Iqra Ameer

111 220 885

Analyse et traitement de données massives

GLO-7027

Project Title: Twitter Sentiment Analysis

Final Report

Professor: Richard Khoury

18th April 2018

Working of Naïve Bayes Algorithm

We have chosen Naïve Bayes (NB) algorithm for our Sentiment Analysis Task. In simple terms, the NB algorithm enables us to predict a class, given a set of features using probability. In technical point of view, researchers are concerned about the prediction that is the posterior probability. The prior knowledge is named as prior probability that returns the most possible guess on the outcome without additional knowledge. The present indication is expressed as probability that reflects the probability of a predictor given a certain outcome.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (1)$$

Where P (A) and P (B) are probability of events A and B respectively without concerning each other. Term P (A|B) is indicating the probability of A conditional on B and P (B|A) is the probability of B conditional on A.

I will use an example to explain how the naïve Bayes classification works. Let us assume that we have data on 1,000 pieces of fruit. The fruits in the dataset are bananas, oranges and some other fruit. We are considering three features (Long, Sweet, Yellow) for fruits as listed in Table 1 below.

Table 1: Fruits Example - Features & Quantities [1]

Fruit	***Long***	***Sweet***	***Yellow***	***Total***
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	**500**	**650**	**800**	**1000**

We can predict whether a fruit is an orange or a banana, based on its features. Here we can see that, in the whole set of fruits the percentage distribution is: 50% of the fruits are bananas, 30% are oranges and 20% are other fruits. From our training set, which attributes are showed in

Table 1, we can declare that, from 500 bananas:

- 400 are long.
- 350 are sweet.
- 450 are yellow

Similarly, from 300 oranges:

- 0 are long.
- 150 are sweet.

- 300 are yellow.

From the remaining 200 fruits:

- 100 are long.
- 150 are sweet.
- 50 are yellow.

To predict the category to which another fruit belongs, let us suppose that we get an input a fruit to classify, on the basis of the following features: long, sweet and yellow, then we can classify it using the Naive Bayes Equation (1) for each case and compare all of the results. The winner will be the one with the highest probability, as follows:

• P (Banana) = 0.252.

• P (Orange) = 0.

• P (Other fruit) = 0.01875.

The highest probability is 0.252, so as per Naive Bayes we can assume that the long, sweet and yellow fruit that was recently introduced is in fact a Banana.

We would like to show general architecture of our approach before moving on results and discussion.

In our experiments original Naïve Bayes algorithm is performing slightly better than other classifiers as results are listed in Table 2.

Our data set of 5300+ positive and 5300+ negative tweets, which are much shorter. When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and we need less training data and NB does not overfit as much as others. It is easy, fast to build a model and less expensive computationally to predict class of test data set.

Figure.1 showing a general architecture of our model.

Results and Analysis

We have used dataset of short tweets 5300+ positively tagged and 5300+ negatively tagged text which is publically available[1] and experimented with different classifiers as listed: Original Naïve Bayes classifier, BernoulliNB_Classifier, SDG_Classifier, LinearSVC_Classifier and

[1] https://pythonprogramming.net/static/downloads/short_reviews/ Last visited: 4/15/2018

NuSVC_Classifier in our previous submission. We have used accuracy as evaluation measure Accuracy.

$$Accuracy \quad = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

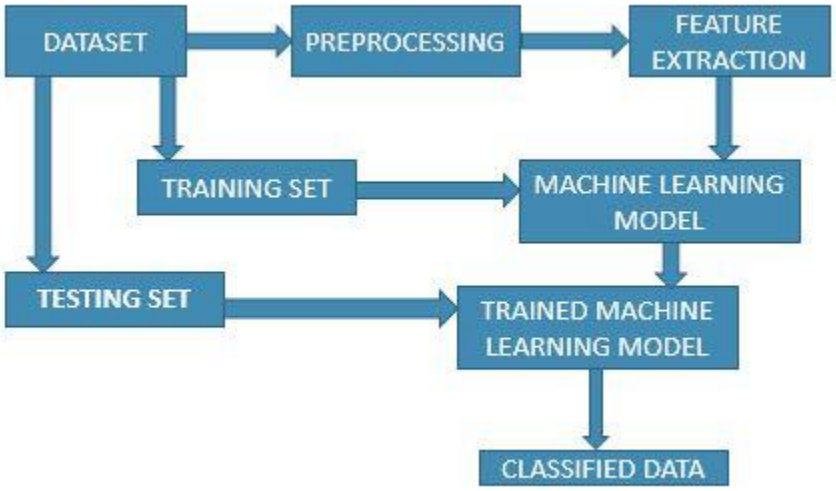

Figure 1: General Architecture of the model

We have seen that out of all these settings Original Naïve Bayes is giving us maximum scores (Accuracy = 75.679) as compared to MNB_Classifier, BernoulliNB_Classifier, LogisticRegression with minimum margin. So, for the final submission we have chosen Naive Bayes classifier to run our rest of the experiments due to the following facts: Naive Bayes can tackle missing data. Features are handled separately by the algorithm at the time of model construction and for prediction time as well.

As such, if a data example has a missing value for an attribute, it can be ignored during the model construction, and unnoticed when a probability is calculated for a class value.

Table 2: Results of different classifiers from last submission with Voted Average

Classifier	***Accuracy (%)***
Original Naïve Bayes	**75.67**
BernoulliNB_Classifier	75.37
SDG_Classifier	69.03
LinearSVC_Classifier	72.96
NuSVC_Classifier	73.26

Naïve Bayes Algorithm

Here, the job is to determine the probability that specific tweet *t* is in class *c*, and class can be c = 1 or 0. We implemented a unigram based Naive Bayes model.

$$p(c|t) = P(c) \prod_{1 \le n \le m_t} P(d_n|\, c) \qquad (3)$$

Here d_n indicating the *nth* token in sequence in a tweet, and m_t denotes the size of a specific tweet. The Naive Bayes hypothesis is that these probabilities for each token are independent, and therefore the combined probability is just the product.

