An NLP-based Approach to Improving Prediction of Twitter Follower Count

by Brandon Carter and Brandon Axe

Introduction

- Twitter corpus presents variety of interesting ML and NLP problems due to size (over 200 million users) and lack of syntactic structure, spelling, grammar, etc. (which follow from 140 char tweet limit)
- NLP approaches need to be modified to accommodate the Twitter corpus
- Our goal was to combine ML and NLP techniques to predict the number of followers for a Twitter user based on a given tweet
- Number of followers is a good metric because it is an easily-observable indicator of a user's influence in the Twitter social network

Methods: Twitter Dataset

- Tweet Scraping from Twitter API (scraped ~5 million tweets in Nov 2015)
- Extract basic features about user for each tweet
- Example tweets
  - Want to work at OfficeTeam? We're **#hiring** in **#MERCER**, PA! Click for details: **https://t.co/HiQUTX8i7X #Clerical #OfficeTeam #Job #Jobs**
  - When **@EricaKacprowicz** mom gets us a cookbook cause all we eat is pasta **#thankskim**
- Tweet Text Tokenization
  - Use published Twitter tokenizer from CMU
  - Remove case-sensitivity and remove NLTK stopwords
  - Replace any links with ‘URL’ string
- POS Tagging
  - Use published Twitter POS tagging methods (use tool by CMU)
- Used ~50k tweets for training models and ~20k tweets for testing

Methods: Feature Vectors

We used the following data points to generate the feature vectors for each tweet

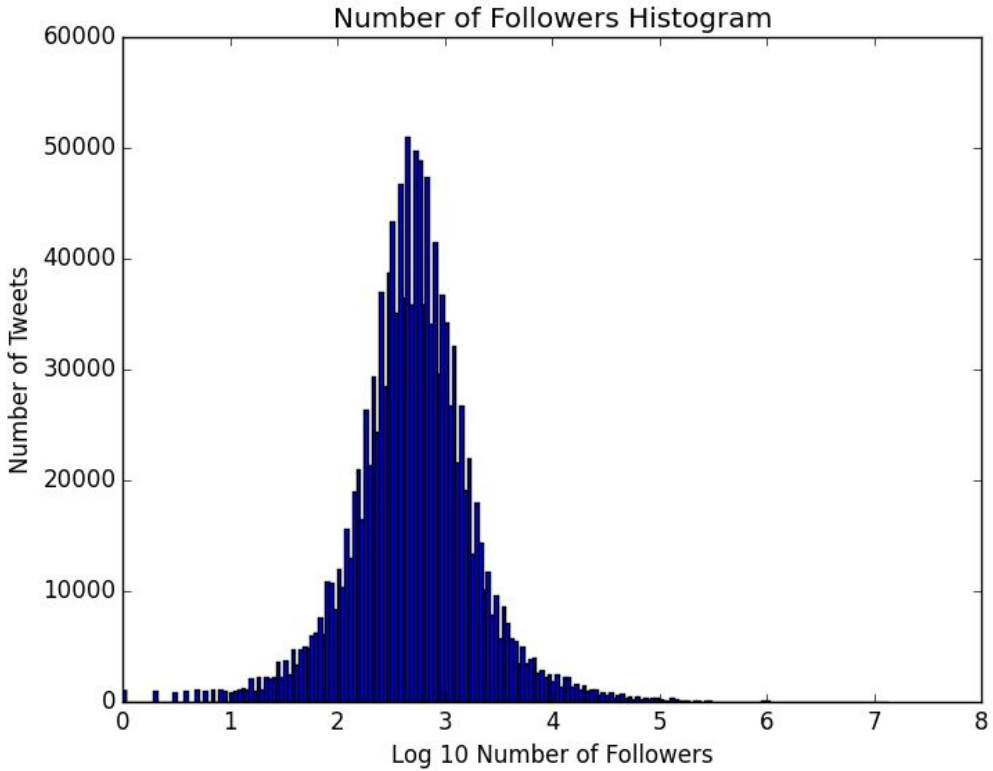
- Word Embeddings via word2vec
  - We trained a word2vec model on all of the tweets in the tokenized tweets file, and used the average of the word2vec mappings for each of a tweet's tokens in the tweet's feature vector.
  - The dimensionality of the word2vec model used was 100.
- NLP Features (based on the tweet)
  - Length of tweet, # of hashtags, # of URLs, # of emojis, # of mentions in the tweet
- ML Features (based on the user)
  - Account age, # of tweets made, # of tweets favorited by others, if the account is "verified"
- Part-of-speech tags
  - The frequency for each part of speech in the tweet

