RUBIX

PROJECT REPORT

NATURAL LANGUAGE PROCESSING

Member Details:

PRERIT JAIN

TARUN BHATTAR

INDEX

1

MACHINE LEARNING APPROACH TO SENTIMENT ANALYSIS

❖ Introduction

- ❑ **Problem Statement :**
  - ➢ To use different machine learning algorithms for sentiment analysis of reviews of companies given by their employees on Glassdoor’s website and accordingly giving them scores.

- ➢ To Generate an overall score for the company which can be further used as an independent variable for credit scoring.

- ❑ **Implementation:**
  - ➢ Scraping data from Glassdoor's website which would be further used for creation of training set.
  - ➢ Once we get the reviews scraped , we would then create the dataset **manually** by reading about 3000 to 4000 reviews and assigning the polarity on the basis of our understanding.
  - ➢ Finally we need to apply various machine learning algorithms for the classification of the reviews and finding the best out of them.

❖ Methodology

- ❑ **Data Extraction:**
  - ➢ The access to scrape the glassdoor website is denied as per there robots.txt file.
  - ➢ We have for now settled to a temporary solution of getting the data using a google chrome extension(web scraper). It provides all the reviews in a csv file format which can be easily used for further analysis.

- ❑ **Training Dataset creation :**
  - ➢ We have created a dataset consisting of approximately 4000 reviews with their polarity **manually** assigned . We can now use it as a training data set for our machine to learn the pattern.

  - ➢ As assigning polarity manually to each and every review is a tedious task we are now looking for a hybrid approach, i.e the task which we have done manually, shall be done using the **Dictionary-Based approach.**

  - ➢ In a dictionary based approach we have a dictionary in which words are classified as positive or negative before hand. So when a review is analysed, it compare the words in the review with the words in the dictionary and predict whether a review is positive or negative on the basis of occurence of positive or negative word on it.

- ➢ We have used the dictionary **“bing”** which is an inbuilt general purpose dictionary in tidytext package of R.

❑ **Machine Learning :**

- ➢ A normal machine learning problem is like a machine, learning the underlying relationship between some variables using features selected from a given dataset and then implement that knowledge on new data to make predictions on the basis of its learning.

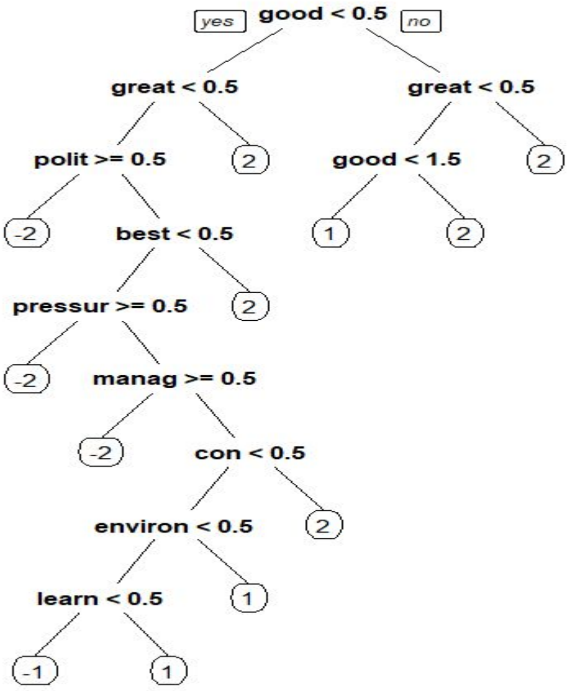
- ➢ So in our case what we want is ,our machine should understand using the training set the relation between the occurrences of several words in the reviews and classify them being positive or negative. And when a new review come it should be able to classify and predict it as positive or negative.

