Sarcasm Detection in News Headline Data Using Data Mining Techniques

Md. Jannatul Nayeem, Md. Arafatullah Turjoy, Joy Bhowmick, Rifat alam Akash
Department of Computer Science and Engineering, East West University
jannatulnayeem423@ieee.org,arafatullah@gmail.com

ABSTRACT

In recent years, data mining has turned out to be a standout amongst the most requesting areas of software engineering which generally deals with discovering frequent patterns by applying intelligence tools, techniques and methodologies from various kind of databases. But with the rapid growth of modern technology, high volumes of data with different characteristics are generated by today's applications. We are proposing an application which will basically works for sarcasm detection in news headline. This will works by using sentiment analysis with a huge dataset. By collecting many real time data, this application detect the sarcasm. By using data mining techniques, sarcasm is detected. And in this paper, we focused on accuracy and errors by implementing different algorithms called Naive Bayes Classifier, Decision tree classifier, Neural network with multi layer perception and then compare the accuracy of these algorithm for sentiment analysis.

KEYWORDS

Data Mining, Sentiment Analysis, Sarcasm Detection

ACM Reference format:
Md. Jannatul Nayeem, Md. Arafatullah Turjoy, Joy Bhowmick, Rifat alam Akash . . Sarcasm Detection in News Headline Data Using Data Mining Techniques. In *Proceedings of , , ,* 3 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In recent year sentiment analysis is very popular analyzing topic in the field of artificial intelligence specifically in machine leaning and data mining. Sentiment analysis is about to provide feelings to the machine. Some popular sentiment datasets are twitter review, restaurant review and some datasets about abusive contents. In this research we have chosen a dataset which contain news headlines of some articles along with labels which are a headline is sarcastic or not. We have applied two different data mining techniques as well as a deep learning approach for constructing a classifier for each method which can be classified after that between sarcastic or non-sarcastic news headline and also for a comparative study among all those approaches.

, ,
© Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 DATASET DESCRIPTION

Our dataset is about News Headlines dataset for Sarcasm Detection which have been collected from two news website. TheOnion aims at producing sarcastic versions of current events and from those events collection has been done all the headlines from News in Brief and News in Photos categories (which are sarcastic) as well as real (and non-sarcastic) news headlines from HuffPos and rather have been uploaded a website name Kaggle and we have collected this dataset from there. This dataset has some advantages over twitter dataset like since news headlines are written by professionals in a formal manner, there are no spelling mistakes and informal usage. This reduces the sparsity and also increases the chance of finding pre-trained embeddings. Furthermore, since the sole purpose of TheOnion is to publish sarcastic news, it contain high-quality labels with much less noise as compared to Twitter datasets. Unlike tweets which are replies to other tweets, this obtained news headlines are self-contained. This would be helpful for teasing apart the real sarcastic elements. Every tuple of this dataset contain 3 attributes which are sarcastic: 1 if the record is sarcastic otherwise 0, headline: the headline of the news article and articlelink: link to the original news article. Useful in collecting supplementary data.

3 IMPLEMENTATION DETAILS

The dataset is in json format. We have read the data using pandas library and converts it into dataframe and later we have dropped one feature article link which is unnecessary in this analysis. Then we have preprocessed the data in following steps:.

1. At first we have replaced all the non alphabatic letters through space of each records of the input feature using regular expression.
2. Then we have converted all letters into lower case letters and after that we tokenized a sentence word by word using split function.
3. Then we have applied portertemmer algorithm for word streaming.
4. After stemming we have joined all the stemmed word of a sentence again and saved it into a variable.
5. Then we have appended above variable into a corpus list which is a collection of stopwords.
6. At last we have converted corpus string into a numeric array through countvectorizer and saved in a variable x. Which is our bag of words model. We saved the output label into a variable y.

Then we have split all the data and labels into training and testing data and labels ina ratio of 80: 20. After that we have applied two different classification algorithm into training data which include decision tree and naive bayes and then a multilayer perceptron to the training dataset.

