**Natural Language Processing of Sentiment in Tweets**

Quenton Xiao

August 1, 2022

We conduct sentiment analysis of tweets using Natural Language Processing (NLP) techniques. The sentiment consists of two polarized values, positive and negative. We describe and use standard techniques in the field, achieving high accuracy.

**Introduction**

Machine Learning is the field of computer science that uses data to develop computational models to make predictions and help make decisions [1, 3]. There are several classes of type of problems within machine learning. The problem we consider in this article belongs to the class of problems known as supervised learning [2].

Consider Table 1. That table lists tweets and their sentiment. This is not the data set we analyze in this article. Table 1 contains only a few, very short tweets. Nevertheless, Table 1 serves well to illustrate the concepts we will be introducing.

|  |  |
| --- | --- |
| Tweet | Sentiment |
| I am happy because I am wealthy | Positive |
| I am happy | Positive |
| I am sad because I am poor | Negative |
| I am sad | Negative |

Table 1. List of tweets and their sentiments.

The first column of Table 1 contains the tweet. The second column contains the sentiment of the tweet, i.e. if the tweet is positive or negative. The sentiment of each tweet is provided by the person that wrote the tweet; it is not inferred from any other source. For example, the first tweet is *I am happy because I am wealthy* and its sentiment was given as *Positive.*

The standard practice in the field of machine learning is to use the term “example” to refer to the combination of, in our case, a tweet and its sentiment. The first example of Table 1 is the tweet *I am happy because I am wealthy,* together with its sentiment *Positive.* The sentiment of each tweet is called the “label” of the tweet. Thus, the label of the first tweet in Table 1 is *Positive.* On the other hand, we will refer to the tweet as the “feature” of the example. The feature of the first example in Table 1 is the tweet *I am happy because I am wealthy.* Problems such as the one we consider in this article, where the data is a collection of examples, and each example consists of one or more features and a label, are known as supervised learning problems.

When working on a supervised learning problem, the goal is to develop computational models to accurately predict the label of an example from its features. The strategy to accomplish this can be summarized as follows:

1. We start with a data set where both the features and the labels of the examples are known. In our case, we start with a data set consisting of the tweets and their sentiments.
2. The data set described in the previous step is split into two sets of examples. One set is called the training set, and the other set is called the test set.
3. The training set is used to develop the computational models.
4. The test set is used to evaluate how well the model performs.
5. The model obtained in the previous steps is used to predict the label of any new example of which only the features are known.

Note that the model is a function that takes as input the features of an example and gives as output an estimate of the label of the example. In our case, the model will predict the sentiment of a tweet not in the data set given to us (and thus, with an unknown sentiment).

In the rest of the article, we give an overview of the machine learning techniques we will use. We then apply those techniques to our data set, and finally report the results we obtain.

**The data set and the pre-processing of the tweets**

The data set we use are the tweets in twitter\_samples, that can be imported from the Python library nltk. This is a set that has been used widely to test different models. It consists of 10,000 tweets. The sentiment of half of them is positive, and the sentiment of the other half is negative. One of the positive tweets is

*My beautiful sunflowers on a sunny Friday morning off :) #sunflowers #favourites #happy #Friday off…* [*https://t.co/3tfYom0N1i*](https://t.co/3tfYom0N1i)

Note that this tweet does not contain just words. It contains a hyperlink, [*https://t.co/3tfYom0N1i*](https://t.co/3tfYom0N1i)*,* it contains punctuation such as “:)” and “…”, and it contains symbols, #. The first step in working with a data set such as this one is to clean and pre-process the data. Pre-processing the data includes removing hyperlinks, some punctuation and manipulating words by using regular expression and employing a well-known NLP public library called NLTK. Manipulating words may include but not be restricted to tokenization, removal of stop words, stemming or lemmatizing and converting upper-case to lower-case. Stop words are some function words in linguistics, such as ‘the’, ‘is’, ‘which’, which do not contribute a lot of meaning to the main content of text. Stemming is the process of reducing inflectional forms and sometimes derivationally related forms of a word, such as plural forms, gerunds, etc., to one canonical base form. This will lessen the burden of computation and avoid treating tokens with the same root as different words. For example, after pre-processing, the above tweet may read

*my beautiful sunflowers on a sunny Friday morning off :) sunflowers favourites happy Friday off*

While pre-processing is a crucial step, and we have carried it out in our analysis, we will not discuss it in further detail here. We will assume that each tweet contains only words, no symbols or punctuation, like the tweets in Table 1.