- ➢ **Working**
  - ➔ We are considering the frequency of the important words as independent variables or we can say as factors to predict a review being positive or not.
  - ➔ So first it will learn from our training set that which words are there which are more frequent in positive reviews and negative reviews . This knowledge would be used to predict and classify the new reviews as positive or negative.)
  - ➔ For eg. our training set has a lot of sentences which are classified as positive by us and have phrases like “great work life balance”, “good co workers”. So when a new review comes and consist of words like “good” and “great”, it will be assigned as positive by our model.

- ➢ **ALGORITHMS USED:**

➔ **CART (Classification and Regression Trees)**

❑ This algorithm forms a tree which looks somewhat like as shown above. And in the case of a new review it follows the path from the tree and allot a polarity.



➔ **Random Forest**

❑ This algorithm makes a lot of trees similar to the one shown above and hence is called a forest. It then take votes from the result of each tree and then assign a polarity to the review on the basis of that.

➔ **Naive Bayes**

❑ This algorithm simply calculate the probability of a review being positive and negative considering the occurences of words in it. And assign the one which has a higher probability.

➔ **Multinomial Logistic Regression**

❑ This algorithm creates a function using a hypothesis on the basis of training set and if the value of the function for a particular review is >=.5 then it assigns a particular class.

➔ **Support Vector Machine (SVM)**

❑ This algorithm creates an hyperplane according to the training dataset points and then classify on the basis of that i.e the datapoint lying on one side of dataset is classified as a different class than the one lying on the other side. Binary classification seems easier in this case.

So, training these models and then applying it on testing dataset, we will get reviews and there assigned polarities as an output. After the assignment of polarities to the individual. We have to choose a metric which should be used in order to get an overall score of the data.

❑ **Overall Score:**

❑ We are using the proportion of positive reviews out of total reviews scaled to five as a score to the company which can be used as a metric for judging how the company manages its employees.

Overall_Score = 5*(pos) / (pos + neg)

Pos → No. of reviews classified as positive.
Neg → No. of reviews classified as negative.

❖ RESULTS :

ALGORITHMS	CART	RANDOM FOREST	SUPPORT VECTOR MACHINE	NAIVE BAYES	MULTINOMIAL LOGISTIC REGRESSION
TRAINING SET ACCURACY (CONSIDERING MULTI-CLASS CLASSIFICATION)	54%	86.3%	-	43.5%	69.11%
TEST SET ACCURACY (MULTI-CLASS CLASSIFICATION)	55%	64.5%	-	40%	61%
TRAINING SET ACCURACY (BINARY CLASS CLASSIFICATION)	77.3 %	94.24%	88.5%	73.24%	88.3 %
TEST SET ACCURACY (BINARY CLASS CLASSIFICATION)	79.2%	86.2%	84.5%	71%	85.2 %

- ➢ The accuracy of machine learning model is always compared with a **baseline model.** For eg., In case of a classification problem a baseline model predicts the most frequent outcome as a result to any input. Hence the accuracy of any model we apply should be at least greater than this obvious baseline model.

- ➢ **Baseline Model's Accuracy:**
  - ○ **Multi-class Classification :** 29.2%
  - ○ **Binary Classification :** 50.1%

- ➢ So we can see that our models are predicting quite good as compared to the baseline model. Accuracy of more than 60% is quite good in case of multi-class classification.

- ➢ So our result tell that random forest is doing quite well in case of multiclass classification and for binary classification multinomial Logistic Regression gives a good accuracy.

- ➢ Here is the table consisting of Overall rating of the companies and their rating given on the website.

Company Name	Rating given using our model	Rating as given in website
Onicra	2.93	3.3
Star India Pvt. Ltd	4.17	4.2
TCS	3.93	4.0
Paytm	4.01	4.1
Flipkart	3.78	4.1
Birla	3.68	4.0
Kotak Mahindra Bank	3.6	3.9
Express Scripts	3.12	3.4
Amazon	4.24	3.8
Facebook	3.85	4.2
Shapoorji Pallonji	3.73	3.9
Dish tv	3.67	3.9

- ➢ We have noticed that the ratings given by our model and the given in website are almost similar.
- ➢ According to our analysis of about 12 companies we have decided a benchmark below which a company can be concluded as a company with low employee satisfaction.

