Supervised learning using Word Vectors

Author:
Oindrilla Chatterjee

Email ID:
oc@bu.edu

December 24, 2018

Contents

1 Abstract

In this project I performed analysis on on a large dataset of movie reviews from IMDB to predict whether a movie review is positive or negative. The data is split into training and testing sets. I have used a Random Forest classifier to fit the "Bag of Words" model. This model is further used to predict the sentiment label of movie reviews in the test dataset. I have understood how learning word vectors could play an important role in predictions using supervised learning in a text corpus.

2 Motivation

People express their emotions in language that is often obscured by sarcasm, ambiguity, and plays on words, all of which could be very misleading for both humans and computers. Sentiment analysis systems are being applied in almost every business and social domain because opinions are central to almost all human activities and are key influencers of our behaviors.[4] Our beliefs and perceptions of reality, and the choices we make, are largely conditioned on how others see and evaluate the world. That is was my motivation to choose this project.

3 DataSet

I have obtained this data set from Kaggle[1]. The labeled data set consists of 25,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary, meaning the IMDB rating less than 5 results in a sentiment score of 0, and rating greater than or equal to 7 have a sentiment score of 1. No individual movie has more than 30 reviews.

3.1 File descriptions

It is a labeled data set. The file is tab-delimited and has a header row followed by 25,000 rows containing an id, sentiment, and text for each review.

3.2 Data fields

- id - Unique ID of each review

- sentiment - Sentiment of the review; 1 for positive reviews and 0 for negative reviews
- review - Text of the review

4 Research

4.1 Feature Extraction

Text Analysis is a major application field for machine learning algorithms. The raw data, a sequence of symbols such as the movie reviews here cannot be fed directly to the algorithms. The algorithms expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. [2] A corpus of documents can thus be represented by a matrix with one row for each document and one column for each token occurring in the corpus.
Vectorization is a general process of turning a collection of text documents into numerical feature vectors. This specific strategy (tokenization, counting and normalization) is called the Bag of Words representation. Bag of words is a traditional approach to represent the presence or absence of a word in an observation(sentence or document). Documents are described by word occurrences while completely ignoring the relative position information of the words in the document. Such encodings often provide sufficient baselines for simple NLP tasks.

4.2 Classification

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size.[3]

4.3 Feature Reduction

In classification there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This

is where feature reduction algorithms come into play. Feature reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables.
In this project, I tried experimenting with some mechanisms for dimension reduction. After fitting the model using a random forest classifier, I gathered the feature importances from the trained tree based on the "gini importance" or "mean decrease impurity" which is given by the total decrease in node impurity. The model was retrained after reducing the least important features to obtain a higher classification accuracy.

5 Methodology

5.1 Pre-processing

I used Apache Spark(pyspark) to preprocess the data because of the ease of performing the same steps for every record in the dataset using an RDD.

Steps that I have done to preprocess the data -

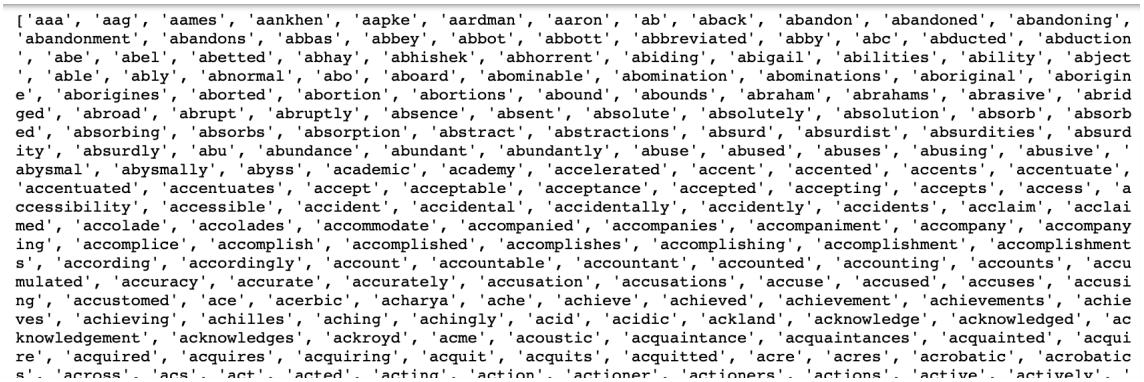
- Remove HTML tags using BeautifulSoup
- Remove numeric characters and punctuations from the reviews
- Remove the stop words from the reviews
- Lemmatization of words

5.2 Feature Extraction

I used sci-kit learn's CountVectorizer to convert a collection of text documents to feature vectors having the top 25000 words as features for a bag of words model.

This can be done by assigning each word a unique number. Then the document is encoded as a fixed-length vector with the length of the vocabulary of known words.

In this bag of words model we are only concerned with encoding schemes that represent what words are present or the degree to which they are present in encoded documents without any information about order.



