Sentimental Analysis on Amazon Fine Food Reviews

Baobao Geng
Marquette University

Introduction

"What others think? "is always important information in a decision-making process. Every day people discuss various products on social media sites. Sentiment analysis of reviews is a popular task in natural language processing[1]. In the era of social media and internet, sentimental analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. As an extremely valuable tool for social media companies, business owners and advertisers, sentiment analysis is already providing insights that help drive effective business decisions, strategies. As we all know that amazon is one of the biggest e-commerce website. Here you can buy many different varieties of product you want and for every product we buy we can write a review as well as rate the product. This project aims at making prediction model where we will be able to predict whether a recommendation from a user of amazon is positive or negative. Studying the opinions of customers helps to determine the people feeling about a product and how it is received in the market. And the result will help business owners make timely adjustments to their products.

Method

I. Data set Used

Here we use the amazon rating review dataset provide by "Kaggle" which is one of the largest machine learning platform where you can get standardised dataset. The Amazon Fine Food Reviews dataset is a vast dataset having 568454 reviews about fine food products from Amazon. The reviews have been written by reviewers over a span of 10 years. ach record in the dataset comprises of ten columns:
1.Id
2. ProductId - unique identifier for the product
3. UserId - unqiue identifier for the user
4. ProfileName
5. HelpfulnessNumerator - number of users who found the review helpful
6. HelpfulnessDenominator - number of users who indicated whether they found the review helpful or no
7. Score - rating between 1 and 5
8. Time - timestamp for the review
9. Summary - brief summary of the review
10. Text-textofthereview
But among them the column consisting user information, product information, review, review summary and ratings are considered as the main features. This project mainly focuses on the

important features namely ‘score’ and ‘summary’ which are the ones that can be used in building the prediction model for sentiment classification.

II. Data PreProcessing

The mean of all the scores is 4.18. So, to initiate and distinguish the ratings into two sentiments i.e. positive and negative, it is assumed for all reviews having score above 3, i.e. 4 and 5 can be considered as positive and the rest are considered as negative having a value of 3 or below. Next we normalize the review using a series of steps: 1. Switch to lower case 2. Remove numbers 3. Remove punctuation marks and stopwords 4. Remove extra whitespaces

III. Feature Extraction

The weighted word frequencies are determined by TF-IDF method[2]. Term Frequency and Inverse Document Frequency is one of the most favored methods of determining the weighted word frequency. TF-IDF measures the relative importance of a word to a document. In this step, we use a R package called tm. To reduce the dimension of the TF-IDF, we remove the less frequent terms.

IV. Statistical Methodology

For any given dataset, many conceivable features can be chosen. An important point to consider is which features to utilize. PCA (Principal Component Analysis) is an eminent strategy of feature reduction. Its point is to take in a discriminative transformation matrix to lessen underlying component space making it to a lower dimensional[3]. After that, we select the first 300 principal components and logistic regression model is applied on the selected data. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable[4]. We use ROC (Receiver Operator Characteristic Curve) and (AUC Area Under the Curve) as evaluation metrics for calculating the performance of any classification model. ROC is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate and False Positive Rate. AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).
In this case, we use a R function called “prcomp” to perform PCA analysis. For logistics regression analysis, we use “caret” package. The caret package (short for Classification And REgression Training) is a set of functions that attempt to streamline the process for creating predictive models. In this step, we partition the data into 70% training and 30% evaluation. “pROC” packag is used here for visualizing ROC and calculating AUC.

Results

I. Word Clouds

The visualization of opinion words and its counts with sizes relative to their counts is displayed in Fig.1. This word cloud is created by R software using text mining packages. Frequency matrix is constructed before generating the word cloud.

Figure. 1 Word Cloud of the entire dataset.

II. Screeplot of PCA Results

Screeplot which shown in Figure.2 plots the variances against the number of the principal component.

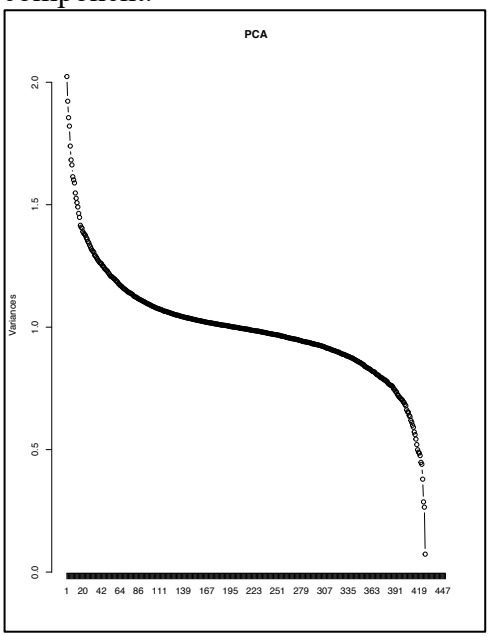

Figure. 2 Screeplot of PCA results. X-axis indicates the principle components and Y-axis shows the Variances.

