Sentiment Analysis Model Comparison: The Case of Tesla

Group A :
Belen Gutierrez, Binpeng Xiu, Marc Castillo, Siqi Zhao, Zehui Wang, Or'el Anbar

Contents

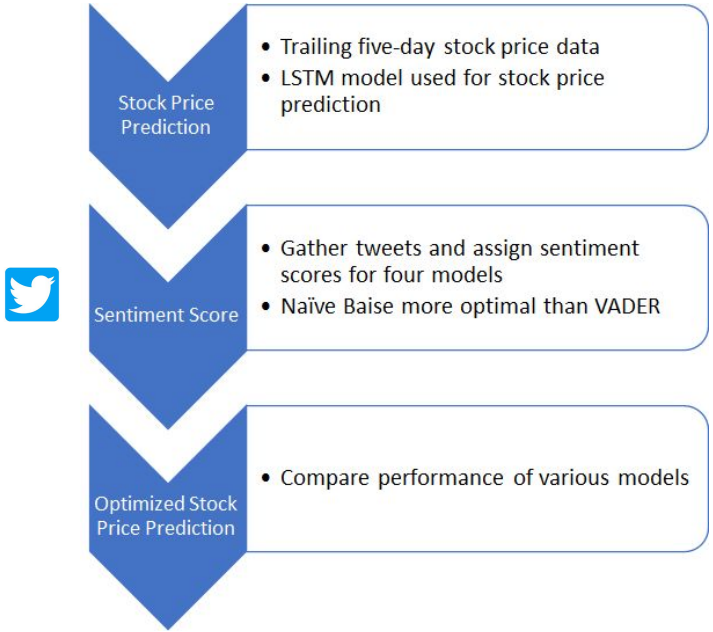
Reference Article: TANG, WEI, YANG, CHEN

Article Title: Sentiment Analysis of Company-Related Tweets and their Prediction Power

Project Goal: “[T]o incorporate sentiment analysis result of tweets related to specific companies into the prediction for the company’s performance, i.e. their stock prices, and see how much weight user feedbacks on twitter have in the model.”

Methodology Overview: TANG, WEI, YANG, CHEN

1. Create a baseline by using a conventional stock price prediction model (LSTM)
2. Develop a sentiment score based on tweets about the company that is the subject of the analysis for four models
    a. model with only daily score, model with only weekly score, model with tweet count and daily score, and model with tweet count and weekly score
3. Compare sentiment analysis with conventional stock price prediction to determine accuracy applicability



Outline of Our Methodology

Goal: Predict Tesla stock price using twitter sentiments

Phase 1: Data - sentiment140

- Train our own classifier using Naive Baye and VADER for tweets

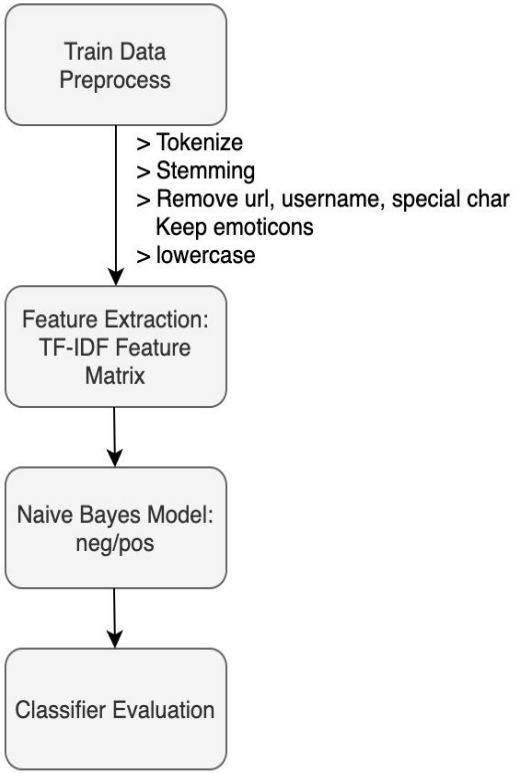
Phase 2: Data - tweets, & stock data

Phase 3: Stock price prediction using above two sets of senti scores: NB & VADER

Phase 1: Train Naive Bayes Classifier

Training Set: Sentiment140

- Provided by Stanford University students
  http://help.sentiment140.com/for-students
- From April to June in 2009
- 128M raw twitter texts across multiple fields with sentiment label(0 for negative and 1 for positive)



Phase 1:

Dataset Pre-processing:

- Remove special characters, replace url, username with placeholder
- Keep emoticons
- Lowercase, Tokenize, Stemming

