Sentiment Analysis of Tweets

Definition

Project Overview

In the digital world, we have become accustomed to sharing our sentiments across mediums such as Facebook, Instagram, and Twitter, to anyone that is listening. It's a mass expressing of the pain, hope, joy, and love that can both inspire change or commence destruction. As humans we crave the information, we are pack animals by nature and both consciously and unconsciously alter our decision based on the mood of the herd. As a result, social media has become a very powerful marketing tool, businesses bombard the platforms with advertisements with the hopes of their own finical gain. The goal of this project is to remove this *muddying of the waters* from a thread of Twitter messages.

Problem Statement, Evaluation Metrics, and Benchmarks

Problem Statement

Using a combination of supervised and unsupervised algorithms we hope to analyze the sentiment of any given Twitter message on a scale of Genuine Expression to Finically Motivated. The final result will be a binary classification of any message as Genuine or Motivated.

Evaluation Metrics

To evaluate our solution will we use the *score* method from our supervised learning algorithm on a validation set is taken from a conglomerate of unique messages.

Benchmarks

Because our solution is binary (Genuine or Motivated) our benchmark will be random guessing. If labels were assigned at random the program should score about 50%.

Analysis

Dataset

The dataset for this project is the *challenge_en.json* consisting of ~8,000 twitter messages posted over the course of 15 minutes. Each post consisted of both the user-created text as well as background information about its origin and interactions within the twitter world (retweets). Below is an example of one entry taken from the JSON file.

Figure 1 - Attributes of Twitter Message

The binary classification of Genuine Expression or Financially Motivated was then later added to the dataset on about 700 entries by the engineer. However, it should be noted that some of these labels were placed on repeated messages and do not appear in later training processes.

Algorithms and Techniques

Bayesian Optimization

Bayesian optimization is a derivative-free optimization method that resembles human-tuning selection. Within this, a Gaussian Process and Acquisition Function are combined to create a regression of possible

function scores and expected quantities of improvements. Credits for the inspiration and implementation of this technique need to be given to Yurii Shevchuk and his blog NeuPy (Shevchuk, 2016). The figure below provides an example of the Gaussian process' prediction of the optimum value to test next based on previous results. For every iteration past values were entered into the Bayesian optimization function, fitted to a Gaussian curve, plotted, and used to select the next 'guess' of the program. In this specific case, the optimum number of outputs for the first convolutional network was being determined based on its score (accuracy minus training time). In the case of this project Bayesian Optimization was used for hyperparameter selection on the supervised learning algorithms.

