Sentiment Analysis for the Restaurant Reviews

Maulik Lalani

Pattern Recognition

Spring 2017

UF ID:13044420

Content

- Introduction
- Sentiment analysis overview
- Goal of the project
- Implementation using Naïve Bayes classifier
- Steps followed to perform the sentiment analysis.
- Compare with other classifiers

Introduction

- In the recent years, they have been tremendous increase in the data.
- More than 1.2 million terabytes of data on the internet as of 2013.
- There are more than 4.63 billion web pages on the internet till 2014.
- Different technologies are invented to store and retrieve the data efficiently.

Sentiment Analysis

- Because of the huge amount of data, it is not possible for us to find the insights using simple calculations.

- Sentiment analysis is the process of finding the valuable insights and future prediction from any form of data using pre processing of the data, applying different statistical modeling and algorithm.

Goal of the project

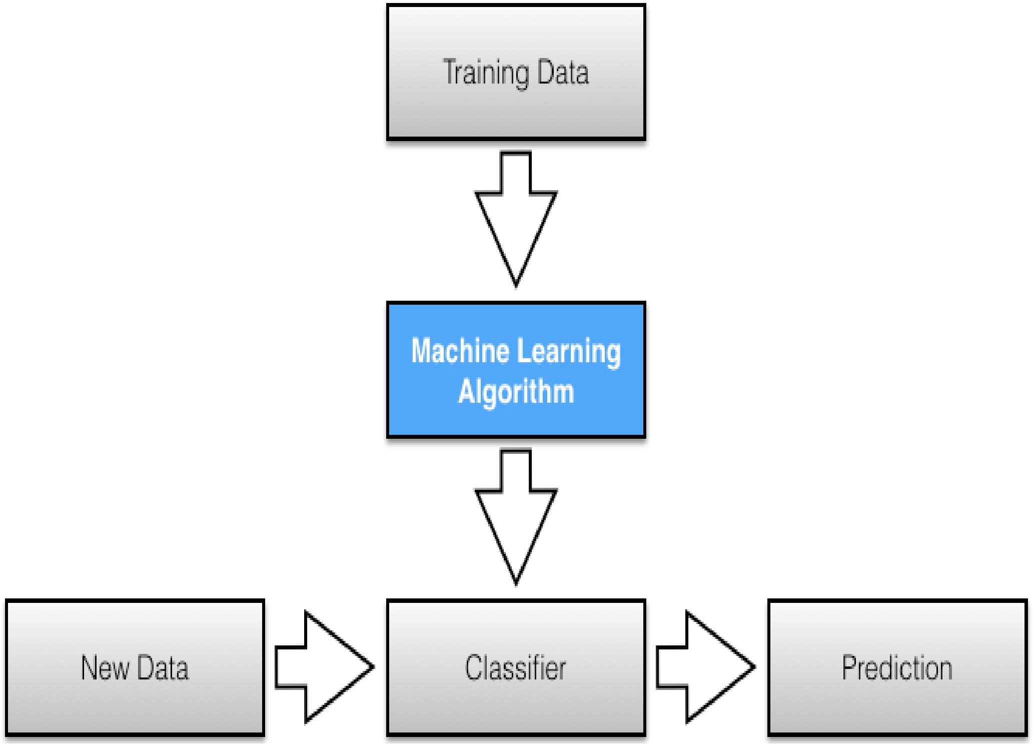
- To study different kinds of algorithms for the sentiment analysis and learn which one to use in different kinds of scenarios.
    - Naïve Bayes classifier
    - Decision Tree
    - K nearest neighbors
    - Maximum entropy
    - Support vector machines

- Implement end to end sentiment analysis process using Naive Bayes algorithm on the restaurant reviews and categorize the reviews as either positive or negative and either truthful or deceptive.

Implementation using Naïve Bayes Algorithm

- Steps that I have followed for doing the sentiment analysis:
  - Gathering Data.
  - Text cleaning.
  - Sentiment generation.
  - Create model.
  - Prediction.