0	__userhandl ahaha honestly? not even tired, the supris woke soo much. good luck school tomorr…
1	wtf?? goldfish hate me, they keep die poor
2	now serious.. have finish some thing
3	__userhandl come think it..i had one, and have one. but but.. tanga ako

TF-IDF Vectorizer Matrix

Naive Bayes Modeling

Phase 1: Model Evaluation

Validation Set: 20% Hold-out Set

- Accuracy = 80%
- Precision = TP/(TP+FP)
- Recall = TP/(TP+FN)

Test Set: Manually Labeled

- Accuracy = 82%
- Precision = TP/(TP+FP)
- Recall = TP/(TP+FN)

```
In [64]: print(metrics.classification_report(y_test, pred))
              precision    recall  f1-score   support

           0       0.79      0.80      0.80    159789
           1       0.80      0.79      0.80    160211

    accuracy                           0.80    320000
   macro avg       0.80      0.80      0.80    320000
weighted avg       0.80      0.80      0.80    320000
```

```
In [18]: print(classification_report(y_val, y_val_pred,
digits=4))
              precision    recall  f1-score   support

           0     0.7897    0.8701    0.8280       177
           1     0.8598    0.7747    0.8150       182

    accuracy                         0.8217       359
   macro avg     0.8247    0.8224    0.8215       359
weighted avg     0.8252    0.8217    0.8214       359
```

Custom Dataset for Sentiment Extraction

Connecting to official Twitter API via tweepy

- Couldn't directly search further back than a week with official search endpoint
- Solution: snscrape to webscrape tweet URLS

```
# below constructs list of urls to tweets satistfying query; requires snscrape
# takes a long time to run so comment out after running once
cmd = "snscrape twitter-search \"#Tesla since:2020-01-01 until:2020-10-31\" > tesla.txt"
os.system(cmd)
```

- Then could connect to different endpoint in official API to fetch tweets from URL
- Collected tweets from January to October; sample size 345,605 after dropping missing values

Preprocessing

- Dealt with special characters just as before
- Used the TfIdfVectorizer to transform this into sparse matrix

Plan for Custom Dataset

Extract Sentiment

- Via own model and established ones out there

Merge with Financial data

- Easy to download from Yahoo Finance - Tesla & NASDAQ
- Then have to aggregate by Date to get a “composite” sentiment score per day

Ended up with a date-level dataset that spanned 194 trading days

Phase 2: Sentiment Extraction

Method1: VADER

We use vader_lexicon in NLTK to execute the sentiment analysis, get the polarity for the Tweets of each day, which is how much positive, negative, neutral the Tweets are.

Date	text	positive	negative	neutral
2020-01-01	To paraphrase 6 reasons by a Tesla fanboi to NOT b	0.147	0.057	0.796
2020-01-02	Get a free stock. Claim a stock now without invest	0.117	0.051	0.831
2020-01-03	I should become a Tesla analyst and also just pull	0.098	0.064	0.838
2020-01-04	What do the iPad, Tesla, and XboxOne have in commo	0.108	0.06	0.832
2020-01-05	Días de niebla en valladolid . Mangueras de carga	0.129	0.053	0.818
2020-01-06	The one with the software that actually works?	0.097	0.051	0.852

Phase 2: Sentiment Extraction

Method2: Naive Bayes

Index	Date	text	sentiment
0	2020-01-01	To paraphrase 6 reasons by a Tesla fanboi to NOT b	0
1	2020-01-01	Can Tesla hold its charge? Elon Musk's electric ca	0
2	2020-01-01	Tesla Inc. to face lawsuit over alleged racism tow	0
3	2020-01-01	Batteries Now Obsolete? The "Tesla Killer" Is Here	0
4	2020-01-01	Happy 2020, Elon! You ROCK! Thank you for inspirin	1
5	2020-01-01	Get a free stock. Claim a stock now without invest	0
6	2020-01-01	10 Electric Vehicles to Watch – So 2019 was, final	1