**Positive and negative frequencies of words**

The positive frequency of a word is the number of times the word appears in all the positive tweets in the data set. To illustrate this, assume the data set is Table 1. The positive frequency of the word *I* is 3, because *I* appears twice in the first tweet, which is *I am happy because I am wealthy*, and it appears once in the second tweet, which is *I am happy,* and these two tweets are the only positive tweets in the data set. Similarly, the negative frequency of a word is the number of times the word appears in all the negative tweets in the data set. Table 2 shows the positive and negative frequencies of all the words that appear in a tweet of the data set, when the data set is Table 1.

|  |  |  |
| --- | --- | --- |
| Word | Positive frequency | Negative frequency |
| I | 3 | 3 |
| am | 3 | 3 |
| happy | 2 | 0 |
| because | 1 | 1 |
| wealthy | 1 | 0 |
| sad | 0 | 2 |
| poor | 0 | 1 |

Table 2. Positive and negative frequencies of the words in the tweets in Table 1.

**Feature engineering**

In this section we continue using the data set in Table 1 to illustrate the concepts we introduce.

To each tweet, we will assign two numbers. We will call these numbers the positive and the negative features of the tweet. The positive feature of a tweet is the sum of all the positive frequencies of the words that appear in the tweet. The positive frequency of a word counts toward the positive feature of the tweet as many times as the word appears in the tweet. For example, the first tweet in Table 1 is *I am happy because I am wealthy.* The positive frequencies of the words are listed in Table 2. Thus, the positive feature of this tweet is .

The negative feature of a tweet is defined analogously. The negative feature of a tweet is the sum of all the negative frequencies of the words that appear in the tweet. The negative frequency of a word counts toward the negative feature of the tweet as many times as the word appears in the tweet. For example, the negative feature of the tweet *I am happy because I am wealthy* is .

In Table 3 we list the tweets in Table 1, their features, and their labels. We denote by the positive feature, by the negative feature and by the labels. For each tweet, its value of is equal to if the tweet is positive or if the tweet is negative.

|  |  |  |  |
| --- | --- | --- | --- |
| Tweet |  |  |  |
| I am happy because I am wealthy |  |  |  |
| I am happy |  |  |  |
| I am sad because I am poor |  |  |  |
| I am sad |  |  |  |

Table 3. is the positive feature. is the negative feature. is the label.

In the rest of this article, when we refer to the features of an example or a tweet, we refer to the positive and negative features, i.e. the and of the tweet. When we refer to the label of an example or tweet, we refer to the of the tweet.

**Binary classification problems**

Supervised learning problems where the label of each example takes one of two possible values are known as binary classification problems. Each tweet is either positive or negative. The label of each tweet takes one of two possible values: if the tweet is positive, and if the tweet is negative. Thus, the problem we are considering in this article, of classifying each tweet as either positive or negative, is a binary classification problem. We will say that an example or tweet belongs to the category if it is positive. Similarly, a tweet or example belongs to the category if it is negative.

A model for binary classification problems is a function that takes as input the features of an example and gives as output a number between 0 and 1. As is common practice, we denoted this function by . Thus, assuming that each example has only two features, as it is in our case, we have , i.e. the output is a function of the features and . The prediction of the model is that the example belongs to the category 1 if and to the category 0 if .

Note that we have not explained yet how the function is constructed. We will provide such an explanation later in this article.

**Logistic regression**

Logistic regression is a machine learning technique that is used to develop models for binary classification problems. This is the technique that we use in this article, and that we explain in this section.

We start by introducing the sigmoid function. This function is

The graph of this function is displayed in Figure 1.

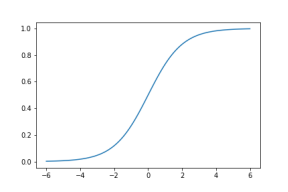


Figure 1: Plot of the sigmoid function

The important properties of the sigmoid function are:

1. for all
2. is an increasing function of
3. approaches as becomes large in absolute value but negative.
4. approaches 1 as increases.

In the rest of this article, we assume that each example has two features, even when we talk in general terms, not only when referring to sentiment of tweets.

Let and denote the two features. In Logistic regression, the functional form of the prediction is assumed to be

where , and are some numbers known as parameters. To explain how these parameters are selected, we first need to introduce the notion of error.