Flow diagram for the process



Gathering Data

- Extract the data from the internet using different APIs and store it in the local system.
- I have used the online dataset and also have used yelp API to extract the data.
- Yelp also provides an API to extract the data and play with it

Text cleaning

- The input data that I got from the internet has to be cleaned before it can be passed to further processing.
- All the text is converted from the upper case to lower case to maintain the consistency.
- Punctuations are removed using python functions.
- Reviews are broken down into words.
- Duplicate words are also removed to avoid redundancy.

Create model using Naïve Bayes classifier

Likelihood

Class Prior Probability

$$P(c\mid x)=\frac{P(x\mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

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

Naïve Bayes theorem from my project perspective

- **P(class|word) = (P(word|class) * P(class)) / P(word)**

Where

- **P(class|word)** = probability of a review being in a class given the word in a sentence is deceptive.
- **P(word|class)** = probability of a word in a sentence is a class given the sentence is present in that class.
- **P(class) =** Probability of a review to be in class.
- **P(word)** = Probability of a word in a sentence to be in class.

Kind of probability in my model file

- **Class Probabilities:** The probabilities of each class in the training dataset.

- **Conditional Probabilities:** The conditional probabilities of each input value given each class value.

Prediction (Classifier)

- Once I have created the model file, whenever we give input review, each review is broken down into words and conditional probability for each word is multiplied and the final result is multiplied with the prior probability.

$$P(c|x) = P(x1|c)^{*}\ P(x2|c)^{*}\ldots\ {}^{*}P(xn,c)^{*}\ p(c)$$

- The above step is done for each of the classes.

- Then maximum of deceptive and truthful is taken and maximum of positive and negative is taken.

Evaluation

- Data is divided into training data and production data in 75:25 ratio.
- Production data is given as an input to the model and the result is compared with the training dataset and we can find the efficiency of the model.
- Accuracy = 80%
- Another way is to calculate the F score. It is the harmonic mean of recall and precision.

F = 2tp/(2tp+fp+fn) Weighted Avg: **0.89**

	Precision	Recall	F
Deceptive	0.81	0.9	0.85
Truthful	0.89	0.79	0.84
Negative	0.94	0.93	0.93
Positive	0.93	0.94	0.93

Study of other models: Support Vector Machines

- It is very accurate and very popular for the text classification.
- The data is divided into 2 classes using hyper plane such that it maximizes the margin between them.
- Hyper plane is given by:

(w.x) + b = Summation(yi*ai(xi*x)) +b =0

Where xi are the n dimensional input vectors

yi is the output values,

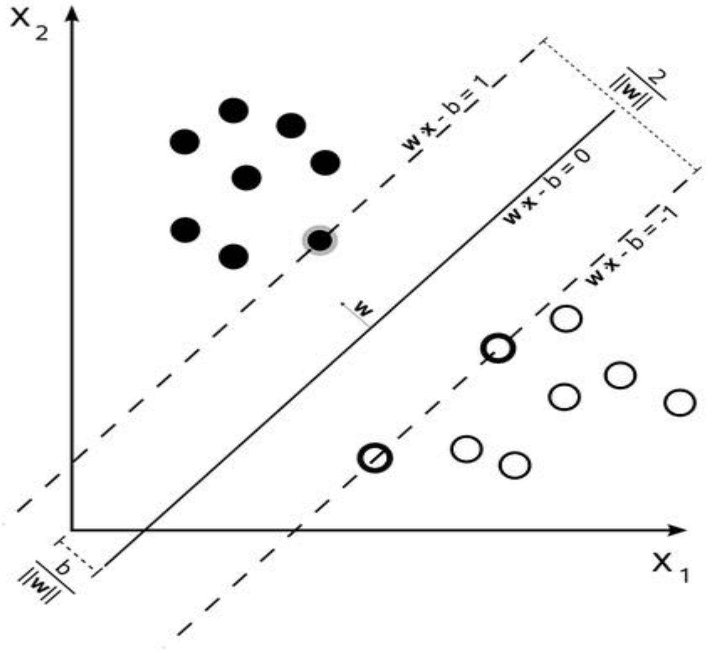
wi is the weight vector

ai is the Lagrangian multiplier.

