Machine Learning Engineer Nanodegree

Capstone Proposal

Alfred Anthony Castillo
June 2nd, 2018

Proposal

Domain Background

This project is based on the intersection of two domains: politics and tweets.

The twittersphere is fertile ground for machine learning and natural language processing research. A quick web search easily finds a number of machine learning research projects over twitter: Tweets Sent While Drunk, Tweets by Sellers of Opioids, and more generically, Sentiment Analysis of Tweets.

Politics in the United States, although less immediately accessible to machine learning research, is also an intriguing domain. For example, a class being offered at Duke University (POLSCI 189FS) attempts to predict congressional voting by examining the text of bills. The long-term ramifications of projects like these are significant.

Examining tweets that are political in nature or attempting to determine the political nature of tweets can be interesting research projects. Furthermore, they can serve as the starting points for deeper analysis if desired.

Problem Statement

This project is a straightforward classification problem: identify tweets as either `Democrat` or `Republican`. We intend to use tweets as inputs to a model. The algorithm will then output a prediction of whether each tweet was made by a `Republican` or a `Democrat`.

Text sentiment analysis and classification is a well-trodden area of research. One example is email spam classification. This has reached the point where the spam problem is considered by many as simply a minor nuisance (1, 2). This success was achieved using Bayesian filtering, an approach that has been under consideration since at least 1998. This technique has also been used to classify text in other domains, like detecting positive or negative sentiment in movie reviews. It is therefore reasonable to attempt to use it classify the tweets in our dataset as `Republican` or `Democrat`.

More recently, research has been conducted to see if neural networks can improve upon the Bayesian approach.

For this project, we attempt to improve upon the Bayesian approach by using a neural network.

Because of the pass or fail nature of the classification task, success or failure is easily ***quantifiable*** (it will be easy to identify success or failure for each case) and ***measurable*** (summing up correct and incorrect label predictions is relatively simple). The experiment should also be ***replicable*** since the tweets and algorithms used are freely available for similar studies in the future.

Datasets and Inputs

The dataset for this project can be found on Kaggle. It is a collection of over eighty-six thousand tweets, each of which is labeled as either `Democrat` or `Republican`, depending on the affiliation of the person sending the tweet. This dataset will be used to train algorithms that create prediction models that can be applied to other tweets. Natural language processing techniques will be used to analyze the text of each tweet and determine its party affiliation (as defined by the accompanying label).

Approximately forty-four thousand tweets are labeled `Republican` from 222 different Republicans, while the rest (about forty-two thousand) are labeled `Democrat` from 211 Democrats — almost an ideal 50/50 split. The average length of each tweet is 129 characters and the maximum length is 166 characters.

The dataset was retrieved by a Kaggle user (*kapastor*) from Twitter's public feed and was released under the Creative Commons Public Domain license.

Solution Statement

We are fortunate to begin with a clean and labeled dataset, therefore minimal data cleanup (if any) will need to be conducted.

In preparation for algorithm training, each tweet in the dataset will be vectorized using the bag-of-words technique.

Once vectorized, we will train a multinomial naive Bayes classifier to set a benchmark.

Finally, we will train at least one neural network algorithm to see if it can exceed the performance of naive Bayes.

The performance of each approach can be quantified using an F1 score. This can be measured by observing its performance over testing data sets. Randomizing test and train sets may result in variations in results between iterations, but using a fixed test and train set will allow reproducibility. In addition, executing many iterations over randomized test/train sets should converge to a reproducible value.

Benchmark Model

Naive Bayes is a workhorse of the classification genre. This model's performance will be the standard against which we will compare our neural network techniques.

We will quantify its performance by calculating an F1 score.

Evaluation Metrics

The F1 score is a standard metric used for measuring binary classifier accuracy.

