Capstone Project

Ahmed Kaboud

Machine Learning Engineer Nanodegree

16/11/2018

Movies Reviews Analysis

I. Definition

Project Overview

Opinion mining and sentiment analysis is the computational study of people's opinions, attitudes and emotions toward an entity. It is rapidly growing area. This is very much helpful to both the individuals, who are willing to buy that product and the organizations which want to improve and market their products. There are many types of product. The product can be electronic device, song, movie, picture...etc.

Opinion mining is one of the most active research areas in Natural Language Processing. With the growth of Social Media, there is a need to know people's opinions. So, here Opinion mining play an important role by analyzing and clustering the user generated data like reviews, blogs, comments, articles etc. which can help us to understand people attitudes and know if it is a positive feeling or negative feeling.

There are different types of algorithms to analyze sentiments. Some these techniques are documented at the survey in References. These types divided into Machine Learning Approach which can be Unsupervised Algorithms and supervised Algorithms like Decision Tree, SVM, Neural Network, and Naïve Bayes. The other approach is Lexicon Based Approach which is like Dictionary Based Approach.

The key advantage of supervised learning for sentiment classification is that it can automatically learn from all kinds of features for classification through optimization. Most of these features are difficult to use by a lexicon-based method. Supervised learning depends on the training data, which needs to be manually labeled for each domain. But these classifiers trained from the labeled data in one domain often do not work in another domain and also it is hard to learn things that do not occur frequently.

Problem Statement

I used sentiment analysis in analyzing movies reviews. There are many movies in the internet or cinemas. A lot of people can't decide which movie to watch. The trivial solution to decide which movie to watch is to search manually for any website which provides reviews for every movie. Absolutely, with growing up the number of websites which provide these reviews, the mission to search and decide become more complex.

Here machine learning play an important role to facilitate the difficult task to decide which movie you should watch. Sentiment analysis uses machine learning to analyze the reviews of people for many movies. So, anyone can find an overall feedback of the movie whatever positive or negative. It not only classify the movie reviews not in positive or negative But also can give a stochastic rating like average, good, very good and excellent. So, it reduces the time to decide which movie to watch.

I will perform sentiment analysis on IMDB reviews. I took this dataset from Kagle Competition. I use the reviews from this dataset to train the model to detect this is a positive movie or negative one. I will apply different supervised learning models which will predict the polar of each movie. I will start with Naïve Bayes because it does well in text classification problem. Then, I will compare between the accuracy of each model.

Metrics

For each model, I will use Receiver Operating Characteristics [ROC] to compute Area Under Curve [AUC]. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm and can be calculated as (1 − specificity). The area under the curve is a measure of accuracy.

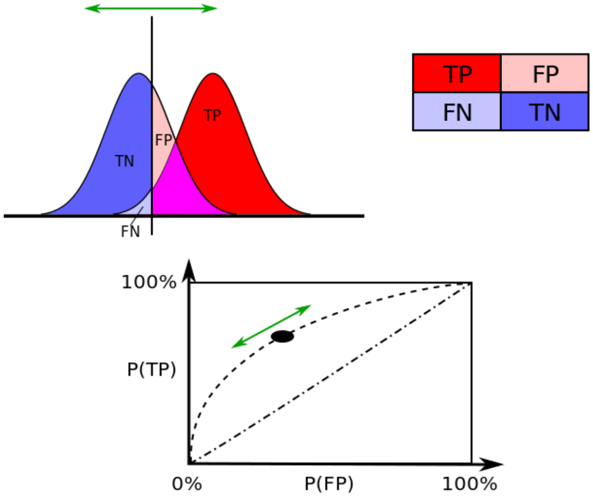


Fig. 1 Curves in ROC Space

The ROC is used because it illustrates the performance for any **binary classification problem** which used binary classifier like Naïve Bayes or Logistic Regression. It has a threshold value which discriminates between positive and negative reviews.