Below table is indicating the scores achieved at different sizes for the Naïve Bayes algorithm.

Table 3: Accuracy of Naïve Bayes Algorithm on different Sizes of the Dataset

Dataset	***Accuracy (%)***
10	56.23
50	53.23
100	55.25
500	62.22
1000	62.25
5000	64.71
10000	**72.50**

Following is the list of 10 most informative features after the execution of the classifier on the training data.

provides = True pos : neg = 18.9 : 1.0

refreshing = True pos : neg = 13.6 : 1.0

refreshingly = True pos : neg = 13.0 : 1.0

warm = True pos : neg = 12.6 : 1.0

inventive = True pos : neg = 12.3 : 1.0

absorbing = True pos : neg = 12.3 : 1.0

wonderful = True pos : neg = 11.8 : 1.0

mesmerizing = True pos : neg = 11.6 : 1.0

delicate = True pos : neg = 11.6 : 1.0

captures = True pos : neg = 11.4 : 1.0

Effect of Stopwords

When we run the Naïve Bayes algorithm on different sizes of the dataset, it gave maximum an accuracy of 72.50 percent. We noticed that Stopwords hardly carrying any information, orientation and arising repeatedly [2].

The next thing we considered in our final experiments was elimination of stopword. In our development we were able to solve this by using Natural Language Toolkit (NLTK) English Corpus Stopwords.

After removing the stop words Naïve Bayes was run, it gave an accuracy of 74.56 percent, approximately 2% increase in accuracy. We also did a quick experiment on weka[2] by using character n-grams 1-3 (min and max limit). That was not giving us significant results, one reason why it fails could be that words are more likely to present the theme of the discussion than characters.

Following table demonstrating the score achieved at different sizes for the Naïve Bayes algorithm with removal of stopwords and based on unigram constructed model.

Table 4: Accuracy of Naïve Bayes Algorithm on different Sizes of the Dataset (removal of stopwords + unigram)

Dataset	***Accuracy (%)***
10	51.41
50	52.05
100	54.94
500	62.32
1000	63.02
5000	70.57
10000	**74.16**

Following is the list of 10 most informative features for Naïve Bayes after the removal of stop words.

[2] Weka is a collection of machine learning algorithms for data mining tasks. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

dull = True neg : pos = 20.7 : 1.0

generic = True neg : pos = 17.0 : 1.0

routine = True neg : pos = 15.7 : 1.0

unexpected = True pos : neg = 14.3 : 1.0

flat = True neg : pos = 13.8 : 1.0

boring = True neg : pos = 13.3 : 1.0

90 = True neg : pos = 13.0 : 1.0

warm = True pos : neg = 13.0 : 1.0

wonderful = True pos : neg = 12.6 : 1.0

stale = True neg : pos = 11.7 : 1.0

The predictions are varying; we observed the same situation when used SVC classifier. This indicating that stopwords are having the influence on scores. An intuition to this situation can be attained from the experience that given the short length of tweets [2], people generally use stopwords such as any, again, after, been and so on. Therefore removal of stopwords makes a considerable difference in predictions.

As the above table 4 shows, when the experiments were done on the larger corpus, the accuracy scaled. Naïve Bayes accuracy scaled up to 74.16 percent. It took us 3:32 minutes to run for Naive Bayes algorithm and we are running it on i7 4GB. We can say that by increasing the dataset we can achieve the better accuracy but not always.

We have observed a mistake in our plan that, we should have considered the miss-spelling words. We have seen some features of tweets that could be standardized through preprocessing of corpus. Due to the limitation of 140 characters on twitter users, there is possibility for authors to have spelling mistakes, idioms, and slang words in their text. For Example BFN: Short for "bye for now", b4: Twitter shorthand for "before."[3] Replacing these words with original once can help us in order to improve performance of our model.

Moreover, we could use more features for the Naive Bayes instead of just word based. For example, use of Part-of-Speech tags to determine not only which words contribute to negative/ positive sentiment [3], but also what POS are affecting the sentiment of a tweet. For instance, the two tweets "just fooling around before my econ final" and "my last test made me feel like a giant fool" carrying completely changed levels of sentiment however the word "fool." is in both sentences. We know that POS tagger needs more clean data, and Twitter users used to have

[3] https://www.webopedia.com/quick_ref/Twitter_Dictionary_Guide.asp Last visited: 04/16/2018

grammatical and spelling mistakes as mentioned in above paragraph, after dealing with spelling mistakes we can consider this option for twitter data.

Although word unigram is giving reasonable accuracy with Naïve Bayes algorithm but in future, we can come up with different approach like, can focus on combining machine learning methods in order to improve the accuracy of sentiment prediction.

Here is Github link to the project: https://github.com/iqraameer133/Big-Data-Project

References

[1] AYLIEN. Naive Bayes for Dummies; A Simple Explanation. Accessed on: 31 of January of 2017, Jun 2015. URL http://blog.aylien.com/naive-bayes-for-dummies-a-simple-explanation/

[2] KHARDE. Sentiment analysis of twitter data: A survey of techniques. Journal: arXiv preprint arXiv: 1601.06971, 2016. URL https://arxiv.org/ftp/arxiv/papers/1601/1601.06971.pdf

[3] PARIKH. Sentiment analysis of user-generated twitter updates using various classification techniques. Journal: CS224N: 118, 2009. URL https://nlp.stanford.edu/courses/cs224n/2009/fp/19.pdf