The above image is a snapshot of the first few bag-of-words collected using the Count Vectorizer.

The fit function fits the model and learns the vocabulary; second, it transforms our training data into feature vectors. The input to fit transform is a list of movie reviews.

5.3 Classification

5.3.1 Train and Test Split

After I extract the features out of the reviews from the dataset we need to split it into training and testing sets. The ratio for the division is set to 80:20 for training and testing respectively. Split arrays or matrices into random train and test subsets.

Sci-kit learn's train_test_split is a quick utility that wraps data into a single call for splitting data in a oneliner.

5.3.2 Random Forest Classifier

This algorithm uses a decision tree approach and can be used for both classification and regression tasks. Overfitting is one critical problem that may make the results worse, but for Random Forest algorithm, if there are enough trees in the forest, the classifier won't overfit the model. Random Forest can handle missing values and can also be modeled for categorical values.

I used sci-kit learn's RandomForestClassifier and fit the training data onto the model. Then I predicted the sentiment values for the test dataset using the trained model.

5.4 Feature Reduction

After fitting my model on the selected 25000 features, I wanted to analyze the impact of reducing the dimension by reducing the number of features and retraining the model on the reduced features.

5.4.1 Feature Importances

Random forest models are often used to determine feature importances and rank features by order of importance. Sklearn's Random forest classifier lets us compute the feature importances which suggest how important certain features are.

5.4.2 Model on Reduced Features

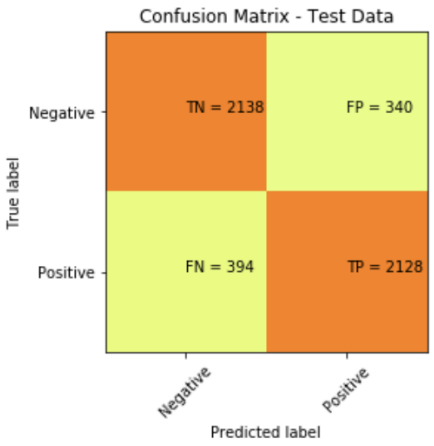
Based on the returned weights of feature importances, I selected the top 20000 features and reduced the features corresponding to the least important 5000 features from the training and test set and then refitted the model and the classification accuracy to see if the classfier perfomed better.

6 Results

Testing the model on the dataset with 25000 features yields about 85.3% accuracy. The figure below shows the precision, recall, f1-score values on the test data.

	precision	recall	f1-score	support
0	0.84	0.86	0.85	2478
1	0.86	0.84	0.85	2522
avg / total	0.85	0.85	0.85	5000

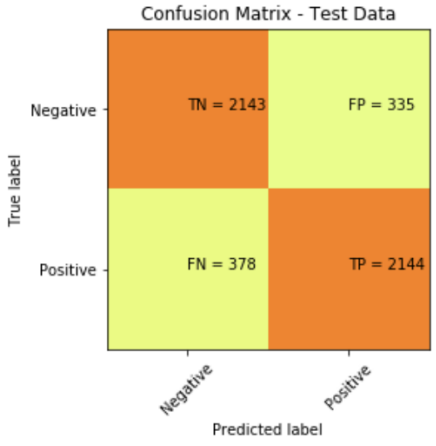
The diagram below shows the confusion matrix obtained.



After reducing the dimension and bringing down the features to 20000, the model yields about 86% accuracy. The figure below shows the precision, recall, f1-score values on the test data.

	precision	recall	f1-score	support
0	0.85	0.86	0.86	2478
1	0.86	0.85	0.86	2522
avg / total	0.86	0.86	0.86	5000

The diagram below shows the confusion matrix obtained.



7 Conclusion

Accurate classification and prediction of sentiments in sentences is of high importance in the recent times. As far as movie reviews are concerned, gathering an overall sentiment score of reviews and being able to classify new movie reviews based on a past trained set could turn out to be important as it directly translates to determining what consumers like and what kinds of content sells. Here, using the Random Forest Classifier model, I am able to classify the review sentiments as positive or negative by fitting it on the IMDB dataset on a selected 25000 features. I further perform feature reduction by selecting only the top 20000 ranked important features. The model performs better after the dimension reduction process. The future scope of this project would be to implement deep learning models like recurrent neural networks and LSTMs to capture information about sentence semantics and sequence of wordings.

References

[1] Kaggle Bag of Words. https://www.kaggle.com/c/word2vec-nlp-tutorial/data Accessed 2018-12-13

[2] Count Vectorizer www.scikit-learn.org Accessed 2018-12-13

[3] Random Forest https://scikit-learn.org/stable/modules/generated/ sklearn.ensemble.RandomForestClassifier.html Accessed 2018-12-14

[4] Bing Liu, 2012, Sentiment Analysis and Opinion Mining,