II. Analysis

Data Exploration

The datasets which I used here is come from Kaggle competition **Bag of Words Meets Bags of Popcorn**. The competition was based on using machine learning in sentiment analysis. They tried to analyze the Movies review through 2 datasets.

The data set consists of 50,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary and that is meaning that the IMDB rating < 5 results in a sentiment score of 0, and rating >=7 have a sentiment score of 1. No individual movie has more than 30 reviews. There are 2 Datasets one labeled and another unlabeled. I used the labeled one which has 25,000 different reviews.

The dataset which i used exists in (**labeledTrainData.tsv**). It is tab-delimited and has a header row followed by 25,000 rows containing an id, sentiment, and text for each review.

- **ID:** integer represents the key of every row in the dataset.
- **Sentiment:** it the label of film reviews. It has Binary value which be 0 for negative or 1 for positive.
- **Review:** it contains the reviews of every movie which we will use to train and build the model.

I didn't use "**Id**" feature. Also, I Shuffled and split all data into Training and Testing Data with ration 80:20 respectively. I preprocessed these data before using it to clean it and make a bag of word to be used in classification. The processed data is balanced which mean the positive equal the negative data. There are 12,500 positive reviews and the same number for negative reviews. So the positive and negative reviews are equally sized. And there is no correlation between the number of words in review and sentiment.

Data Visualization

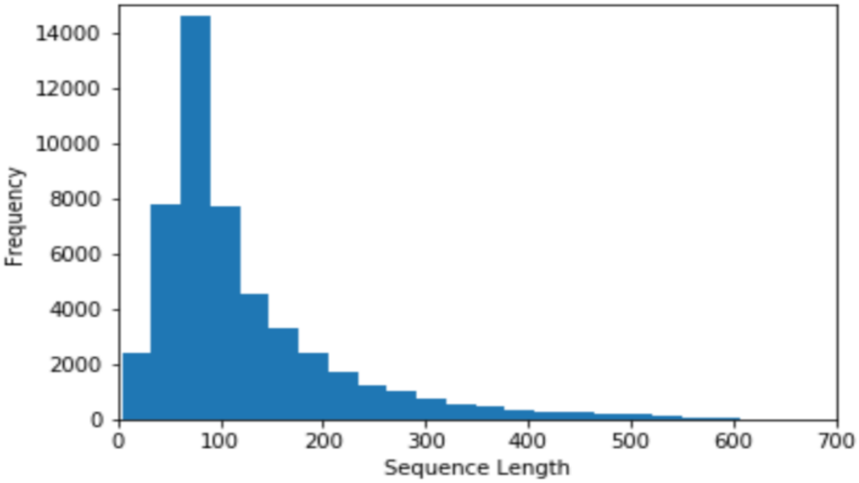


Fig. 2 Distribution of word count across the reviews

The above histogram represents the distribution of word count across the reviews. On the X-axis, it represents the distribution of the number of words. On the Y-axis, it represents how many each review falls in that category.

From the chart, the most reviews fall under 250 words. In addition, this chart was prepared after cleaning the reviews from punctuation, special chars and stop words

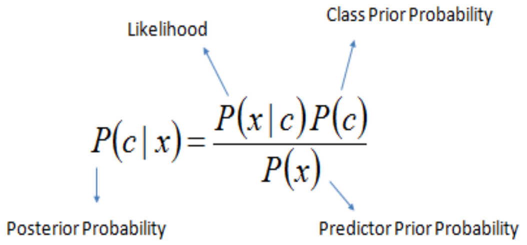
Algorithms and Technique

The problem is binary classification problem. So I will build a model that tries to classify the reviews into 1/positive or 0/negative. I will try to take the advantage of labeled data and use supervised learning algorithm to predict the sentiment of the reviews. I will try different classifier and build different model.

Naïve Bayes

Naïve Bayes extends bayes' theorem with strong (naive) independence assumptions between the features. This is saying "the probability that classification **c** is correct given the features **X**, and so on equals the probability of **c** times the product of each **x** feature given **c**, divided by the probability of the x features".