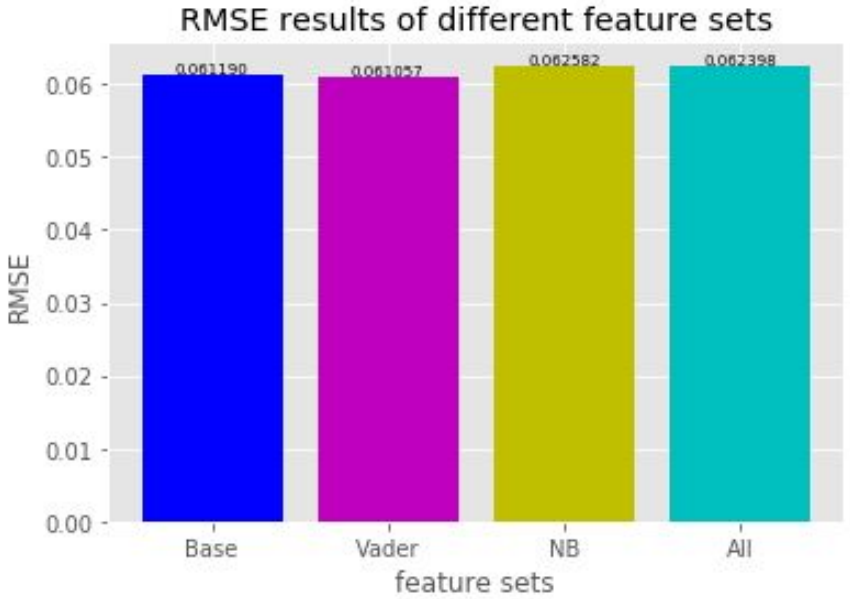
Date	neg_per
2020-01-01	0.29783
2020-01-02	0.247849
2020-01-03	0.31106
2020-01-04	0.287129
2020-01-05	0.238832
2020-01-06	0.228693
2020-01-07	0.185833
2020-01-08	0.203154
2020-01-09	0.226695
2020-01-10	0.2213
2020-01-11	0.252555

Phase 3: Stock Return Prediction

- Models:
  - Linear Model: Ridge Regression
  - Nonlinear Model: Random Forest
- Performance Measures
  - MSE - Mean Squared Error

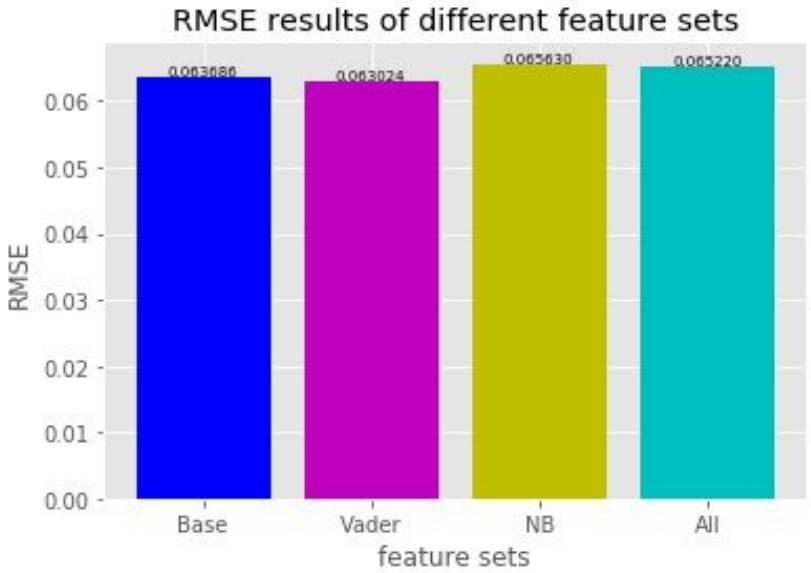
Linear Model - Ridge Regression

- Four regressions:
  - **Base:** pre day Nasdaq/Volume
  - **VADER:** VADER score+base
  - **NB:** Naive Bayes scores+base
  - **All:** Vader+NB+base
- Conclusion:
  - VADER+
  - NB -



Nonlinear Model - Random Forest

- Sentiment Extraction Method- VADER
- Both our linear model (Ridge) & non-linear model (Random Forest) confirm that using VADER only as the sentiment extraction give the best MSE score
- Ridge VADER (MSE: 0.003728) VS. Random Forest VADER (MSE: 0.004070)
- **Conclusion** - Ridge Regression using VADER as the Sentiment Extraction methods does the prediction best



Best/Least Favourite Part

- Learning Python modules and libraries by researching a topic we were interested in
- Interacting with classmates to discuss methodology and learn from one another
- Applying NLP to finance industry matters in line with latest industry trends

- Delays in Twitter API approval
- Communication wasn't always easy
- Small/ outdated training data size

Conclusions

Summary of findings

- Minor performance boost using the VADER polarities but not with our own classifier
- One limitation of our own classifier was training set labels only contained 2 classes

Next steps/Points of improvement

- Can tinker with other preprocessing choices: changing tfidf properties, PCA
- Try other classifiers on sentiment140 dataset

References

- Reference Article: http://comp562fall18.web.unc.edu/files/2018/12/PP2.pdf
- VADER Documentation: http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf
- Pulling from twitter guide: https://medium.com/@jcldinco/downloading-historical-tweets-using-tweet-ids-via-snscrape-and-tweepy-5f4ecbf19032