Sentiment Analysis of Restaurant Reviews

Kurmapu Satya Prasad (1001722452), Veena Ahir (1001771342)
Nithya Balasundar (1001763661)

Abstract

There has been steep rise in the number of blogs, reviews, comments, social network portals and web sites on the internet. Thus, analyzing this text data for determining the success of a business, customer trends, marketing, advertisement campaigns, spam filtering is important.

The key aspect of sentiment analysis is to analyze a body of text for understanding the opinion expressed by it. Nowadays, the process of gathering the data for this research can be not only “push-based”, but also “pull-based”, as users can freely and publicly express their opinions in the online medium through reviews. However, it is hard to make sense of such a large quantity of unstructured information. But this is where natural language processing and machine learning come in hand. It allows machines to understand how human speaks.

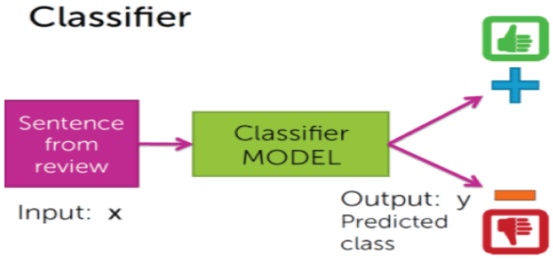
In this project, we focus on Sentiment Analysis that helps us analyze people's sentiments, emotions, and evaluations from written language. The implementation of NLP models to identify a given set of words as positive sentiment or negative sentiment.

Introduction

Several analysis tools of data mining (like, clustering, classification, regression etc.) can be used for sentiment analysis task. Sentiment mining is one of the important aspects of data mining where important data can be mined based on the positive or negative senses of the collected data. Sentiment Analysis also known as Opinion Mining refers to the use of natural language processing, text analysis and computational linguistic to identify and extract subjective information in source materials. Here the source materials refer to opinions / reviews /comments given in various social networking sites.

The Sentiment found within comments, feedback or critiques provide useful indicators for many different purposes and can be categorized by polarity. By polarity we tend to find out if a review is overall a positive one or a negative one.

For example: 1) Positive Sentiment in subjective sentence: “I loved the food”-This sentence expresses positive sentiment about a particular restaurant and we can decide that from the sentiment threshold value of word “loved”. So, threshold value of word “loved” has positive numerical threshold value. 2) Negative sentiment in subjective sentences: “I hated the ambience there”- This sentence expresses negative sentiment about a particular restaurant, and we can decide that from the sentiment threshold value of word “hated”. So, threshold value of word “hated” has negative numerical threshold value. 3) Similar we have included suggestive opinion. “I wish they had Thai curry on their menu as well” -This sentence expresses suggestive sentiment about a particular restaurant, and we can decide that from the sentiment threshold value of words “I wish”. So, threshold value of words “I wish” has a suggestive numerical threshold value.



We have also implemented fine grained sentiment analysis. Fine-grained sentiment analysis tries to predict sentiment in a text using a finer scale namely positive, very positive, neural, somewhat negative, negative and suggestive. This data can be used by the restaurants for analyzing their business and make better future decisions based on the reviews received.

By using Natural Language Processing, we will make the computer truly understand more than just the objective definitions of the words. This analysis will help us segregate the data that has good as well as bad movie reviews. It includes using Bag of words model which is a way of extracting features from the text for use in modeling. Not only that but also using Classifier module to identify whether a given piece of text is positive or negative. In this case, we are using Naïve Bayes as our classifier using decision trees. The scikit library will help the algorithm learn with a faster curve, helper class to clean our data, Pandas will help us read our CSV files and NTLK removes unnecessary data from the dataset.

Dataset Description

The dataset used for this task was collected from Large Movie Review Dataset which was used by the AI department of Stanford University for the associated publication. Two important conditions are considered while computing the sentiments: 1. A negative review has a score ≤ 4 out of 10, 2. A positive review has a score ≥ 7 out of 10. The dataset is divided into two sets, namely training and test sets. The training set has 25,000 labeled reviews used to induce word vectors. The 25,000-review labeled training set does not include any of the same movies as the 25,000-review test set. In addition, there are another 50,000 IMDB reviews provided without any rating labels.