Assume that the features of an example are and . Assume that we know the label of that example and its label is . Note that is either 1 or 0. On the other hand, our model predicts the label of this example to be . Note that, as previously discussed, . The binary crossentropy error on this example is

where denotes the natural logarithm.

While we will not go into the details of the binary crossentropy error, we list here its properties that are most relevant to us:

1. if
2. The closer is to , the smaller is.
3. If , then .

For the reasons listed above, is a measure of the difference between and . Thus, can be considered as a measure of the error the model makes in predicting the label of the example.

The mean binary crossentropy error on a set of examples is the average of the binary crossentropy errors on the examples in the set. Note that this error depends on the parameters , and . If we change those parameters, the error will also change. In logistic regression, the parameters that are selected are those that make mean binary crossentropy error on the set of examples we use to develop the model as small as possible.

The set of examples that are used to develop the model is called the set of training examples. Note that we know the labels of the training examples. To develop a model, we are initially given a set of examples from whom we know their features. We may use all those examples as training examples, or only a fraction of those examples. Usually around 70% of all the examples given are used as training examples. The rest of the examples are saved as test examples. The test examples serve to evaluate the performance of the model once the parameters are found. We refer the reader to [1, 2, 3] for more details.

We will not go into any details on the algorithms used to find the parameters , and . In practice, these parameters are usually found using software libraries that are available to be used by the public at no cost.

**Results**

In this section we apply logistic regression to our data set of 10,000 tweets. We start by displaying a scatter plot of our dataset in Figure 2. Each dot corresponds a tweet. The horizontal component of the tweet is its positive feature. The vertical component of the tweet is its negative feature. The positive tweets, i.e. with label , are plotted in orange. The negative tweets are plotted in blue.

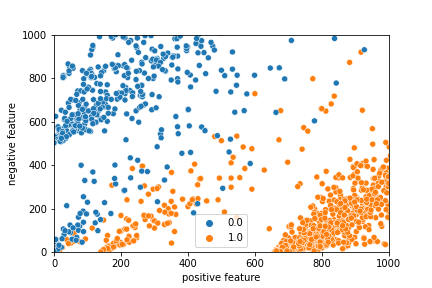


Figure 2. Scatter plot of the data set.

We first split our set of examples into two sets: the training set and the test set. The training set consisted of 75% of the examples, picked randomly. The remaining 25% of the examples formed the test set. We use the set of training examples to create the model, i.e. to find the parameters, and . We use the library Keras to find the values of these parameters. We found that, in 99% of the test examples, the model predicted the correct sentiment. Therefore, the model is extremely accurate.

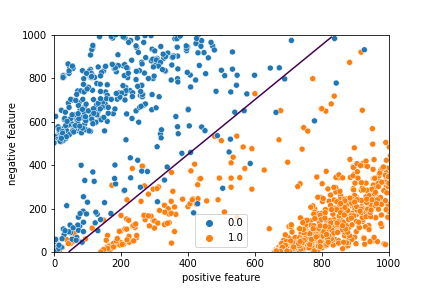
Note that the model predicts an example with features and to be positive if . We are not displaying the values , and that the model found because they are not very instructive. Note that is the same as . Similarly, the model predicts an example with features and to be positive if . In other words, the straight line splits the plane of Figure 2 in two. The model predicts the tweets to be positive if its corresponding point is on one side of this line and negative if the point is on the other side. In Figure 3, we made the same scatter plot as in Figure 2, but we also plotted in Figure 3 the line . This line is known as the decision boundary. Note that, in fact, the decision boundary leaves to one side most the positive tweets, i.e. the orange dots, and most of the other tweets on the other side, the blue dots.

Figure 3. Scatter plot of the data set and plot of the decision boundary

**Conclusions**

Machine learning is a powerful field of computer science that is finding applications in numerous fields. In this article, we showed how it can be used as a tool in predicting the sentiment of tweets. These techniques are also applicable in other areas of everyday life, such as marketing, user analytics, online content moderation, and more.

**References**

[1] Andriy Burkov. The hundred-page machine learning book, volume 1. Andriy Burkov Canada,  
2019.  
[2] Sebastian Raschka. Python machine learning. Packt publishing ltd, 2015.

[3] Cha Zhang and Yunqian Ma. Ensemble machine learning: methods and applications. Springer,  
2012.