Benchmark Score 1 = 3.3 (Out of 5)

Benchmark Score 2 = 3.9 (Out of 5)

- **Interpretation of company's overall score based on benchmark scores :**

  - If Overall_Score given by our model is **greater than benchmark score 2** then the company is said to be **good** in employee satisfaction.

  - If Overall_Score given by our model is **between benchmark score 1 and benchmark score 2** then the company is said to be **average** in employee satisfaction.

  - If Overall_Score given by our model is **less than benchmark score 1** then the company is said to be **poor** in employee satisfaction.

❖ LIMITATIONS:

- **DATASET SIZE:** The dataset size is quite small now. And Increasing its size increases the accuracy.

- **SARCASM HANDLING:** As we are considering the frequency of words in the reviews, so our model will fail in case the user uses sarcasm in a review.

- **STATEMENTS HAVING MULTIPLE SENTIMENTS:** Our model may get confused where the reviews consist of both words showing positive and negative statements. For eg. Consider the Review "The company has great infrastructure, good peers, good work life balance but there is a lot of politics". So in this case our model may classify it as positive just because it is more on the positive side. And the half negative part may get neglected.

- **ACCURACY OF MULTI CLASS CLASSIFICATION:** Even if the multi-class classification is doing far better than the baseline models but still we need much more accuracy in order to give a better and trustworthy review to the customer.

- ➢ **LACK OF A GOOD DICTIONARY:**
  - ○ As we are using a General-purpose dictionary from an R package for the sentiment analysis, the results are not up to the mark.
  - ○ This is because the dictionary is designed by keeping in mind a particular context which doesn’t matches with our field of interest.
  - ○ For eg: The general purpose dictionary doesn’t even consider ‘politics’,’pressure’ as negative words.
  - ○ Hence there is an urgent need of creating our own dictionary.

❖ SUGGESTIONS TO RUBIX

- ❏ **To Handle the limitations**
  - ➢ For Sarcasm handling part we have to try some thing which is hybrid of lexicon(own purpose) based approach and machine learning approach.
  - ➢ Getting a larger dataset.

- ❏ Also in order to overcome the limitations of the multiclass classification what can be done is to first classify the reviews into positive and negative reviews using the binary classification model and then separately have models for classifying the negative reviews further into weakly negative or strongly negative and same with positive one.

- ❏ In order to save the manual effort we have opted a hybrid approach in which we will be using lexicon based approach to create the training dataset.

- ❏ We are providing you with a list of **2,750 frequently occuring words** in the Reviews which we have taken from glassdoor. Please **get the polarity of words assigned by someone** so that we can have a better dictionary which can be used for training dataset creation and getting more promising results. The CSV file consisting of the words is there in the google drive folder shared and also in your PC.

2

ASPECT - BASED SENTIMENT ANALYSIS

❖ INTRODUCTION

❑ WHAT IS AN ASPECT ?

- ➢ An aspect is the explicit reference of an entity about which an opinion is expression in a sentence. Opinion can be positive, negative or neutral.

❑ ATTRIBUTE-BASED SCORES:

- ➢ If we go down to the question that why do we analyse reviews and dig a little deeper, we realise that the central idea of analysing unstructured data is to extract the information from a particular piece of text in a form which can be used to make interpretations and draw insights from it.

- So, analysing the review as a whole and just doing binary classification can be considered as loss of information.

- Sometimes, multiple aspects of a product or service are discussed in the same Review. Assigning an overall score, doesn’t allow us to extract the feedback about each individual aspect rather it gives an overall sentiment of the review.

- Hence separating the aspects in a review can help us to produce better results considering the employees as customers.

