News Classification

Aditya Mittal (MT18061)
Piyush Dhiyani (MT18131)
Vikash Kumar Pandey (MT18086)

Abstract

This report summarizes the work done for the project of news categorization. The report tells all the necessary steps taken by the members to complete the project work and also explains the techniques which are used in the project with proper results and conclusion at the end.

1 Introduction

News classification is a supervised learning task in which news articles are assigned to categories based on the training. This project focuses in the field of data mining. With the rapid growth of online text data, text categorization has become an essential technique for handling and dealing with this vast amount of data. To build a classifier manually is very difficult and time-consuming, so the better way is to make a classifier learn from examples.

In this project, we are using LDA (Linear Discriminant Allocation), Random Forest, SVM(Support Vector Machine) and Naïve Byes classifiers to classify text. After crawling the news data from the internet, preprocessing it then training the model with 80% of data and testing it on remaining 20% to check the accuracy to the trained model.

2 News Categorization

The aim of news categorisation is the classification of news document to the class to which it belongs. An article can be in multiple categories or none of the types. Using classifier the goal is to learn the classifiers from examples so that it can assign the class automatically. That's why it is being called supervised learning algorithm as a model is trained in supervision with the help of examples.

The first step in news categorisation is to collect the data set. The dataset is available on the internet can be downloaded, or we can also crawl the data from various websites using crawler methods in python.

The data is now to be converted to the suitable format suited for each of the classification algorithms which can be typically a list of characters or numbers.

The information which is collected is very noisy as it contains much garbage in the form of punctuations and digits which have no significance with our news, so now the step is to remove such noise from the data. After making the data clean, the next task is to remove the stop words from the data. Stop words are the words which occur very frequently in our data and doesn't have any significance with the context, for example, words like- is, an, the, you, here, there, etc..

Now to extract the features of the data, we have used a technique of TF-IDF (Term Frequency-Inverse Document Frequency). Features are the terms which represent a particular document, TF counts the frequency of a word occurring in a document divided by the total number of words in a document.

TF(t) = (no. of time t occur in a document)/(total no. of words in a document)

The words which are common in most of the documents are supposed to be not a useful feature. So IDF computes logarithmically scaled inverse fraction of total number of documents by the total number of documents containing that word

IDF(t) = log(total number of documents/total number of documents containing t)

The final value is TF(t) * IDF(t)

So with this, we have a final feature vector which is going to be the input of our classifier.

Before giving data to classifier we have to break our data into two parts, one part for training the model, i.e. 80% and another part for testing the accuracy of the model, i.e. 20%.

2.1 Support Vector Machine (SVM)

Support vector machine is a supervised machine learning algorithm which can be used for classification challenges. In this algorithm, we plot a graph of data points, and then we perform the algorithm by finding the hyperplane that can differentiate between two classes.

The algorithm finds out the many hyperplanes, among these hyperplanes the one which best sorts our data into two classes is selected as the right hyperplane. The right hyperplane is the one which segregates the two classes better. If some of the hyperplanes segregate the classes equally then find the margin distance of data points from the plane. The one which is at the most distance is used as the final right hyperplane.

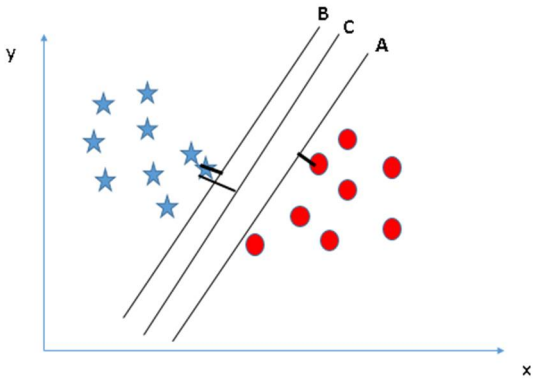

Fig. 1
https://www.analyticsvidhya.com/wp-content/uploads/2015/10/SVM_4.png

SVM is a binary classifier as it creates a hyperplane between data of two classes. Now the problem arises for classification with multiple classes, so for this purpose SVM has two methods which can be used to classify with multiple classes. One-to-one and one-to-rest. In the first method the resulting class is obtained by a majority of votes of all classifiers, and in the second method, you have the N classifier, and the resulting class will have the highest score.