Methods: Regression Algorithms

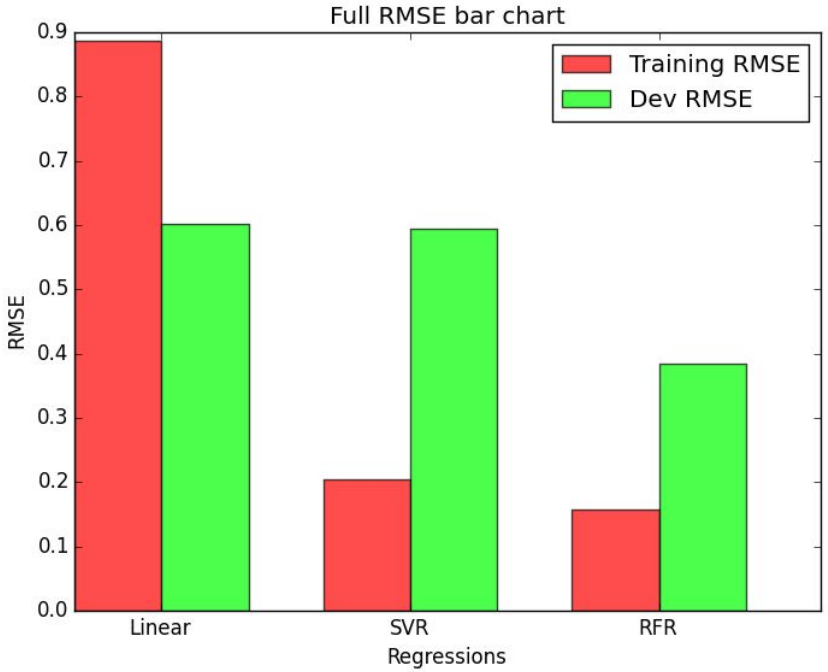
Each of the regression algorithms used were implemented using sklearn

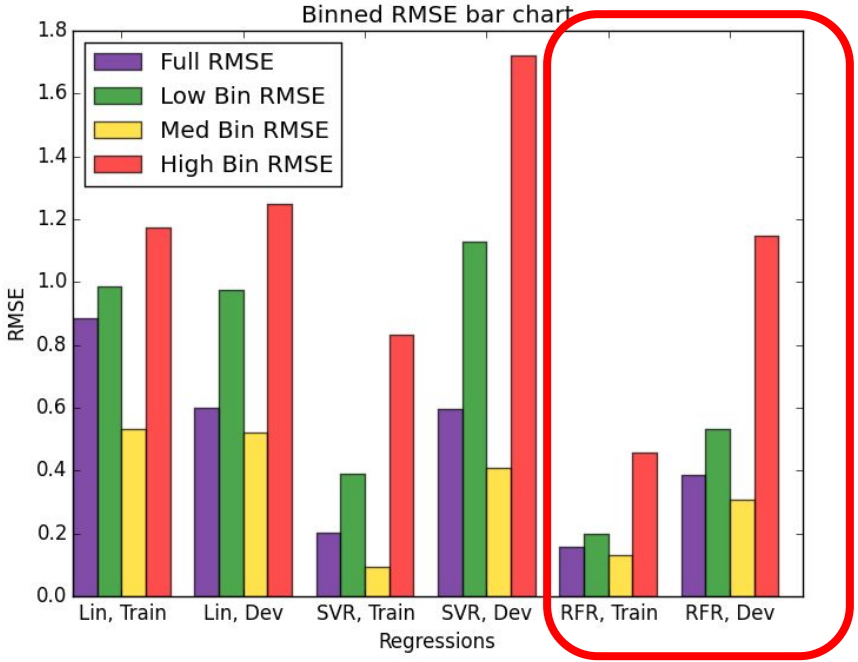
- Linear Regression
  - Quicker compared to other models because of a multi-core implementation
  - Less likely to be accurate, seeing as there is likely no perfectly linear relation
- SVR (Support Vector Regression)
  - Allows for non-linear relationship between the feature vectors and the regression output
  - Variety of kernel choices. We found the best results using the radial basis function (RBF) as our kernel
  - Much slower than linear regression for both training and prediction
- Random Forest Regression
  - Quicker compared to other models because of a multi-core implementation
  - Uses an ensemble classifier, which allows for
- Logistic Regression
  - Doing low/medium/high bin classification is a simpler problem, but also does not specify the relationship between each of the classifications

