Machine Learning Nanodegree
Capstone Proposal

Vaishnav Vishal

April 1, 2017

1 Domain Background

Supervised learning is the machine learning task in which the algorithms reason from externally supplied instances to produce general hypothesis, which then make predictions about future instances. It is the task of deriving a function from labeled training data.

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages and, in particular, concerned with programming computers to fruitfully process large natural language corpora. Challenges in natural language processing frequently involve natural language understanding, natural language generation (frequently from formal, machine-readable logical forms), connecting language and machine perception, managing human-computer dialog systems, or some combination thereof.

Tweet encoding can be used to take data from tweets and use NLP on it to know about the behavior of the tweets to analyse similar kind of tweets. Twitter provides many APIs for downloading its tweets by users. Many people use these APIs to analyse the tweets realted to a particular topic.

So, I'm going to take a dataset consisting of tweets from Hillary Clinton and President Donald Trump and would predict whether a given tweet is from Trump or Hillary by training a model on this dataset.

2 Problem Statement

In this project, we will use supervised learning to train a model on a tweet dataset and then given a tweet, we have to predict who tweeted it.

- **Task:** Predicting the tweeter.
- **Performance:** Accuracy - No. of correct predictions.
- **Target function:** A function that gives weights to the terms in the tweets and then tells us who the author is.
- **Target function representation:** A Classification model.

Therefore, I seek to use tf-idf scores and a Naive Bayes classifier to predict whether a tweet is more likely to have been tweeted by @realDonaldTrump or @HillaryClinton.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6444 entries, 0 to 6443
Data columns (total 28 columns):
id                          6444 non-null int64
handle                      6444 non-null object
text                        6444 non-null object
is_retweet                  6444 non-null bool
original_author             722 non-null object
time                        6444 non-null object
in_reply_to_screen_name     208 non-null object
in_reply_to_status_id       202 non-null float64
in_reply_to_user_id         208 non-null float64
is_quote_status             6444 non-null bool
lang                        6444 non-null object
retweet_count               6444 non-null int64
favorite_count              6444 non-null int64
longitude                   12 non-null float64
latitude                    12 non-null float64
place_id                    204 non-null object
place_full_name             204 non-null object
place_name                  204 non-null object
place_type                  204 non-null object
place_country_code          204 non-null object
place_country               204 non-null object
place_contained_within      204 non-null object
place_attributes            204 non-null object
place_bounding_box          204 non-null object
source_url                  6444 non-null object
truncated                   6444 non-null bool
entities                    6444 non-null object
extended_entities           1348 non-null object
dtypes: bool(3), float64(4), int64(3), object(18)
memory usage: 1.2+ MB
```

Figure 1: Info

3 Datasets and Inputs

Figure 1 shows the info about the data.

Twitter has played an increasingly prominent role in the 2016 US Presidential Election. Debates have raged and candidates have risen and fallen based on tweets.

The dataset I chose provides 3000 recent tweets from Hillary Clinton and Donald Trump, the two major-party presidential nominees.

It can be downloaded from Tweets Dataset

It contains 3000 tweets in form of csv values with about 28 attributes. It takes around 1MB space.
The main features of the dataset are:

- twitter handle
- text
- is retweet
- language
- original author
- time
- retweet count
- favorite count

and 20 other features

There are various Twitter APIs also using which we can manually download the tweets dataset if we wish to.

4 Solution Statement

To tackle the problem described in Section 2, we will use Supervised Learning to automatically learn the terms used in the tweets of Hillary Clinton and Donald Trump and will train a classification model to predict the likeliness of the tweet being tweeted by a specific user.(In our case, Donald Trump or Hillary Clinton). So, we will be creating a function that takes an input tweet and then tells us who the author is.

For encoding the tweets I will be using TFIDF Vectors.

In information retrieval, tf–idf, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus It is often used as a weighting factor in information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes. For instance, 83% of text-based recommender systems in the domain of digital libraries use tf-idf.

My Final Model will use Multinomial Naive Bayes

With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial $(p_1, \ldots, p_n)$ where p_i is the probability that event i occurs (or **K** such multinomials in the multiclass case). A feature vector $\mathbf{x} = (x_1, \ldots, x_n)$ is then a histogram, with x_i counting the number of times event i was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document (bag of words assumtion). The likelihood of observing a histogram **x** is given by:

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable to incorporate a small-sample correction, called pseudocount, in all probability estimates such that no probability is ever set to be exactly zero. This way of regularizing naive Bayes is called Laplace smoothing when the pseudocount is one, and Lidstone smoothing in the general case.

5 Benchmark Model

- **Gaussian Naive Bayes**. In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

  Gaussian Naive Bayes: When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute, **x**. We first segment the data by the class, and then compute the mean and variance of **x** in each class. Let μ_c be the mean of the values in **x** associated with class c, and let σ_c^2 be the variance of the values in **x** associated with class c. Suppose we have collected some observation value **v**. Then, the probability distribution of **v** given a class **x**, **p(x=v)**, can be computed by plugging **v** into the equation for a Normal distribution parameterized by μ_c and σ_c^2. That is,

  $$p(x = v \mid c) = \frac{1}{\sqrt{2\pi\sigma_c^2}}\, e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

  I'll be using the Naive Bayes classifier as the benchmark model because it will always predict either of the characters(Hillary or Trump).

  I'll be looking to maximize the accuracy of the predictions using this model.

- **Tweet Dataset**. This dataset will be used as the main benchmark. The learned frequencies of terms from 3000 datapoints will be used to train the binary clasification model.

6 Evaluation Metrics

- **Prediction accuracy.** The TF-IDF scores will be used to predict the user who tweeted the tweet. The no. of correct predictions will be our accuracy and will be used to evaluate the model.

As this is a classification type of problem, accuracy turns out to be the best evaluation metric to evaluate the performance of the model.

7 Project Design

7.1 Programming Language and Libraries

- **Python 2.**
- **scikit-learn**. Open source machine learning library for Python.
- **numpy**. Python's numerical library.
- **matplotlib**. For plotting graphs and curves.
- **pandas, seaborn** For data reading and visualization

7.2 Strategy

First we extract the mentions and hashtags they have used from their individual tweets.

After that we create the bag of words from all the tweets and then put the bag of words in a count vectorizer. And then calculate the TF-IDF scores.

And finally we train our multinomial model on TF-IDF vectors to make a prediction.

This is precisely what we will be doing to make our model work.

7.3 Machine Learning Algorithm

Multinomial Naive Bayes: Please visit Section 5 to know about the algorithm I will be using to train the model.