At first we have applied naive bayes with multinomial distribution in the training data and constructed a classifier then we have

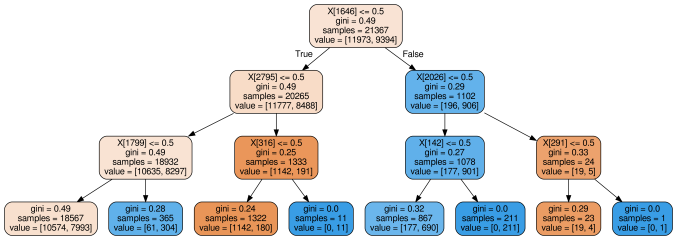


Figure 1: Constructed decision tree model

test the work of this classifier through test data. Then we have applied decision tree algorithm in training data. We have initialize the max-depth of the decision tree is 3 to reduce time and the complexity of memory allocation and we also have initialized random state 26 so that the training sequence of random sample cannot change after each iteration. After that we have test this constructed decision tree classifier through test data.

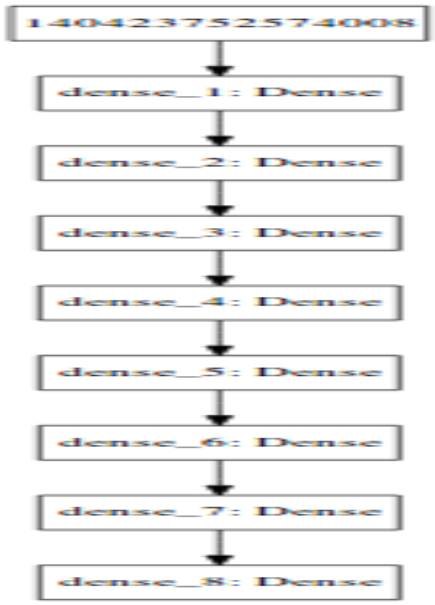


Figure 2: Constructed Multilayer Perceptron Model

At last we have applied multilayer perceptron with one input layer , one output layer and 6 hidden layers. We have initialized kernel initializers of all the layers into uniform so that random weight is generated using uniform distribution for our model. We have used rectified linear unit activation function in all the hidden layers as well as input layer and sigmoid activation function for the output layer since we have a binary class in output label. We have used adam optimizer for optimizing error in training of our model and binary cross entropy loss function for having a binary class.

4 RESULT AND ANALYSIS

To evaluate the performance of the classifiers applied in this work, we use primary performance indicators based on confusion matrix shown in following tables . This matrix contains information about real and predicted classifications carried out by the classification models. Performance metrics of such systems are commonly evaluated using the information in the matrix.

	Predicted Class	Predicted Class	Predicted Class
Actual Class	.	Yes	No
	Yes	TP=2365	FN=647
	No	FP=602	TN=1728

Table 1: Confusion Matrix of Naive Bayes classifier

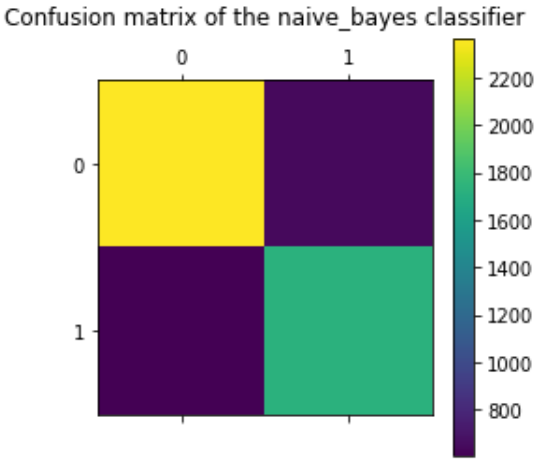
Accuracy = (TP +TN) / (TP + FN + FP + TN)
= (2365 +1728) / 5342
= 0.7661924372894046

Precision: It is the fraction of relevant instances among the retrieved instances. It is also called positive predictive.

Precision = TP / TP + FP
= 2365 / (2365 + 602)
= 0.80

Recall: It isthe fraction of relevant instances that have been retrieved over the total amount of relevant instances. It is also called positive sensitivity.