Follower Count Distribution for Data Set

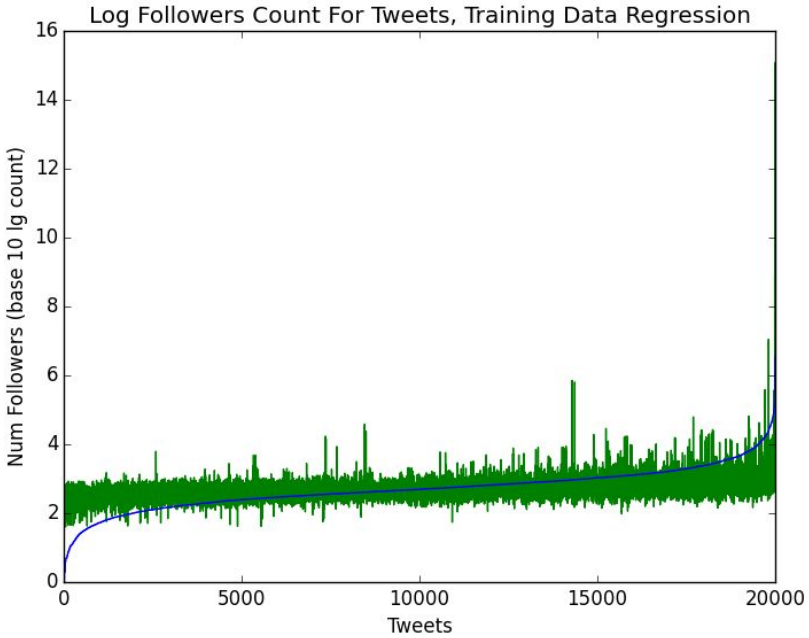


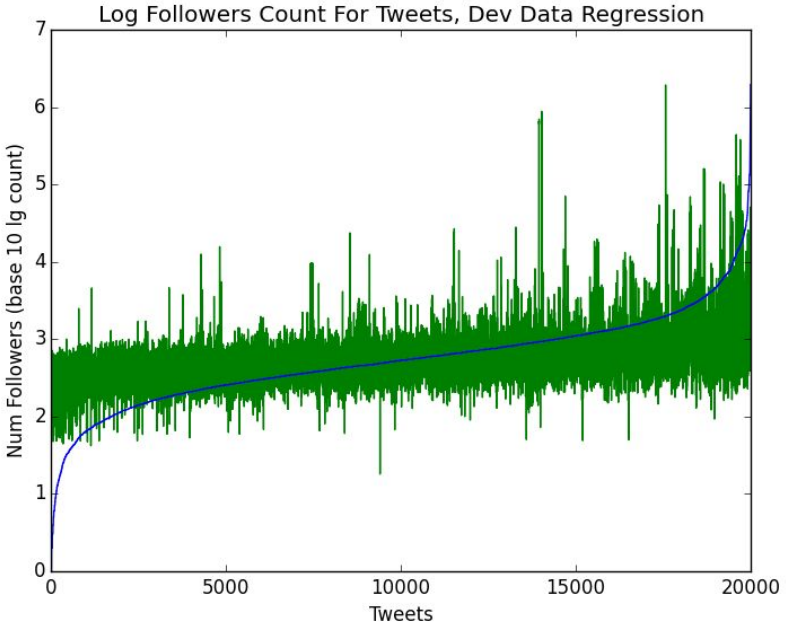
Results - RMSE vs Regression Alg.