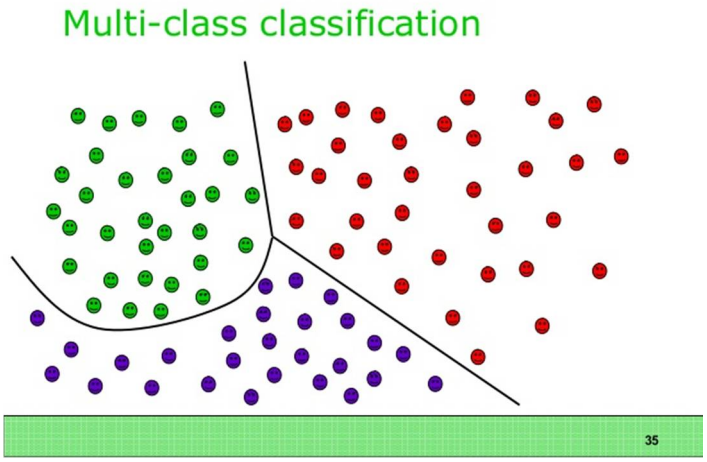

Fig. 2
https://uk.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/62061/versions/3/screenshot.jpg

2.2 Naïve Bayes

The reason for choosing Naive Bayes it performs well in case of categorical input variables compared to numerical.

Naive Bayes is easy and fast to predict the class of the test data set. And It also performs well in multi-class prediction.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c).

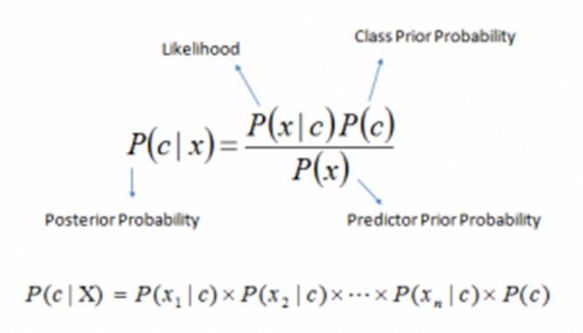

Fig. 2
https://www.analyticsvidhya.com/wp-content/uploads/2015/09/Bayes_rule-300x172-300x172.png

The problem with Naive Bayes is if the categorical variable has a category in test dataset, which not observed in training dataset, then the model will assign a 0 (zero) probability and will be unable to make a prediction. It will lead to decrease the accuracy of the model. That's the reason we didn't choose as our final classification model.

2.3 LDA(Linear Discriminant analysis)

Linear Discriminant Analysis is used to categorised or separates two or more classes. We use count vectorizer for features extraction and with the help of that, we classify the test data. We passed five sets of data with different categories each to our LDA model and passed a test article in the form of the string, and it classified with the probability of 0.7 to 0.8.

```
#######
Topics Discovered In Training data :

(0, '0.006*"compani" + 0.006*"mr" + 0.006*"firm" + 0.006*"market" +
0.005*"bank" + 0.004*"sale" + 0.004*"share" + 0.004*"price" +
0.004*"growth" + 0.004*"economi"')
(1, '0.007*"game" + 0.006*"play" + 0.006*"win" + 0.005*"t" + 0.005*"time" +
0.005*"player" + 0.005*"first" + 0.005*"england" + 0.005*"against" +
0.004*"back"')
(2, '0.013*"film" + 0.008*"best" + 0.007*"music" + 0.007*"award" +
0.006*"star" + 0.006*"show" + 0.004*"includ" + 0.003*"last" + 0.003*"first"
+ 0.003*"actor"')
(3, '0.009*"use" + 0.008*"peopl" + 0.008*"game" + 0.007*"can" +
0.005*"technolog" + 0.005*"mobil" + 0.005*"phone" + 0.004*"servic" +
0.004*"mr" + 0.004*"user"')
(4, '0.016*"mr" + 0.007*"govern" + 0.007*"labour" + 0.007*"elect" +
0.007*"parti" + 0.006*"say" + 0.006*"peopl" + 0.005*"blair" +
0.005*"minist" + 0.005*"tori"')
#######
```

Fig. 3

The figure shows the topics discovered in training set corresponding to given set of data. Then in discover the topic related to the detected topics for test data sets.

```
Detected Topics :


Topic :  (1, '0.007*"game" + 0.006*"play" + 0.006*"win" + 0.005*"t" +
0.005*"time" + 0.005*"player" + 0.005*"first" + 0.005*"england" +
0.005*"against" + 0.004*"back"')
Probability :  0.98995197
```

