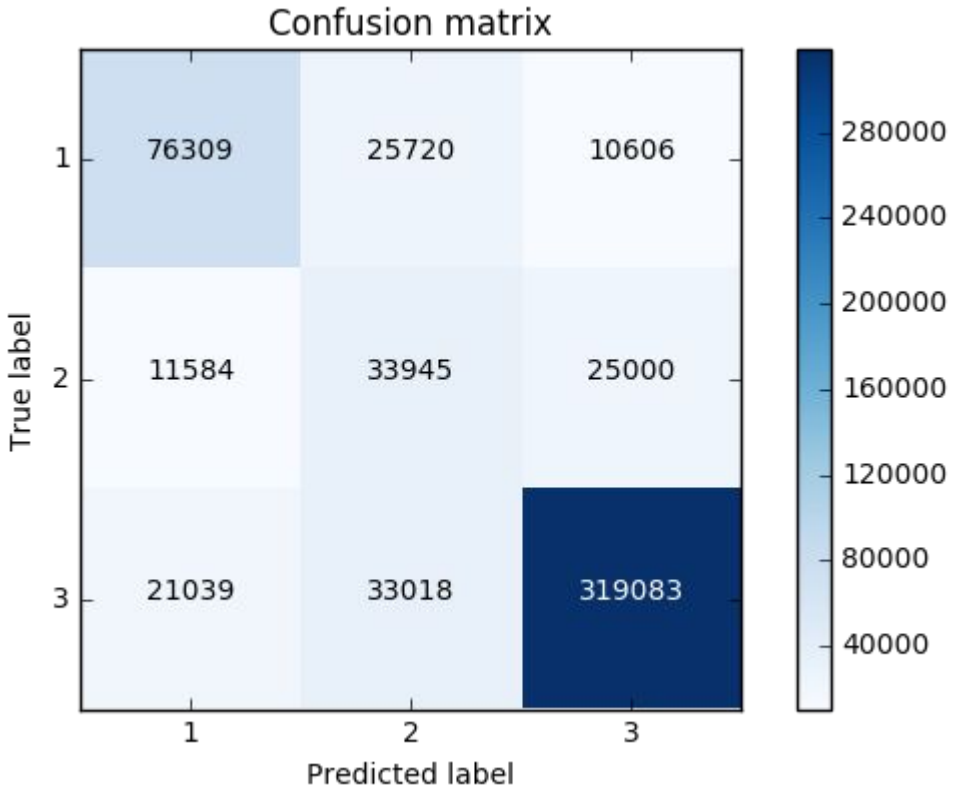
Gridsearch best parameters:
NB: {'alpha': 1, 'fit_prior': True}
SVM: {'C': 0.1, 'class_weight': None}
RF: {'min_samples_split': 2, 'n_estimators': 100, 'bootstrap': False, 'criterion': 'gini', 'max_depth': None}

Model Fit

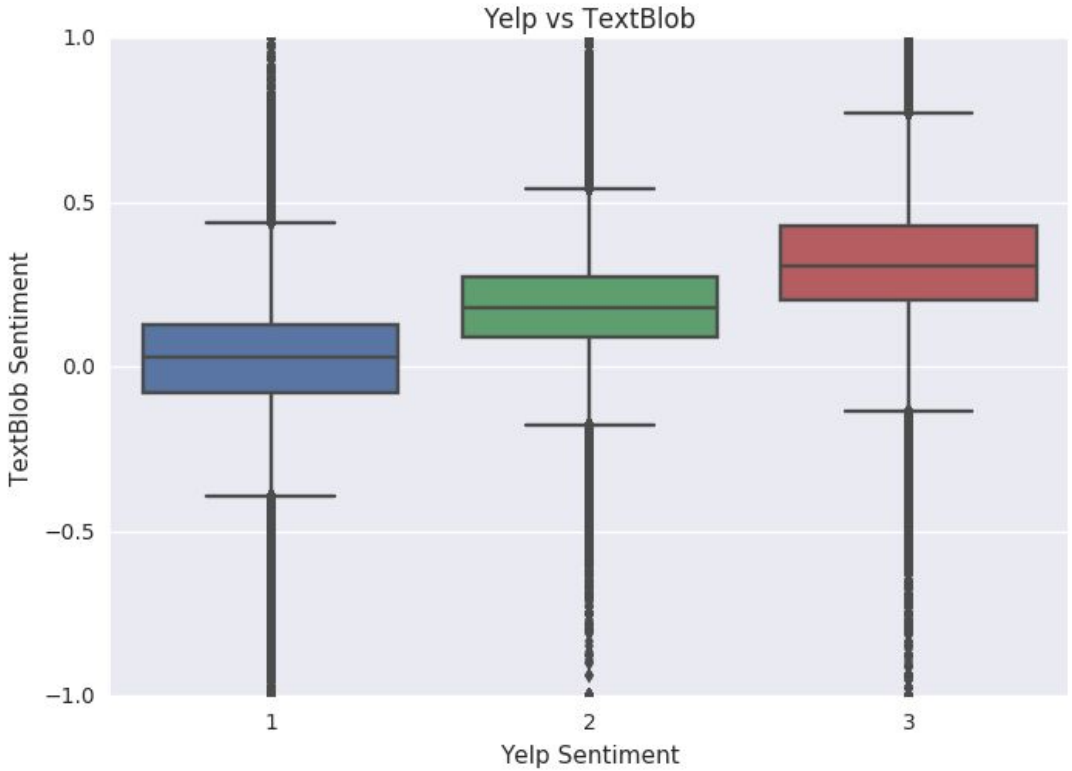
Accuracy: 0.76 (+/- 0.04)
F1: 0.76 (+/- 0.04)