As sentiments are usually bipolar like good/bad or happy/sad or like/dislike, we categorized these ratings as positive (8-10), somewhat positive (6-7), neural (5), somewhat negative (3-4), negative (0-2) based on the ratings. If the rating was above 6, we deduced that the person liked the movie otherwise he did not. For suggestive we used the suggestive.csv dataset.

A typical review text looks like this:

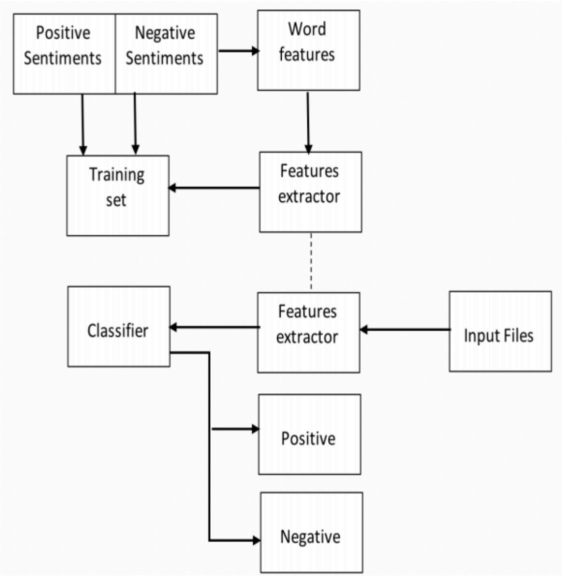
“I'm a fan of TV movies in general and this was one of the good ones. The cast performances throughout were pretty solid and there were twists I didn't see coming before each commercial. To me it was kind of like Medium meets CSI. <br/><br/> Did anyone else think that in certain lights, the daughter looked like a young Nicole Kidman? <br/><br/> Are they related in any way? I'd definitely watch it agin or rent it if it ever comes to video. <br/><br/> Dedee was great. Haven't seen in her in a lot of things and she did her job very convincingly. <br/><br/> If you're into to TV mystery movies, check this one out if you have a chance.”

As seen above, one necessary pre-processing step prior to feature extraction was removal of HTML tags like “<br/>”. We used simple regular expressions matching to remove these HTML tags from the text. Another important step was to make the text case insensitive as that would help us count the word occurrences across all reviews and prune unimportant words. We also removed all the punctuation marks like ‘!’, ‘?’, etc. as they do not provide any substantial information and are used by different people with varying connotations. This was achieved using standard python libraries for text and string manipulation. We also removed stopwords from the text for some of our feature extraction tasks. Stemming and Lemmatization could both be used. One important point to note is that we did not use stemming of words as some information is lost while stemming a word to its root form. Instead, we used lemmatization. Lemmatization removes the unwanted characters from the words and only keep the lemma without changing the meaning of the word.

Project Description

1. Description

Classification algorithm predicts the label for a given input sentence. There are two main approaches for classification: supervised and unsupervised. In supervised classification, the classifier is trained on a labeled example that are similar to the test examples. Contrary, unsupervised learning techniques assign labels based only on internal differences (distances) between the data points.



The above architecture diagram represents the process that takes place throughout the Sentiment Analysis process.

We used 3 methods for extraction of meaningful features from the review text which could be used for training purposes:

Bag of Words: We first calculated the total word counts for each word across all the reviews and then used this data to create different feature representations.

N-Gram Modelling: Bag of Words ignores the semantic context of the review and concentrates primarily on frequency of each word. To overcome that, we also tried n-gram modelling wherein we created unigrams, bigrams and mixture of both.

TF-IDF Modelling: While the two methods of feature extraction descried above concentrated more on higher frequency parts of the review they completely ignored the portions which might be less frequent but have more significance for the overall polarity of the review. To account for this, we created feature representations of words using TFIDF. The feature representation for this model is similar to the Bag of Words model except that we used TF-IDF values for each word instead of their frequency counts.

In classification approach each sentence is considered independent from other sentences. The labels we are interested in this project are (1) subjectivity of the sentence and (2) polarity of the sentence. We consider– Naive Bayes, Decision Trees, SVM, Logical Regression, KNN.

Naïve-Bayes:
Bayesian network classifiers are a popular supervised classification paradigm. An advantage of Naïve Bayes' is that it only requires a small amount of training data to estimate the parameters necessary for classification. Abstractly, Naïve Bayes' is a conditional probability model.