Recall = TP / TP + FN
= 2365 / (2365 + 647)
= 0.79

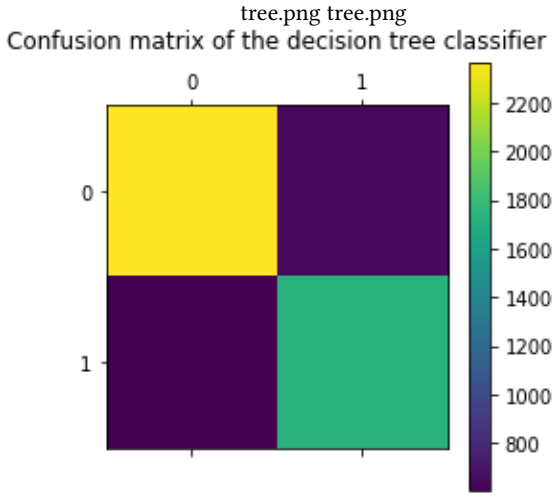


	Predicted Class	Predicted Class	Predicted Class
Actual Class	.	Yes	No
	Yes	TP=2365	FN=647
	No	FP=602	TN=1728

Table 2: Confusion Matrix of decision tree classifier

Accuracy = (TP +TN) / (TP + FN + FP + TN)
= (2365 +1728) / 5342
= 0.7661924372894046 Precision = TP / TP + FP
= 2365 / (2365 + 602)
= 0.80 Recall = TP / TP + FN
= 2365 / (2365 + 647)
= 0.79

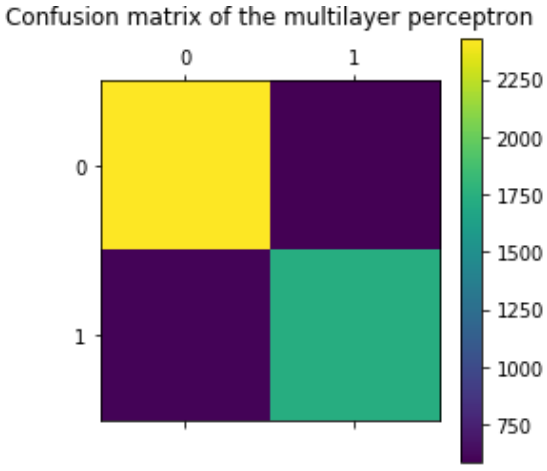
tree.png tree.png



	Predicted Class	Predicted Class	Predicted Class
Actual Class	.	Yes	No
	Yes	TP=2429	FN=583
	No	FP=599	TN=1731

Table 2: Confusion Matrix of mlp classifier

Accuracy = (TP +TN) / (TP + FN + FP + TN)
= (2429 +1731) / 5342
= 77.87345563459378 Precision = TP / TP + FP
= 2429/ (2429 + 599)
= 0.80 Recall = TP / TP + FN
= 2429 / (2429 + 583)
= 0.81



5 INTERPRETATION

In above case we have found same accuracy in testing data in case of decision tree and naive bayes which is 76.61 percent as well as same precision and recall 80 percent and 79 percent but when we have tested some samples collected randomly from this dataset both classifiers behave differently in same samples. We have chosen 5 samples for our testing purpose and have fed both the naive bayes and decision tree classifier and we observed that the first classifier misclassified only one sample whereas the second classifier misclassified in two samples. In this observation we conclude that though they have generated similar confusion matrix but their behavior on each tuple is different. The multilayer perceptron model has provided highest accuracy with highest recall 77.87 percent and 81 percent respectively comparing above two classifiers but it is time consuming and complex. We only selected 10 epochs along with batch size 50 and we have reached 99.71 percent training accuracy after running 10 epochs but the average elapsed time of running of each epoch was 30 seconds which means total 300 seconds or five minutes have been passed for training this model which is a huge amount of time in this type of study. The structure is complex since we have 6 hidden layers and we have implemented 3000 neurons in input neuron and reduce into one in the output layer of the output neuron. Hence we conclude that in case of classifying accurately MLP is the best classifier among all three classifiers but in case of less complexity and timing the Naive Bayes and decision tree are better than MLP. In comparing Naive Bayes with decision tree we cannot say which will be a better classifier.

6 CONCLUSION

: Since sentiment analysis is popular in recent years and this is a different kind of sentiment data rather than twitter data and also a better and clean data in comparing twitter data we have understand very much how sentiment analyzing works after working with this dataset. We have implemented three different data mining techniques and have achieved highest accuracy 77.87 percent and we want to better accuracy than this. ThatâĂŹs why in near future we want to apply some other data mining techniques or hybrid data mining techniques of above methods for achieve better accuracy.