❖ METHODOLOGY :

In order to extract the information regarding different aspects we have to follow a stepwise algorithm which is a logical set of instructions as given below:

❑ ASPECT-TERM EXTRACTION :

- As the heading suggests, this part deals with extracting important aspect terms present in the reviews i.e the parameters on the basis of which a company is judged by an employee.

- Aspect terms are **basically nouns** in the available piece of text. So in this part we have to extract all the important nouns from all the reviews.

- So, We have used a Java based R library called **RDRPOSTagger** which uses a **rule-based approach** in order to tag different words with their respective POS Tags. (Parts-of-Speech)

- We have got about **4,000 nouns** from about **3600 Reviews**.

- ➢ For extracting important nouns from these 4k nouns we first found out the **most frequently occuring 500 words** in 3600 reviews and than took the **words occurring in both the sets** which results a set of about **50 important nouns** which can be considered as the aspect terms.

❏ ASPECT-TERM AGGREGATION :

- ➢ The second task is to aggregate aspect terms into particular aspects. For eg: nouns like **environment**, **atmosphere**, **politics** etc., can be clubbed into an aspect **'Work culture and Values'.**

- ➢ There can be two approaches for aggregation:

  - ○ **Manual Classification:**

    - ■ This is manually classifying the aspect terms into different aspects and then using these set of keywords to tag different sentence on the basis of the aspect which is discussed in that particular sentence.
    - ■ We have done the manual classification of 50 nouns and provided the CSV file for it in the drive link shared.

  - ○ **PCA-based factor Analysis:**
    - ■ Using the PCA-based factor analysis to club different nouns in to a particular aspect. (PCA stands for Principal Component Analysis)

❏ ASPECT-BASED POLARITY :

- ➢ This is the major part where we will assign the aspect based score to the reviews. This is a stepwise procedure:

- **Tokenizing into sentences:**

  - Tokenization at the sentence level will allow us to extract the information about the aspect which is discussed in each particular sentence of the review.
  - Also,we should make sure that we have an index for keeping an account of the review from which each particular sentence comes from.

- **Classifying sentences into Aspects:**

  - Now, on the basis of the occurrence of aspect terms in a particular sentence we will be assigning aspects to that particular sentence.
  - For eg: consider a sentence “I like the work culture”. So, it consist of the word ‘culture’ which is an aspect term under the aspect “**Work culture and Values”.** So we will infer from it that this particular sentence is talking about this particular aspect.

- **Sentiment analysis of the sentence:**

  - After classifying sentences on the basis of aspects, we perform the sentiment analysis of each review by any of the approach which is suitable

- **Combining the Scores of different sentences at Review level :**

  - Now we would take an average of the scores of the sentences talking about same aspect in the same review and assign the average as the aspect score of that particular aspect of that particular Review.

➢ Here, First and second steps are to be performed for just first time. Once you get the **keyword dictionary** for each and every aspect you can perform the third part of assigning scores on any new data.

3

Instruction Set

You should follow following instructions in order to run our code on your machine.

- **Softwares to be used** :
  - R
  - RStudio
  - Java (should be there in the pc for a particular library we have used)
  - MS-Excel

- **Common Instructions:**

- Open our code in RStudio or any other text editor for R.

- Make sure that you have installed all the libraries in your system which are being used in the code.

  - If haven’t done yet, use **install.packages(‘package_name’)** command.

- The working directory should be properly set for reading the csv files.

  - In order to set the correct working directory path you can traverse to the working directory by clicking on the **three dots symbol** present in the right bottom sub-window and then clicking **more** and selecting **set as working directory** from the drop down menu.

  - Than you will see the path to your working directory in the **Console**.

• Code-wise Instructions :

○ Dictionary-Based-Final.R

- **INPUT :** A CSV file consisting of a Reviews stored in a column named **‘Reviews’.**

- **RUNNING INSTRUCTIONS :**
  - In **line number 12** add the path to your input csv file.
  - In **line number 14** add the name of your input csv file.