To find the "right" classification, we just find out which classification (P(c|x1,...,xn)P(c|x1,...,xn)) has the highest probability with the formula.


Likelihood
Class Prior Probability
$P(c\mid x)=\frac{P(x\mid c)P(c)}{P(x)}$
Posterior Probability
Predictor Prior Probability

$$P(c\mid \mathrm{X}) = P(x_1\mid c)\times P(x_2\mid c)\times\cdots\times P(x_n\mid c)\times P(c)$$

Fig. 3 Naïve Bayes Formula

Logistic Regression

It is a supervised binary classification technique. It is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Logistic regression is fast in training and inference and performs well in wide variety of cases.

Let

$$\mathrm{X} = (X_1, X_2, \cdots, X_p)$$

$$B_g = \begin{pmatrix}\beta_{g1}\\ \vdots \\ \beta_{gp}\end{pmatrix}$$

The logistic regression model is given by the G equations

$$\ln\left(\frac{p_g}{p_1}\right) = \ln\left(\frac{P_g}{P_1}\right) + \beta_{g1}X_1 + \beta_{g2}X_2 + \cdots + \beta_{gp}X_p$$

$$= \ln\left(\frac{P_g}{P_1}\right) + \mathrm{XB}_g$$

Here, p_g is the probability that an individual with values $X_1, X_2, \ldots, X_p$ is in outcome g. That is,

$$p_g = \Pr(Y = g\mid \mathrm{X})$$

$$\mathrm{p_g} = \mathrm{Prob}(Y = g\mid \mathrm{X}) = \frac{e^{\mathrm{XB}_g}}{1 + e^{\mathrm{XB}_2} + e^{\mathrm{XB}_3} + \cdots + e^{\mathrm{XB}_G}}$$

Fig. 4 Logistic Regression Formula

Benchmark

Naïve Bayes will be the benchmark model. Firstly, I built a naïve bayes model using training dataset then get the ROC AUC score for testing dataset. It gave me 85%.

I tried to build a logistic Regression model also and get the ROC AUC score for testing dataset. It gave me more than 85%.