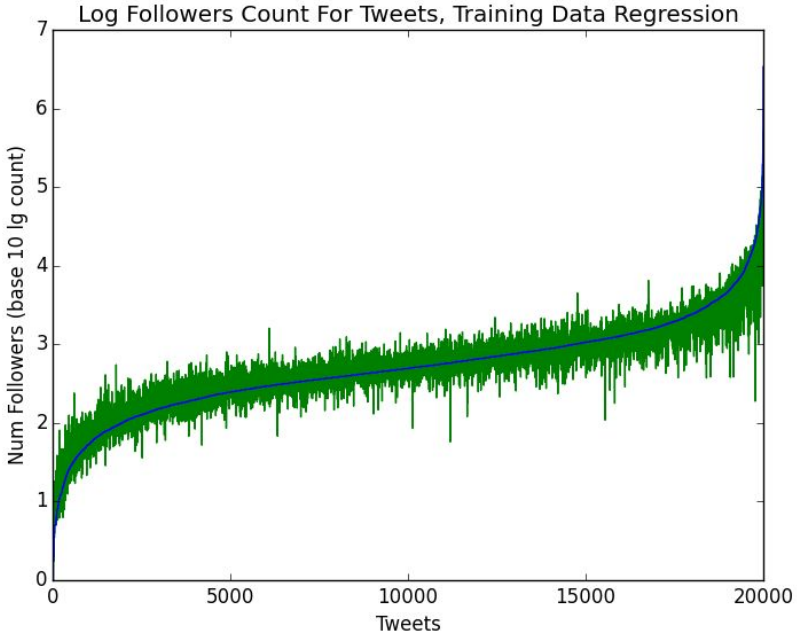


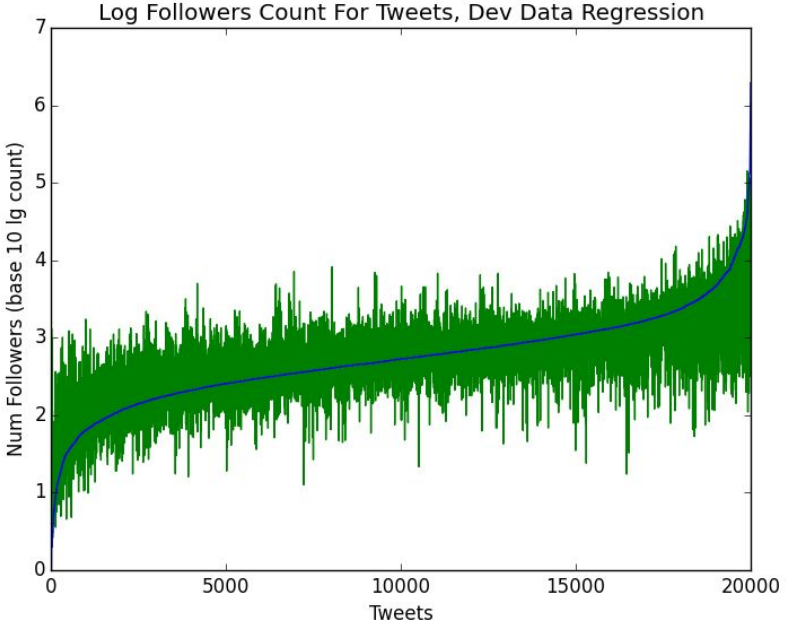
Actual vs Predicted Number of Followers per Tweet Linear Regression