	precision	recall	f1-score	support
1	0.70	0.68	0.69	112635
2	0.37	0.48	0.42	70529
3	0.90	0.86	0.88	373140
avg / total	0.79	0.77	0.78	556304

baseline
3 0.670748
2 0.126781
1 0.202471



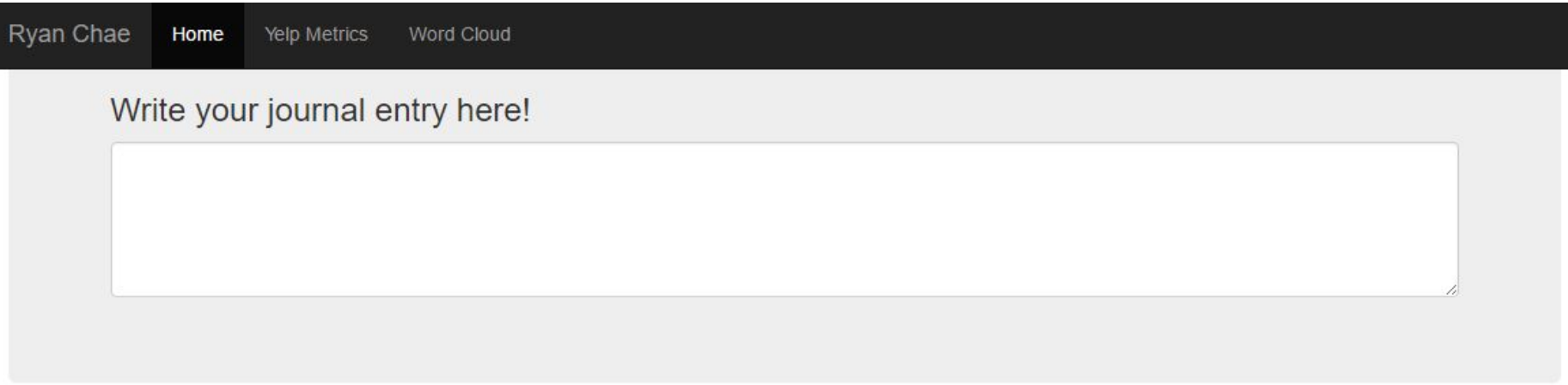
Yelp vs TextBlob Sentiment



Correlation

0.55

Webapp



Backend
Joblib for CV-NB Pipeline storage
Flask for Server

Frontend
Restless for API
AngularJS for API Rendering

Conclusion

Correlation
The correlation between Yelp sentiment analysis and Textblob sentiment analysis was 0.55

Model
On 1% data, Naive-Bayes on CountVectorizer had F1-score of 0.66. While slightly lower than F1-scores for SVM or Random Forest, it performed much faster

1% vs 100% Data

100% Data had F1-score of 076 compared to 0.66 in 1% data. This may be due to extra words in CountVectorizer (26,659 vs .

To Do
Track journal results to see whether Yelp sentiment analyzer is a good indicator of sentiment