III. Methodology

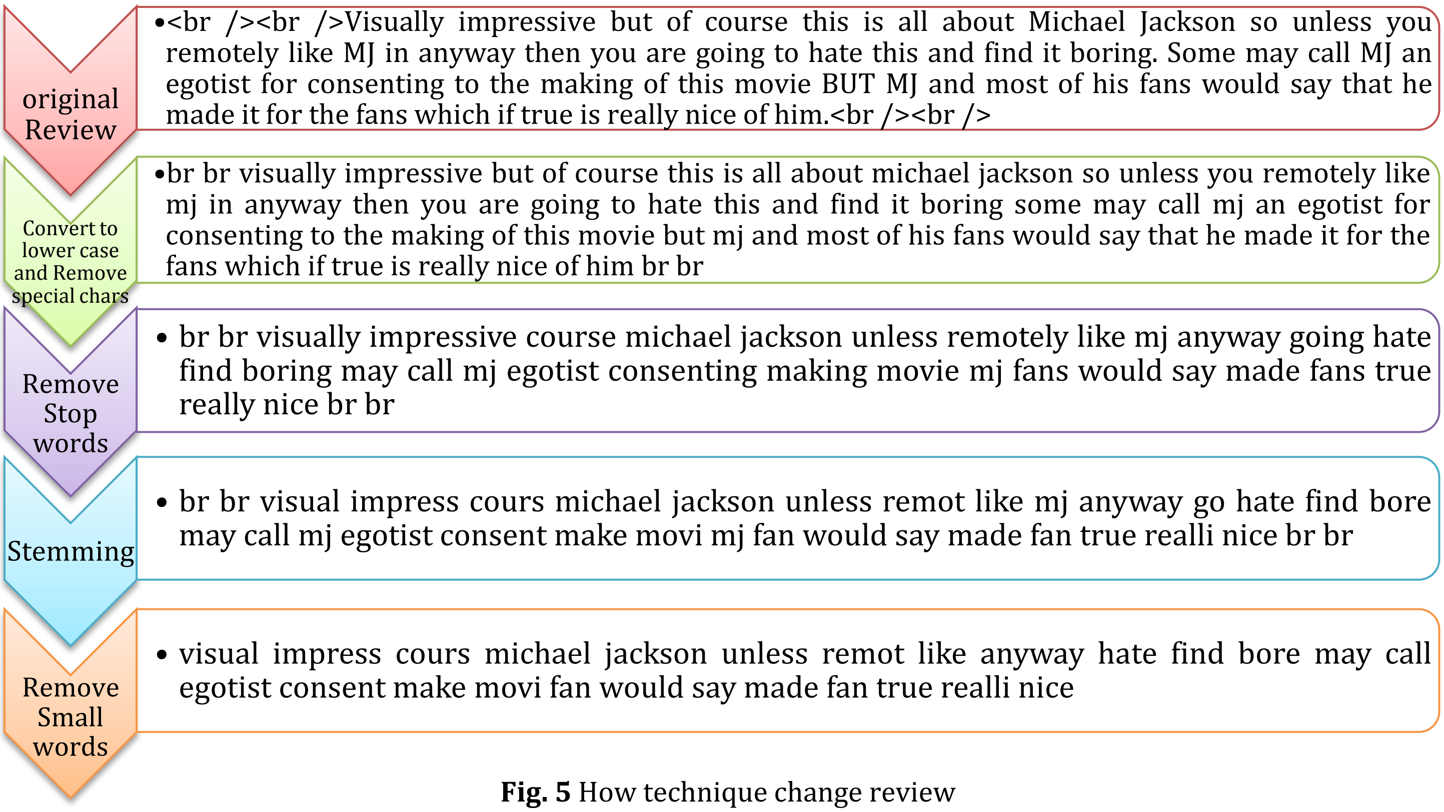
Reading the data

We will remove the delimiters and put the data from the file into 2 arrays. One for review text and another for sentiment label which will be 0 or 1.

Data Preprocessing and Cleaning

There are 5 steps to preprocess and clean all data:

- Convert all the reviews characters into lowercase characters.
- Remove all non-alphabetic characters with space like HTML tags, punctuation and numbers
- Remove English stop words using NLTK.
- Stemming for every word in the review.
- Remove small words which have siz e less than 2 characters.



- <br /><br />Visually impressive but of course this is all about Michael Jackson so unless you remotely like MJ in anyway then you are going to hate this and find it boring. Some may call MJ an egotist for consenting to the making of this movie BUT MJ and most of his fans would say that he made it for the fans which if true is really nice of him.<br /><br />
- br br visually impressive but of course this is all about michael jackson so unless you remotely like mj in anyway then you are going to hate this and find it boring some may call mj an egotist for consenting to the making of this movie but mj and most of his fans would say that he made it for the fans which if true is really nice of him br br
- br br visually impressive course michael jackson unless remotely like mj anyway going hate find boring may call mj egotist consenting making movie mj fans would say made fans true really nice br br
- br br visual impress cours michael jackson unless remot like mj anyway go hate find bore may call mj egotist consent make movi mj fan would say made fan true realli nice br br
- visual impress cours michael jackson unless remot like anyway hate find bore may call egotist consent make movi fan would say made fan true realli nice

Fig. 5 How technique change review

Implementation

The 2 steps to prepare the date for training the model:

Shuffle and Split Data

I shuffled the data to sort them in random way not in specific sequence. I split the data into 80:20 ratios for training and testing respectively.

Creating Features from a Bag of Words

I converted the reviews text to some kind of numeric representation for machine learning by **Bag of Words**. It models each document by counting the number of times each word appears. Also, I can try to explore pre-trained word vector if it quicken the modeling step. This will make the features ready to train the models.