Fig. 4

The below graph shows the detected topics by LDA

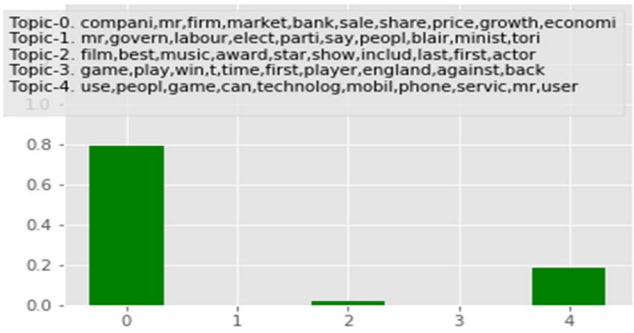

Fig. 5

Because we use count vectorizer the accuracy is not going great because if the file contains a features xi and other files have also the features xi than it not drop that features, and make confusion with that. That's the main reason for decreasing the accuracy of LDA.

2.4 Random Forests

With the help of Random forest, we first classify that news belongs to which category either positive or negative.
Then if the news is positive, then another model decide which type of positive news it is the same for negative.
For doing this we build three model for that, first one is determine either news is positive or negative by random forest it correctly(90%) determine. The second model is for if news is negative it correctly(91%) determine that and

third model is for positive news which correctly(97%) identify the news belong to which of the positive category.

With the help of the Random-Forest, we archive our first goal that is to tell the news is belong to either positive news or negative news on the basis of that we move the direction of analysis of classification done by different classifiers

4. TF-IDF Vs. Countvectorizer

In Countvectorizer each message is separated into tokens and the no. of times each token occurs in a message is counted.
In TF-IDF, TF count how often a word occurs in a document if we have several occurrences of the same word in one document we can expect the TF-IDF to increase.
IDF is representing how common a word is across documents. If a word used in many documents then the TF-IDF will decrease which is a good sign for features selection but Countvectorizer is not handling that's why TF-IDF work better with our classifiers.

4. Results

We use LDA with the feature extractor of count vectorizer and successfully predict with the accuracy of 70 per cent. Because LDA is not performing well, so we use Naive Bayes algorithm as it performs well in multiclass prediction. So we used count vectorizer as feature extractor and Naive Bayes as a classifier, and it gives an accuracy of 89 per cent.

Count vectorizer does not eliminate those features which are common in all the documents in the data set. We consider this as the reason for not performing well. So we applied TF-IDF as a feature extractor with our Naive Bayes classifier to get the better output.

The following tables show the comparison between the combination of the different classifier with different feature extraction methods.

S.No	Classifier	Features Extractor	Accuracy
1	Naive Bayes	TF-IDF	94%
2	Naive Bayes	Count vectoriser	89%
3	Multiclass SVM	TF-IDF	95%
4	Multiclass SVM	Count Vectoriser	90%
5	LDA	Count Vectoriser	70%

The above table shows the comparison with the different classifiers and results clearly shows that Multiclass SVM with the help of TF-IDF give the highest accuracy.
We select the train set, and test set randomly, so we run our model number of times and determine that the accuracy hardly changes.
The model of random forest also give the satisfied accuracy to predict the class is either negative or positive which ensure to classify it further, and we see that result is getting better with that.

5. Conclusion

To conclusion of the project is we used various classifiers and various feature extraction techniques and the best result for out data set is with the accuracy of 95% i.e. by using support vector machine on multi classes with the help of TF-IDF feature extraction.

6. References

- Personalized News Categorization Through Scalable Text Classification.By Ioannis Antonellis, Christos Bouras, and Vassilis Poulopoulos Research Academic Computer Technology Institute N. Kazantzaki, University Campus, GR-26500 Patras, Greece.
- Text categorization with SVM. By Thorsten Joachims Universität, Dortmund informatik LS 8, Baroper Str. 301.
- Automatic text classification By Ahmed Kachkach.
- Vladimir N. Vapnik. *The Nature Statistical Learning Theory*. Springer, New York, 1995.
- Y. Yang. An evaluation of statistical approaches to text categorization. Technical
- Report CMU-CS-97-127, Carnegie Mellon University, April 1997.
- Y. Yang mid J. Pedersen. A comparative study on feature selection in text categorization.In *International Conference on Machine Learning (ICML)*, 1997.
- D. M. Blei, J. D. Lafferty. Correlated topic models. *Advances in Neural Information Processing Systems*; 2006, p. l47-154.
- B. Walsh. Markov chain monte carlo and gibbs sampling. *Lecture Notes for EEB 581*, version 26; 2004.
- Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent*
- *Systems and Technology*, Vol 2. No.3. NY, USA, ACM; 2011.