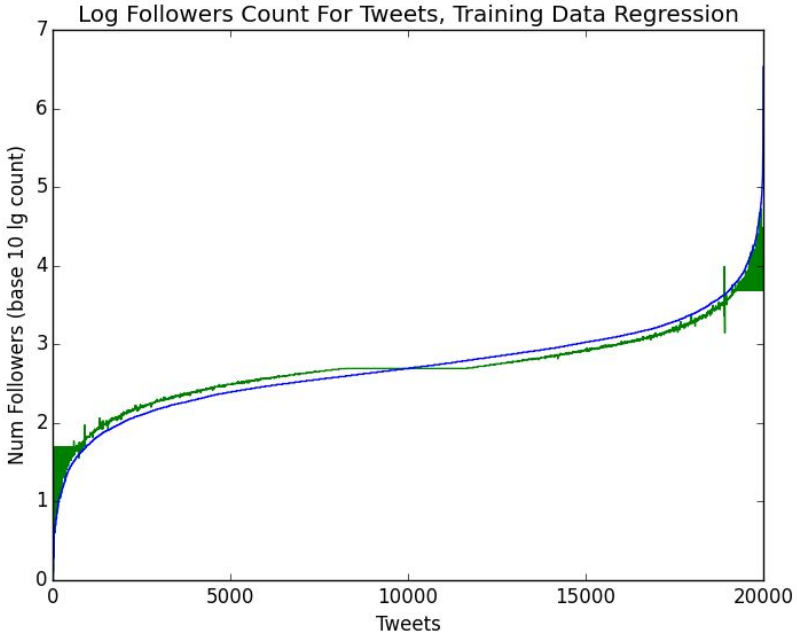


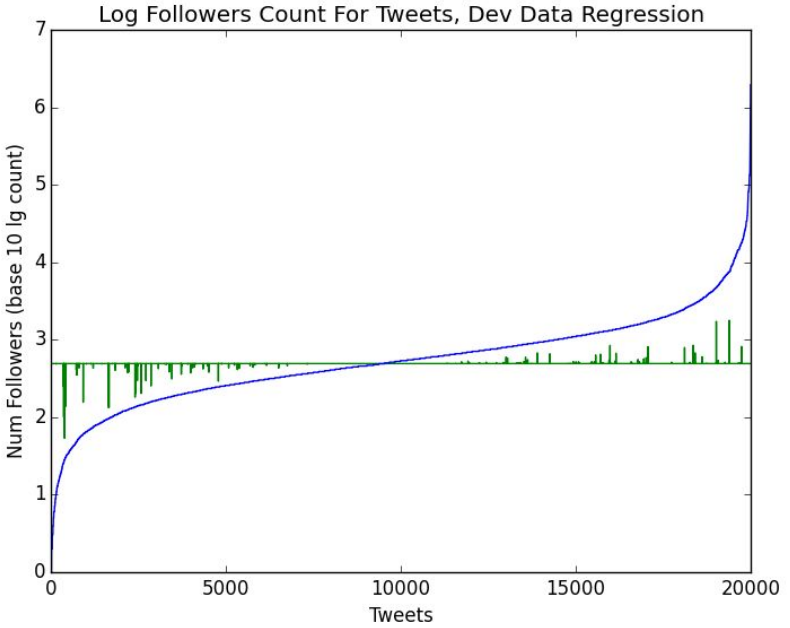
Actual vs Predicted Number of Followers per Tweet
Random Forest Regression