- Once we have the plane, class of the other datasets can be found using

(w* xi+b>=0) then it belongs to the positive class otherwise negative.

SVM graph



SVM is used when there is a very complex relationship between the features and when you want to perform both regression and classification.

Maximum entropy

- This algorithm tries to maximize the entropy along with satisfying all the constraints.
- Steps for implementing maximum Entropy are given below:
  - Define a joint feature f(w,c) =N where N is the number of occurrences for class c.
  - Using optimization, assign a weight to the joint feature. This step is time consuming.
  - The probability of classes is given by the following formula:

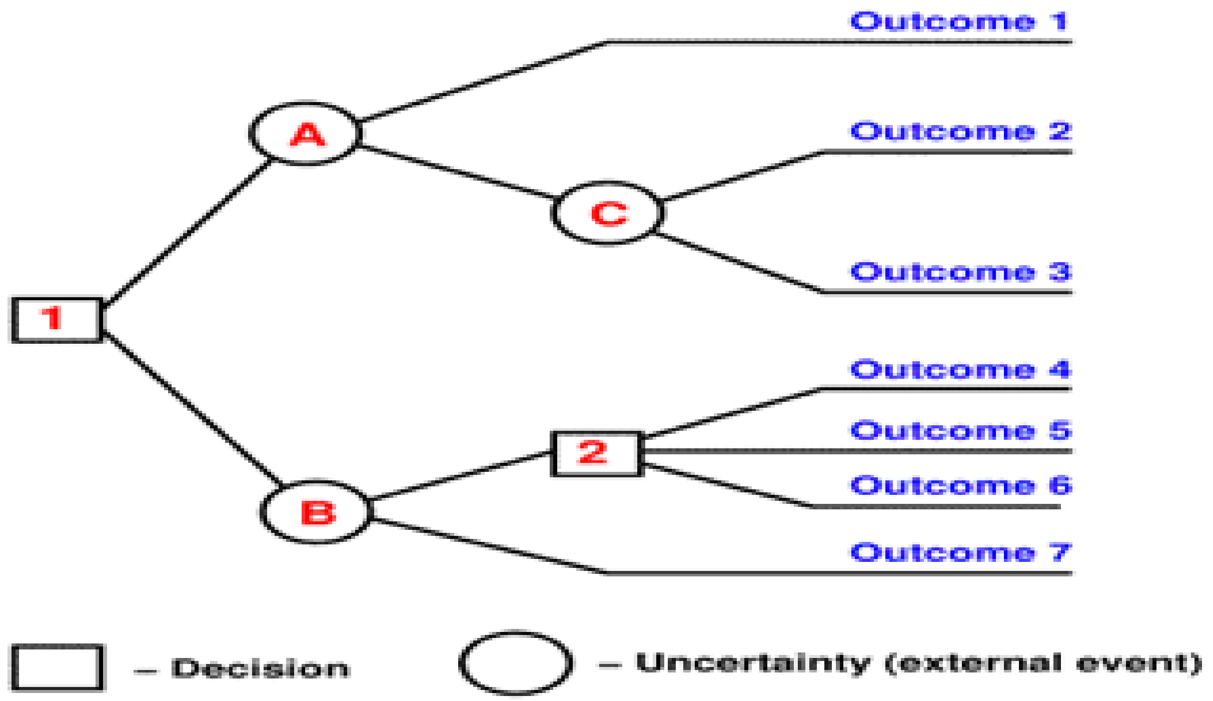
$$P(c|d,\lambda) \overset{def}{=} \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c' \in C} \exp \sum_i \lambda_i f_i(c',d)}$$

Advantages and disadvantages of Maximum entropy

- It is similar to Naïve Bayes classifier but it does not make assumption that all the features are independent and it handles joint feature.
- Hence, if we have features that are dependent on each other then Maximum entropy can be used.
- Also, it can handle Integers and Boolean data types also.
- However, it is slower than Naïve Bayes classifier because it calculates the weights of the joints using search based optimization.
- **Hence Maximum entropy is used when there is a relationship between features.**

Decision Tree

- Decision tree forms the model in the form of tree where each node is a ‘test’ and each branch is the possible outcome of the result as shown in the figure below.