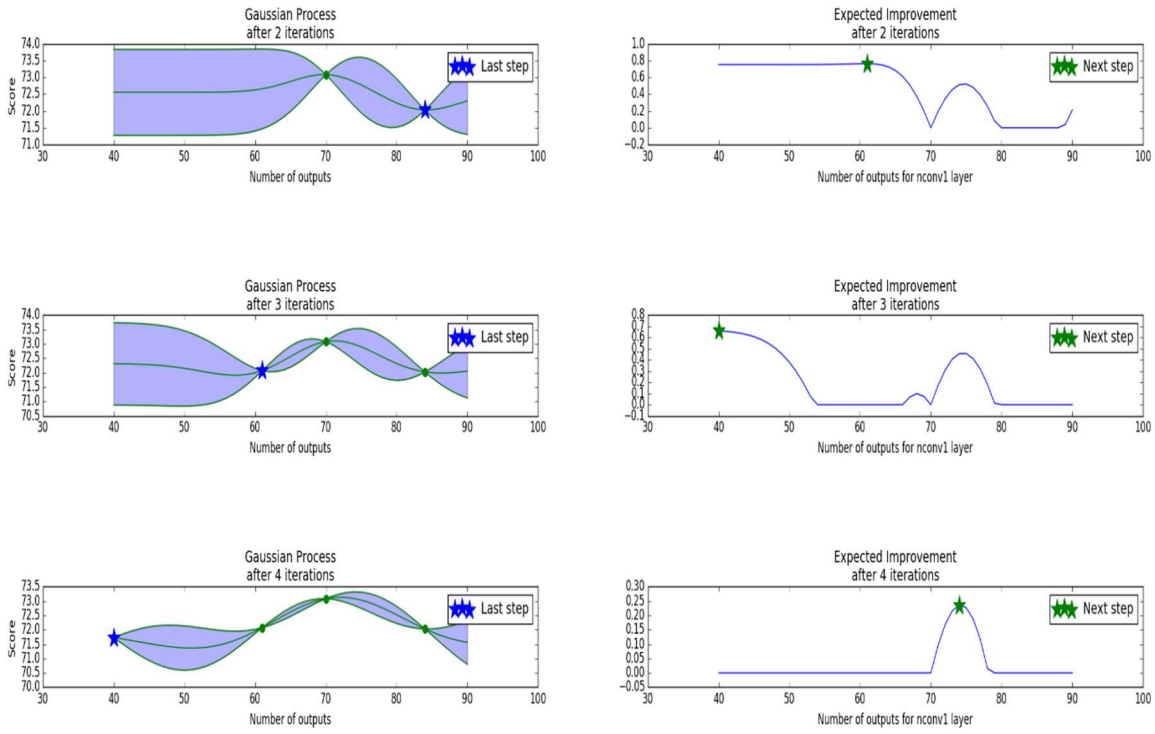


Figure 2 - Bayesian Optimization of Outputs

Multi-gram Vocabulary Building

A gram is a group of words when used together have a different meaning than their individual components. For example, given two sentences:

"I met Jane yesterday."

"I met with Jane yesterday"

In the second example the combination of words "met with" expresses a different sentiment (to have a date or meeting) than the individual words would imply (met = to be introduced, with = alongside someone or something). By building a vocabulary that includes entries like "met" and grams such as "met with" we hope to extract a little more information from the text.

Polarizing Words Vocabulary Pruning

Vocabulary pruning is the concept of removing entries that don't provide any significant value. Pronouns, prepositions, and common verbs such as 'he', 'and', 'the', 'get' appear with relatively similar frequencies regardless of classification and therefore only add noise to the feature set.

In this project, entries were ranked based on a ratio of how many times they appeared in each of the two classifications. Only entries on the polar extremes were retained in the vocabulary.

Kmeans

Kmeans is an unsupervised learning algorithm that groups objects into clusters based on similar features. In this case, it was used as a filter for repeated messages. After grouping like-messages, the standard deviation of the entry-to-cluster-center distances was normalized by the number of entries within the cluster. Thus, giving a 'cluster standard deviation per entry' value for each cluster. If this value was extremely small it implied that the messages were the same or nearly the same containing only a small variation like a different link address for example. These messages were considered spam and removed. The worry was that if they were to remain they could skew the vocabulary by adding an unproportionate number of counts to the specific words in their text.

Multi-layer Perceptron (MLP) Classifier

An MLP classifier is a supervised learning algorithm that learns a function mapping a feature set X to an output set Y by creating layers of perceptrons. An example of a one-layer MLP is shown below.

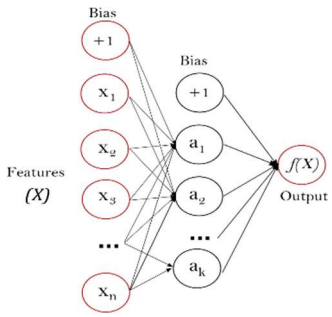


Figure 3 - Visualization of Single Layer Neural Net

Hyperparameter selection was based on an extensive search utilizing a Bayesian optimization process. The optimized parameters include hidden_layer_sizes and alpha.

Gaussian Process Classifier

The Gaussian Process Classifier is a supervised learning algorithm that classifies entries by the probability of which they contained in each group.

No hyperparameter optimization was done on this classifier.

AdaBoost Classifier

AdaBoost is an ensemble classifier that first fits a model on the original data and then creates additional copies with which it fits weighted data placing more emphasis on entries it classified wrong the first time.

Hyperparameter selection was based on an extensive search utilizing a Bayesian optimization process. The optimized parameters include n_estimaters and learning_rate.