F1 is calculated as a weighted average of *precision* and *recall*, where precision is accuracy over selected elements and recall is completeness over all elements:

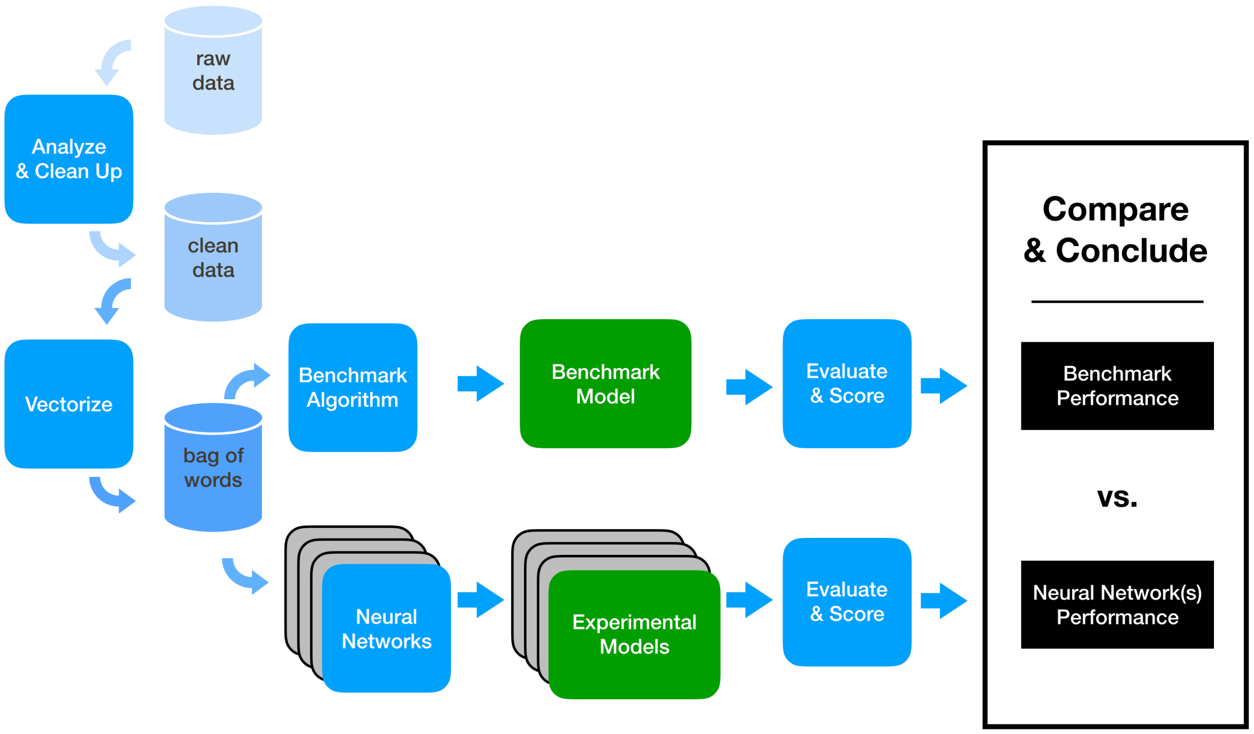
```
given:
precision = true positives / (true positives + false positives)
recall = true positives / total elements

then:
F1 = 2 * (precision * recall) / (precision + recall)
```

Project Design

The following steps describe the planned workflow for this project:

1. Analyze and Profile Data
2. Vectorize Tweets
3. Establish the Benchmark
4. Train a Neural Network
5. Formulate Conclusion



Capstone Workflow

1 — Analyze and Profile Data

The first step is to determine if any cleanup is necessary. We are expecting a minimal amount of work in this step, if any. Additional summaries and statistical evaluations will be made of the data set to get a better feel for what it contains.

2 — Vectorize Tweets

After a clean dataset is created, `bag-of-words` will be applied to each tweet, thus vectorizing them. A tf-idf calculation will be applied to each word in the set which will be used to weight each word's importance to the analysis.

3 — Establish the Benchmark

A benchmark for performance will be created using a multinomial naive Bayes algorithm. An F1 score will be calculated against this model's performance.

4 — Train a Neural Network

In this step of the workflow, experimentation will take place. At least one, possibly several, neural network architectures will be used to create models whose performance(s) will be compared to the benchmark model's performance.

5 — Formulate Conclusion

A conclusion will be drawn as to whether or not we were successful in building one or more neural networks that achieved better performance than the benchmark model. We will project the results into possible directions for future research into the political nature of tweets.