Binary Classification Models

I tried 4 binary classification models using scikit-learn. They are **MultinomialNB, Support Vector Machine, Logistic Regression and Decision Tree Classifier.** The metric used to evaluate each classifier Area under the ROC as explained before.

Naïve Bayes Approach makes a very strong assumption on the shape of your data distribution, i.e. any two features are independent given the output class. Due to this, the result can be potentially very bad - hence, a "naive" classifier. This is not as terrible as people generally think, because the NB classifier can be optimal even if the assumption is violated, and its results can be good even in the case of sub-optimality. Naïve Bayes doesn't take into account relative order of words, it just go by the absence of words and their frequencies.

Logistic Regression is very efficient, does not require too many computational resources, it's highly interpretable, it doesn't require input features to be scaled, it doesn't require any tuning, it's easy to regularize, and it outputs well-calibrated predicted probabilities.

All he above code for data preprocessing, model implementation and evaluation exist in a notebook Capstone-Project.ipynb. You should able to run each classifier with needed library. Make sure that the data set at the right place.

Refinement

For **MultinomialNB Classifier,** I improved the accuracy by changing the value of **alpha** (Additive smoothing parameter). Maximum AUC ROC score= 86.1385% when alpha = 1.6. So, the score increased by 0.3%

For **Logistic Regression Classifier**, The AUC ROC Score increased using **C.** The best value for C =0.031, which make the score 88%. So, it increased by 1.4%.

IV. Result

Model Evaluation and Validation

Performance as notebook attached is:

- Naïve Bayes Classifier: 86.1385%
- Logistic Regression Classifier: 88.00%
- Support Vector Machine Classifier: 74.57%
- Decision Tree Classifier: 71.00%

The model which produces the highest score is Logistic Regression model. It gives 88% when C = 0.031. Although Logistic Regression gives a higher accuracy than Naïve Bayes, It is a bit slower than Naïve Bayes. SVM and Decision Tree have a bad performance in the time to build the model and score for AUC of ROC.

Model Details:

- Convert all the reviews characters into lowercase characters.
- Remove all non-alphabetic characters with space using the special character array.
- Remove English stop words
- Make a stemming for all reviews
- Remove small words which have size less than 2 character
- Shuffle and split the data 80:20 for training and testing respectively.
- Create features from bag of words
- Build Logistic regression model with C = 0.031
- Try to predict the test data.
- Calculate the score using AUC ROC.

The model generalizes the data well. With changing the parameters of classifiers, it won't result a major deviation in the result. But with misspellings that would affect the results. So the answer will depend on the number of misspelling words. If there is small number of misspelling words, it won't matter. But with a huge number of misspellings, it will be a problem.

Justification

Logistic regression and naïve bayes Proved efficiency in Text binary classification problems. But it isn't enough to solve the problem with excellent accuracy. If I use CNN and build a model that take the output of Logistic Regression model as an input. It will improve the score to be more than 90%.