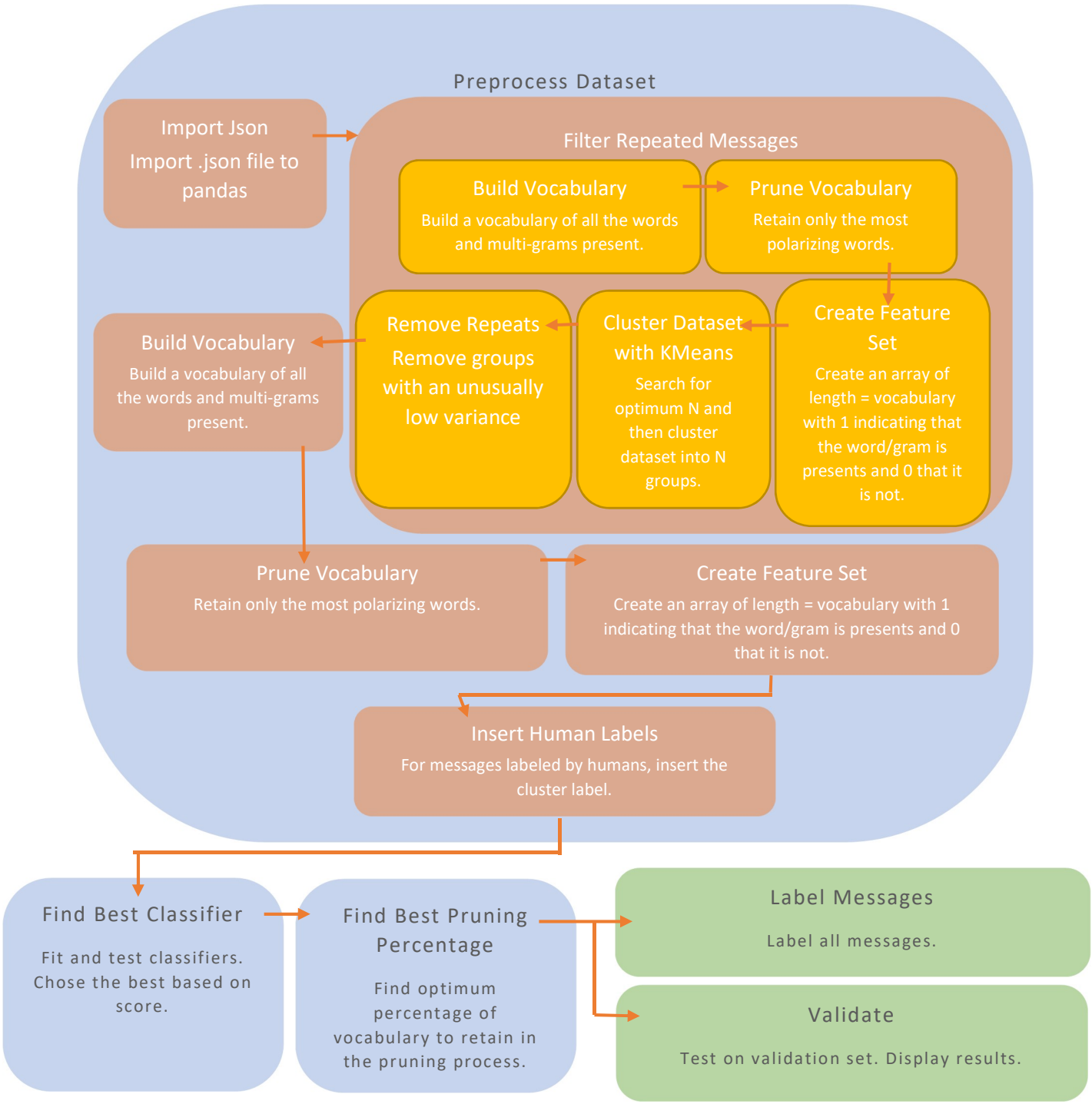
Decision Tree Classifier

A decision tree classifier is a method of analyzing past behaviors and using them to make predictions. The model creates a *tree* that consists of *yes/no branches,* which ultimately lead to a result *leaf,* that is outputted.

Hyperparameter selection was based on an extensive search utilizing a Bayesian optimization process. The optimized parameters include max_depth, min_samples_split, and min_samples_leaf.

Methodology

Project Overview



Results

Model Evaluation and Validation

Validation

The validation score for this model is ~97% surpassing its benchmark objective of 50%.

Outputs

Below are some samples from the two groups.

Group 0 = Financially Motivated

Group 1 = Genuine Expression

```
Messages from Group 0
RT @roushanalam: Fully obsessed with the way Kojo Funds leaves the
stage straight after his bit during Mabel's performance on Sounds Like
F…
It’s Friday finally 😇
DYK #MarieCurie learned to read by age four, impressing her three
older siblings? Have an amazing Friday and weeken…
https://t.co/Z8Yq03pnms
@hstylescouk_ That's just made my Friday so much better 🤩😇😍
RT @nflnetwork: “Love what you do.”

Start your Friday with some of the best advice @DangeRussWilson’s late
father ever gave him.

📺: @NFLG…
RT @supercontractUK: It’s Super #FreebieFriday! Follow
@supercontractUK &amp;amp; RT this status for a chance to win a £20 Amazon
voucher. T&amp;amp;Cs: ht…
RT @BeautyIsZion: It’s finally Friday! 😇
So it’s cash app Friday 🤔

Messages from Group 1
@DWF2006 @smith_lynne @frankcottrell_b @Pawmdapie @AllonsyAlondra
@SteffiKnows @bluebox99 Thank you very much Andre…
https://t.co/AY4rNOu4XJ
RT @arminvanbuuren: It's Friday!! let's trance up your weekend!
#TranceTop1000 https://t.co/nqcfLChjGu
I never knew someone to take Black Friday so serious until I started
working with @mindiii_ 🤩 She's got everything mapped out
```