Decision Trees:
Decision trees are great in the sense that they are easily **interpretable** and follows a similar pattern to human thinking. In other words, you can explain a decision tree as a set of questions/business rules. The prediction is **fast**. It's a set of comparison operations until you reach a leaf node. They can be adapted to deal with **missing data** without imputing data

k-Nearest Neighbor:
K-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally, and all computation is deferred until classification. This rule simply retains the entire training set during learning and assigns to each query a class represented by the majority label of its k-nearest neighbors in the training set.

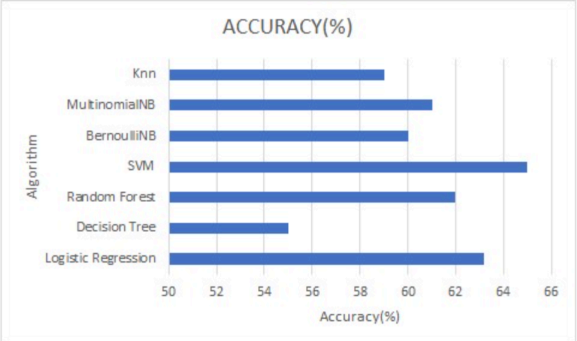
Random Forest:
The random forest model is part of the family set methods that take the decision tree as an individual predictor, they are usually based on the methods of Bagging, Randomizing Outputs and Random Subspace excusing boosting. It is an ensemble learning method for classification and regression that constructs a number of decision trees at training time and delivers the class that is the mode of the classes output by individual trees.

Logistic Regression:
It is used to determine the output or result when there are one or more than one independent variables. The output value can be in form of 0 or 1 i.e., in binary form.

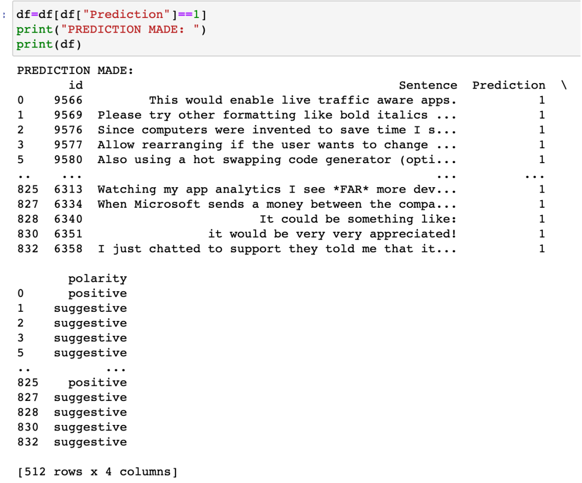
Support Vector Machines:
SVM is used for classification purpose. It can also be used in regression analysis and outlier detection. It constructs a hyper - lane in high dimensionality space

To make our discussion of SVMs easier we will be considering a linear classifier for a binary classification problem with labels y and features x.
We'll use $y \in \{-1, 1\}$ to denote the class labels and parameters w, b: $f(x) = w^T x + b$
w: normal to the line ; b: bias

Results:



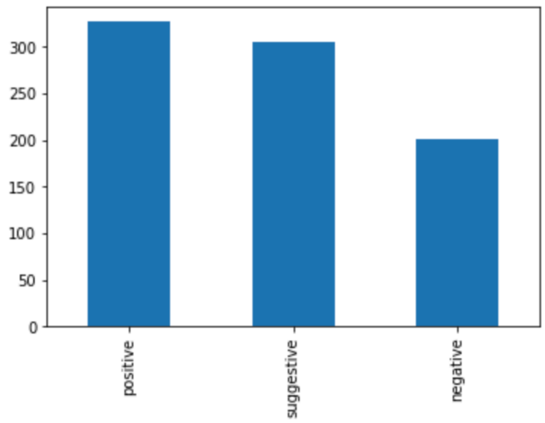
Fig(a): Accuracy comparison



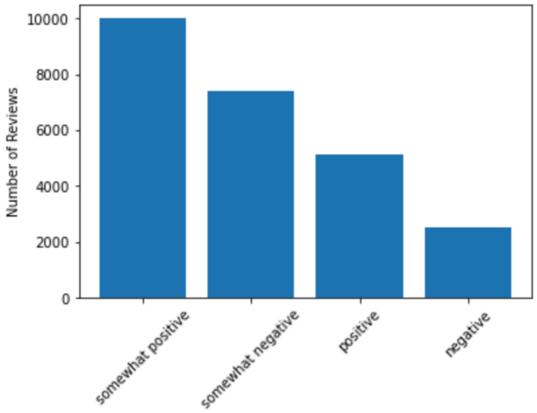
```
: df=df[df["Prediction"]==1]
  print("PREDICTION MADE: ")
  print(df)
```

```
PREDICTION MADE:
       id                                            Sentence  Prediction  \
0    9566          This would enable live traffic aware apps.           1
1    9569  Please try other formatting like bold italics ...           1
2    9576  Since computers were invented to save time I s...           1
3    9577  Allow rearranging if the user wants to change ...           1
5    9580  Also using a hot swapping code generator (opti...           1
..    ...                                                 ...         ...
825  6313  Watching my app analytics I see *FAR* more dev...           1
827  6334  When Microsoft sends a money between the compa...           1
828  6340                          It could be something like:           1
830  6351                   it would be very very appreciated!           1
832  6358  I just chatted to support they told me that it...           1

       polarity
0      positive
1    suggestive
2    suggestive
3    suggestive
5    suggestive
..          ...
825    positive
827  suggestive
828  suggestive
830  suggestive
832  suggestive

[512 rows x 4 columns]
```

Fig(b): Suggestive sentiment analysis result



Fig(c): Suggestive analysis visualized data



Fig(d): Fine-grained sentiment analysis result

Note: **Since the training dataset did not contain any review with sentiment score of 5, we did plot neutral text.**

2. Main References:

[1] Sentiment Analysis of Restaurant Review with Classification Approach in the Decision Tree-J48 Algorithm (https://ieeexplore-ieee-org.ezproxy.uta.edu/document/8884282)

[2] Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor using Naïve Bayes. (https://ieeexplore-ieee-org.ezproxy.uta.edu/document/8850982)

[3] A Deep Learning Approach for Extracting Polarity from Customer Reviews (https://ieeexplore-ieee-org.ezproxy.uta.edu/document/8765282)

3. Difference in APPROACH/METHOD between your project and the main projects of your references:

Our references focused mainly on binary classification. We implemented multiple algorithms for comparing their accuracy and implemented both fine graining and binary classification of text. We used multiple datasets for training and testing namely large movie review dataset and suggestions dataset. For fine graining we extracted the sentiment score from the name of the text files for both positive and negative reviews. For suggestive reviews we used bag-of-words and pattern-strings to train the model.

4. Difference in ACCURACY/PERFORMANCE between your project and the main projects of your references:

Accuracy and performance also depend on the dataset used to train the model. We received the highest accuracy and performance using SVM model whereas our references received highest accuracy for Random Forest model.

5. List of your contributions in the project (your work)

- Added suggestive sentiment analysis to the usual existing sentiments.
- Implemented data visualization of the results obtained for better understanding of results.
- Implemented multiple algorithms for classification for accuracy comparison

Analysis

1. What did we do well?

We successfully implement data visualization of fine-grained sentiments and also include suggestive sentiment classification. We also carried out feature engineering according to our model. We were successful in calculating the accuracy for multiple algorithms.

2. What could we have done better?

If we had access to machine with better configurations, we could have achieved far better results by tweaking our model.

3. What is left for future work?

We intend to create a real time application which can be used by restaurant owners to analyze the reviews received which will help improve their business. We also wish to work on better interactive data visualization of results. We also intend on increasing the accuracy of the model to 90%.

Conclusions

The aim of the project is to evaluate the performance for sentiment classification in terms of accuracy, precision and recall. We can conclude that SVM provides the best text classification accuracy for sentiment analysis. Apart from this, one can also use a Naïve Bayes' Classifier as they also provide good accuracy percentage.

Also, data pre-processing is extremely necessary to get better accuracy. It helps us understand the model construction. One of the major improvements that can be incorporated as we move ahead in this project is to merge words with similar meanings before training the classifiers.

REFERENCES

1. http://athena.ecs.csus.edu/~shevatep/Project_Report.pdf
2. https://towardsdatascience.com/social-media-sentiment-analysis-part-ii-bcacca5aaa39
3. https://www.sciencepubco.com/index.php/ijet/article/view/11991/4716
4. https://www.aclweb.org/anthology/R15-1036.pdf
5. https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/003.pdf