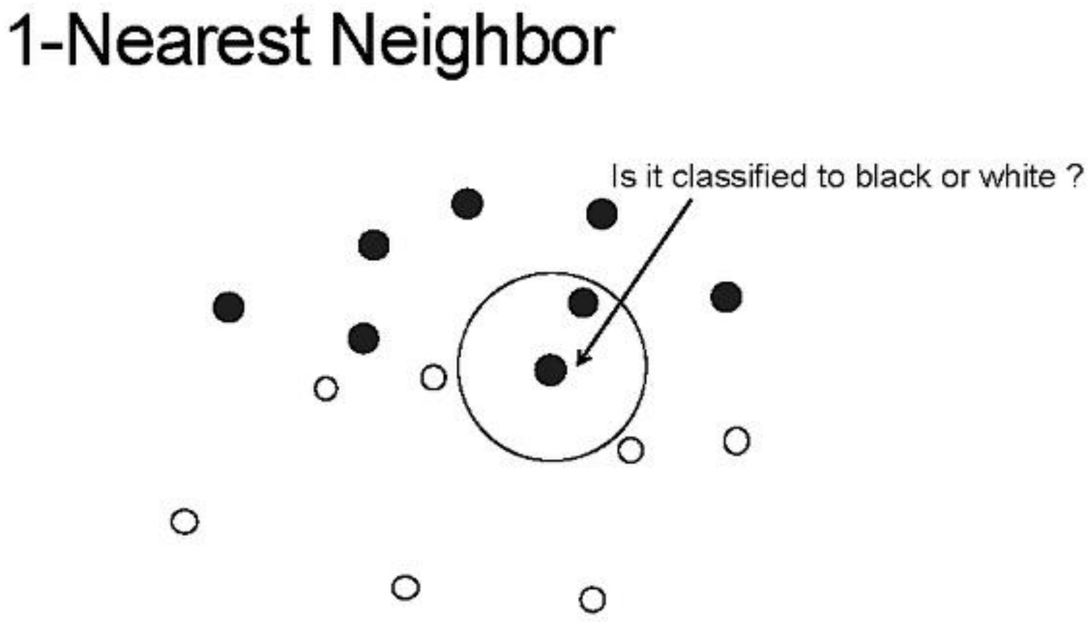
Why to use Decision tree

- Very little effort is required for the data preparation for decision tree algorithm because there no need to do scaling for the numeric values because it does not affect the tree.
- Decision trees are also not sensitive to outliers since the splitting happens based on proportion of samples within the split ranges and not on absolute values.
- Decision trees are also not sensitive to the missing values.
- Non linear relations do not affect the performance.
- It is very easy to interpret and explain the logic to the collogues.
- **Hence, decision tree is used when relationships are highly non linear and data is not proper.**

K nearest neighbors Algorithm

- This algorithm finds the probability of a class from the k closest point to the given input.
- Unlike other algorithms, it does not need to pre process the model.
- However, because of this, speed of the algorithm is reduced when there is huge amount of data because we will have to calculate the minimum distance for each point.
- **Hence This algorithm is used when the data is less and you don’t want to spend resources in training the data.**

K nearest algorithm diagram



Conclusion

- To summarize, I have implemented Naïve Bayes classifier to do the sentiment analysis on the restaurant reviews and find out whether the review is negative or positive and whether it is truthful or deceptive.
- K nearest algorithm is used when the data is less and you don’t want to spend resources in training the data.
- Decision tree is used when relationships are highly non linear and data is not proper.
- Maximum entropy is used when there is a relationship between features.
- SVM is used when there is a very complex relationship between the features and when you want to perform both regression and classification.

Future Scope

- This is done using basic python and naïve Bayes classification.
- We can also do stemming as a part of data preprocessing using NLTK libraries.
- Ex: if we have key words like go, gone, going etc. then we can combine all these words into one word and increase the accuracy.
- Also, because of this project I learnt about the Yelp data challenge contest where you compete with people all around the world and work on the datasets given by them to do some research in topics like Location mining, Seasonal trends, NLP, and social graph mining.
- I will be participating in this contest this year.