RT @Amber02150: 🌥Morning Twitterworld🍺
🍁Happy Friday👍🍂🍁
🍂Great upcoming weekend🍁 https://t.co/FspDWFmLr4
have a blessing friday eveyone

#EMABiggestFansJustinBieber https://t.co/ZppHxbeBTs
@AnayaSen_ Your welcome, have a nice Friday!😘
Thank God It's Funken Friday 2017:14
https://t.co/95gXXM09Q8
Thank 🐻 God 😩 it's FRIDAY 🎉 ...

Justification

This project analyzed the sentiment of a given twitter message based on a classification of Genuine Expression or Financially Motivated. It is capable of filtering spam based on the frequency and variation of messages and then classifying the remaining messages based on content. The project received a score of 97% on a validation set that was never before seen by the program and exceeded the benchmark of random guessing by 47%. All this considered I would consider the venture a success.

Reflection

When I encountered this challenge my first thought was unsupervised learning. It seemed to me that I had a large collection of data and wanted to discover in what ways I could organize it into similar groups. I started with Kmeans and Gaussian Mixture Models but found that they were returning several clusters of messages that were identical (or almost identical). Given this, I calculated the standard deviation for each cluster and removed those which showed almost no variation. After implementing the filter, I fitted Kmeans and GM models again but I was still not getting clear clusters that were meaningful in the real world. I need to adjust my vocabulary to only the most important words and remove those which didn't add value.

When reading through the messages it became clear that there were two types of tweeters. Those that were tweeting to express themselves and those that were hoping for a monetary return. I began categorizing the data into the two groups. It appeared to be a good semi-supervised learning problem where I would begin labeling and the program would "watch", slowly getting better at predicting my classification until it was accurate enough to label the messages on its own. I built a script using Label Propagation and had the program prompt me to label the 25 entries it was most uncertain about. This continued until I had classified ~700 messages. At this point, the Label Propagation script was still not providing accurate results but the supervised learning algorithms were. Because time was the limiting factor in this challenge I set the semi-supervised learning to the side and began focusing on refining the supervised learning classifiers.

I still wanted to evolve my vocabulary/feature set so I built a stack containing my top three supervised learners. My hope was I could step through the data and slowly refine the feature set. For example, I would feed ~100 unlabeled message into my stack. Messages that were agreed upon by all three algorithms were then placed in the "bag" with the human-labeled entries. In the next pass, the

vocabulary would be built on this extended “bag” of labeled messages. In the end, it was not effective. The results were the same as or worse than individual classifiers.

I added multi-grams and manually removed common words from my vocabulary before training with a *prune_vocab* function. After these two efforts, I began to see encouraging results. I tuned the hyperparameters on my top four supervised classifiers using a Bayesian Optimization technique from a previous project. Finally, I found the optimum percentage of common words to remove by iteration over the options in steps of 10%.

Improvement

Going forward I would continue the project in two directions. First, I think it would be interesting to apply unsupervised learning to the new subsets of data and see if it can better distinguish groups. Possible predictions could include sentiments like religion, gratitude, anger, and celebration.

I also believe it would be interesting to chart the frequency of posts in these two subsets against the time of day or geographical location. You might be able to find that a spike in genuine messages is typically followed by a bust in motivated ones or vice versa. Or maybe all the businesses in a region start posting at 7 pm but the people of the region have already been building excitement since 4 pm? In the case of the challenge_en dataset, the messages were all captured over a span of 15 minutes with limited geographical information. As a result, none of these analyses were performed.

Summary

In summary, I enjoyed this project. I expanded my knowledge of semi-supervised learning and stack machine learning. I also have become much better at explaining what I do to my friends and why I was not available for social gatherings throughout the last week ☺

For questions, comments, or concerns my contact information is Kerri.Rapes@gmail.com