V. Conclusion

Free-Form Visualization

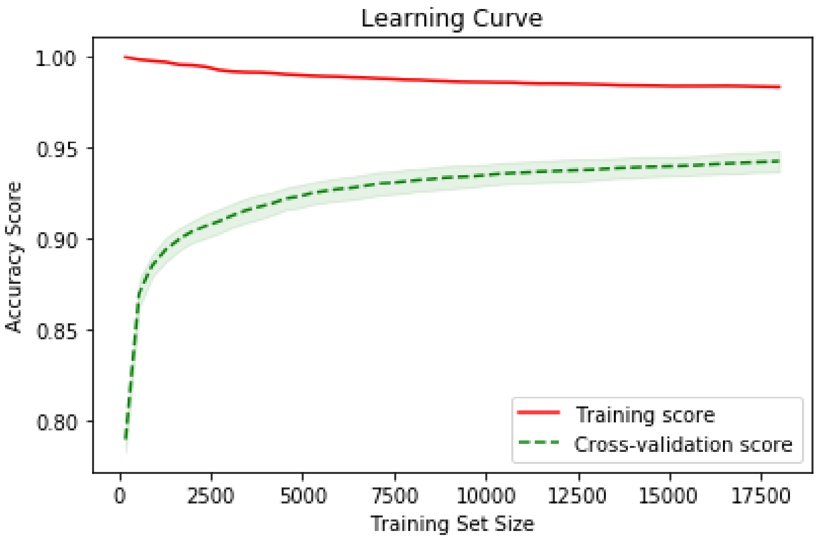


Fig. 6 Learning rate Curve of Logistic Regression Model with mean feature vector

This learning curve is the result of building Logistic Regression Model with AUC-ROC metric and number of Folds in cross validation = 10.

Reflection

I enjoyed working on this project due to many things. The first of them that this is the first project complete project i worked on and make data preprocessing and investigating and search for every part in it. I studied sentiment analysis before but without practice experience. So this adds to me a strong experience in investigating and practicing this field.

One of the interesting parts of the Capstone project is to decide your own evaluation criteria. That made me to think through the problem and recommend using Naive Bayes and Logistic Regression and use them as a benchmark the accuracy.

This is the first time to use kagle. I enjoyed reading a lot of competitions and problems in this site. I interested to do Movies Review as my capstone project and will practice others in the next periods.

The solution starts with reading the data and split the delimiters and save the needed features and labels in different arrays. Then preprocess/cleaning step on these features by converting all characters to lower case, remove special characters, remove stop words, Stemmer for the text reviews and finally, remove the short words which have size less than 2 characters. Then, create the features from a bag of word by converting the reviews text to some kind of numeric representation. Then, this is the step of building classifier.

Naïve Bayes is pretty well for this problem. It is Simple and speed in performance. It gave me a good accuracy in a few seconds. So its accuracy exceeded 86%.

Also, Moving from Naïve Bayes to use Logistic Regression which improves the AUC ROC score more than I expected. It reached to be 88% which increased by 2% larger than Naïve Bayes.

Building different models and comparing between the times, scores and performance for each one is an interesting stage for me.

Deciding the performance based on Receiver Operating Characteristics [ROC] to compute Area Under Curve [AUC] is fine for this problem.

Improvement

I can use Neural Network and use pre-trained model to train this Network. It will improve the performance for this model. Also, Making this Network deeper and use dropout to make sure all parts of Network trained well will increase the performance. Also, I can use Ensemble method to improve the performance. It will produce a competitive model.

Naïve Bayes RNN will improve the performance because it takes into account the meaning of words by taking into account its neighboring words. Also it takes into account the order of words Not like Naïve Bayes which doesn't take into account the order of words; it just goes by its frequencies.

VI. References

Angela Chapman, Competition: Bag of Words Meets Bags of Popcorn, "https://www.kaggle.com/c/word2vec-nlp-tutorial", summer 2014 internship at Kaggle.

Vidisha M. Pradhan, Jay Vala and Prem Balani, "A Survey on Sentiment Analysis Algorithms for Opinion Mining", International Journal of Computer Applications, 2016.

Lei Zhang and Bing Liu, "Sentiment Analysis and Opinion Mining", University of Illinois at Chicago, Chicago, IL, USA, 2016.

Vik Paruchuri, "Naive bayes: Predicting movie review sentiment", "https://www.dataquest.io/blog/naive-bayes-tutorial", 16 MARCH 2015

NCSS Statistical Software, "Logistic Regression", chapter 321.

Niklas Donges,The Logistic Regression Algorithm, "https://machinelearning-blog.com/2018/04/23/logistic-regression-101/", 2018

Wikipedia contributors, "Stemming", Wikipedia.

Wikipedia contributors, "Receiver operating characteristic", Wikipedia.

Wikipedia contributors, "Naïve Bayes classifier", Wikipedia.

Plot Learning Curve, "https://chrisalbon.com/machine_learning/model_evaluation/plot_the_learning_curve/ "