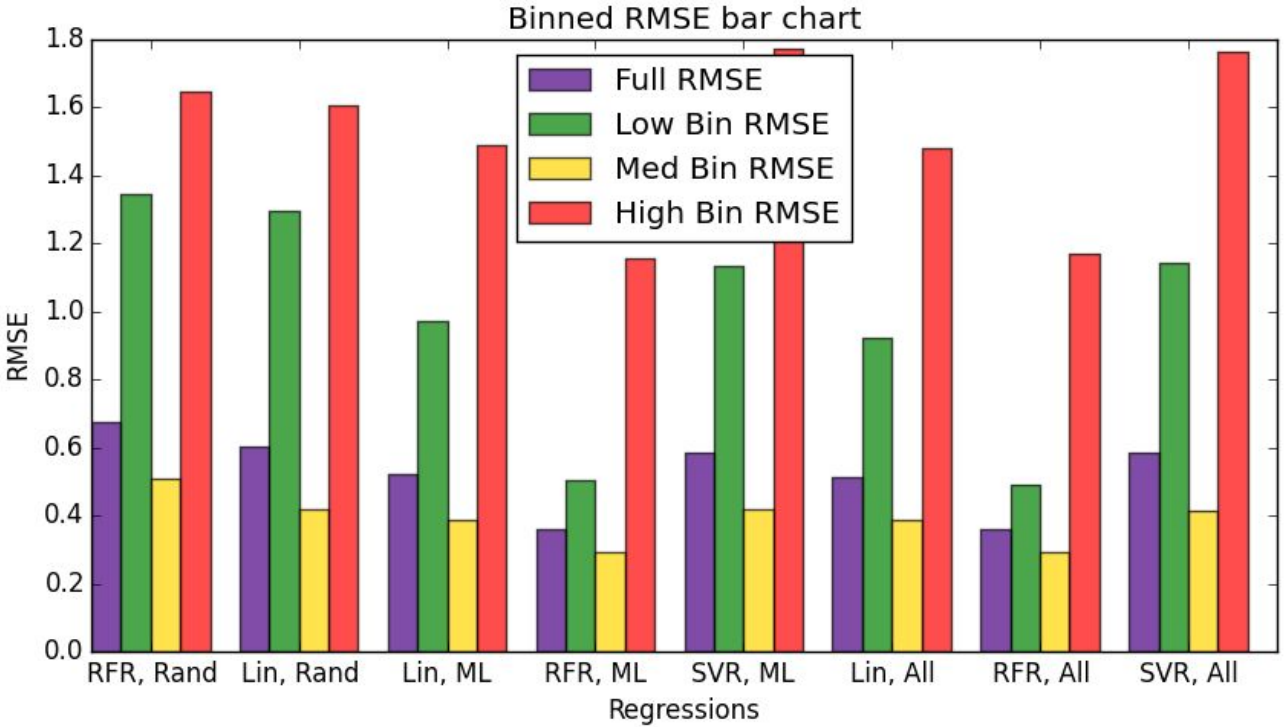


Actual vs Predicted Number of Followers per Tweet Support Vector Regression (RBF Kernel)

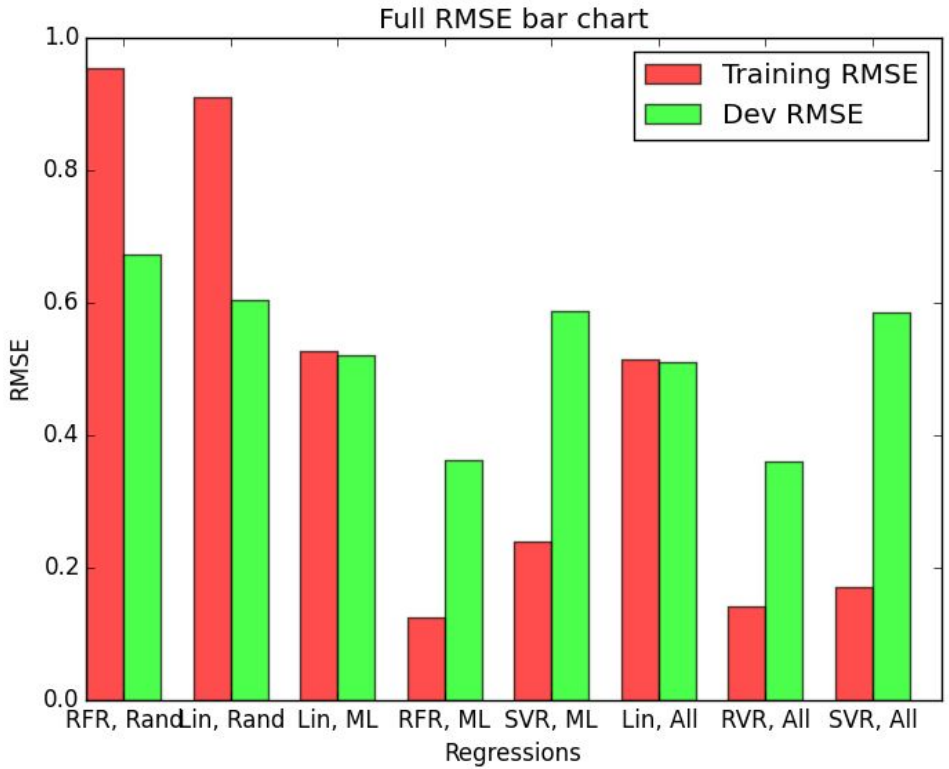




Binned RMSE vs Regression for Dev Data



RMSE vs Regression Algorithm



Conclusions/Future Work

- Able to use textual and semantic features to greatly improve the randomized predictions!
- Small improvement using NLP features over just ML features
- Future Work:
  - Further hyperparameter tuning for regressions
  - Engineer more NLP features to make better of mentions, hashtags, and URLs
  - Try using a neural net to solve this problem
  - Currently English-only, evaluate on other languages?