- **OUTPUT :**
  - Output will be a CSV file named **‘polarity_assigned.csv’.** You can change the name of output file if you want by going to **line number 67** and changing the name.

        - This file will have two columns one consisting of the Reviews and the other the Polarity assigned using the lexicon based approach.
        - You can find the file in the working directory set by you in line number 12.

- ○ **Machine-Learning-Based.R**

    - ■ **INPUT :** A CSV file consisting of **training dataset** that is it should consist of two columns necessarily. One is ‘**Reviews’** and other is ‘**Polarity’.**

    - ■ **RUNNING INSTRUCTIONS :**
        - In **line number 21** add the path to your input csv file.
        - In **line number 25** add the name of your input csv file.
        - Select upto **line number 50** and hit ‘**ctrl + Enter**’.
        - Below that you can find seperate 4 line codes for different algorithms. So you can run whatever you want. It is properly commented that which four line is for which part.

    - ■ **OUTPUT:** Output will be two tables in the console. One consisting of training set predictions and actual values and other same for the test set.

- ○ **Multinomial_Logistic_Regression_Overall_Score.R**

    - ■ **INPUT :**
        - A CSV file consisting of **training dataset.**
        - A CSV file consisting of a Reviews stored in a column named **‘Reviews’.**

    - ■ **RUNNING INSTRUCTIONS :**
        - In **line number 22** add the path to your input CSV file in the path_1 variable.
        - In **line number 27** add the name of your training data CSV file.

    - In **line number 60** add the path to your raw csv file whose polarities you have to predict in the path_2 variable.
    - In **line number 65** add the name of your raw csv file.

  - **OUTPUT :**
    - Company classified as bad, average or good with respect to employees reviews printed on the console.
    - Overall Score of the company out of 5 printed on the console.
    - A CSV file named ‘new_data.csv’ consisting of individual review’s polarity. You can change the name of output file if you want by going to **line number 105** and changing the name.
    - This file will have two columns one consisting of the Reviews and the other the Polarity assigned using the lexicon based approach.
    - You can find the file in the working directory set by you in line number 12.

- ### Random-Forest_Overall_Score.R

  - **INPUT :**
    - A CSV file consisting of **training dataset.**
    - A CSV file consisting of a Reviews stored in a column named **‘Reviews’.**

  - **RUNNING INSTRUCTIONS :**
    - In **line number 22** add the path to your input CSV file in the path_1 variable.
    - In **line number 27** add the name of your training data CSV file.
    - In **line number 58** add the path to your raw csv file whose polarities you have to predict in the path_2 variable.
    - In **line number 66** add the name of your raw csv file.

  - **OUTPUT :**

        - Company classified as bad, average or good with respect to employees reviews printed on the console.
        - Overall Score of the company out of 5 printed on the console.
        - A CSV file named ‘new_data.csv’ consisting of individual review’s polarity. You can change the name of output file if you want by going to **line number 111** and changing the name.
        - This file will have two columns one consisting of the Reviews and the other the Polarity assigned using the lexicon based approach.
        - You can find the file in the working directory set by you.

  - ### Noun_Extraction.R

    - **INPUT :** A CSV file consisting of a Reviews stored in a column named **‘Reviews’.**

    - **RUNNING INSTRUCTIONS :**
        - In **line number 16** add the path to your input csv file..
        - In **line number 18** add the name of your input csv file.

    - **OUTPUT :**
        - Output will be the list of important nouns from the reviews in a list named **imp_nouns.**

- ## Some General Information :
  - The code for Random Forest algorithm may take some time to run so don’t lose patience.
  - Also the code for Noun Extraction will also take some time to learn.(It may take more than 5 minutes too.)
  - All the codes are properly commented. So you can easily understand the use of each and every command.
  - In case of any conceptual information please see the report submitted.
  - You can find all the codes in your PC and also in the google drive we have shared with you.