III. Confusion Matrix

Table. 1 shows the Confusion Matrix and statistics from logistic regression algorithm. The results show that the accuracy of the model is 82.25%.

```
Confusion Matrix and Statistics

          Reference
Prediction    Neg    Pos
       Neg  11263   4129
       Pos  26140 129004

               Accuracy : 0.8225
                 95% CI : (0.8207, 0.8243)
    No Information Rate : 0.7807
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3426

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9690
            Specificity : 0.3011
         Pos Pred Value : 0.8315
         Neg Pred Value : 0.7317
             Prevalence : 0.7807
         Detection Rate : 0.7565
   Detection Prevalence : 0.9097
      Balanced Accuracy : 0.6351

       'Positive' Class : Pos
```

Table. 1 Confusion Matrix and statistics

IV. Performance charts

Figure. 3 summarizes the output found on the logistic regression analysis. In the cumulative lift chart (Figure 3a), the red dotted line represents the baseline model, and the black solid line represents the results from the logistic regression model. In this case, the solid line from the logistic regression model lies well above the red diagonal line which indicates that the logistic regression model performs better. In the Figure 3b, we can see the ROC curve from the logistic regression model lies well above the diagonal line. The area under the ROC curve, or AUC, is 0.8105, indicating that the logistic regression model performs well in terms of sensitivity and specificity.

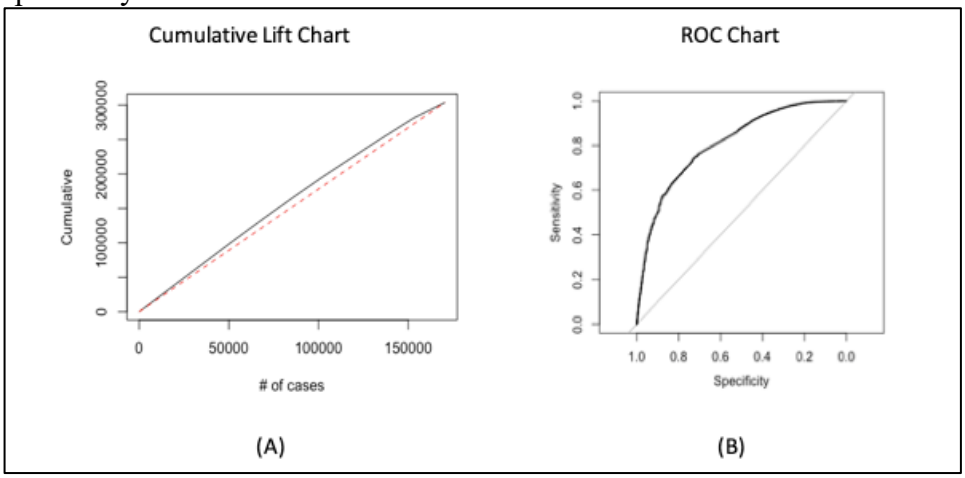

Figure. 3 Plots drawn from the logistic regression analysis. (a) In this Cumulative Lift Chart, X-axis indicates the cases and the Y-axis indicates the cumulative lift. (b) Receiver Operating Characteristic curve, the X-axis is specificity (FDR) and the Y-axis is sensitivity (TPR).

Reference

[1] Manankumar Bhagat. 2018. Sentiment Analysis using an ensemble of Feature Selection Algorithms. Master of Science. San Jose State University, San Jose, CA, USA. DOI:https://doi.org/10.31979/etd.xg3j-fty7
[2] Ian T. Jolliffe and Jorge Cadima. 2016. Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc. A.* 374, 2065 (April 2016), 20150202. DOI:https://doi.org/10.1098/rsta.2015.0202
[3] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. 9.
[4] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M. Ingersoll. 2002. An Introduction to Logistic Regression Analysis and Reporting. *The Journal of Educational Research* 96, 1 (September 2002), 3–14. DOI:https://doi.org/10.1080/00220670209598786

R code

```
#install.packages("tokenizers")
#install.packages("stopwords")
install.packages("tm")
#library(tokenizers)
#library(stopwords)
#library(stringi)
library(tm)
library(wordcloud)
library(caret)

library(gains)
library(pROC)

setwd("~/MU/Courses/DataMining/homework/hw9/")
data <- read.csv("Reviews.csv")
subset <- data[, c(7,9)]
subset$Score <- ifelse(subset$Score > 3 , "Pos", "Neg")
#subset$Summary <- as.character(subset$Summary)
#subset$newterm <- tokenize_word_stems(subset$Summary)
#subset$Summary <- stri_trim(subset$Summary)
#subset$Summary <- stri_trans_tolower(subset$Summary)
subset_corpus <- Corpus(VectorSource(subset$Summary))
subset_corpus <- tm_map(subset_corpus, content_transformer(tolower))
subset_corpus = tm_map(subset_corpus, removeNumbers)
subset_corpus = tm_map(subset_corpus, removePunctuation)
subset_corpus = tm_map(subset_corpus, removeWords, c("the", "and", stopwords("english")))
subset_corpus =  tm_map(subset_corpus, stripWhitespace)

#subset_dtm <- DocumentTermMatrix(subset_corpus)
```

```
#findFreqTerms(subset_dtm, 1000)
#subset_dtm = removeSparseTerms(subset_dtm, 0.95)
#freq = data.frame(sort(colSums(as.matrix(subset_dtm)), decreasing=TRUE))
#wordcloud(rownames(freq), freq[,1], max.words=50, colors=brewer.pal(1, "Dark2"))

subset_dtm_tfidf <- DocumentTermMatrix(subset_corpus)
subset_dtm_tfidf = removeSparseTerms(subset_dtm_tfidf, 0.999)
freq = data.frame(sort(colSums(as.matrix(subset_dtm_tfidf)), decreasing=TRUE))
wordcloud(rownames(freq), freq[,1], max.words=50, colors=brewer.pal(1, "Dark2"))

subset <- cbind(subset, as.matrix(subset_dtm_tfidf))
subset$Score = as.factor(subset$Score)
subset2 <- subset[,c(-1,-2)]
PCA <- prcomp(subset2, scale=TRUE)
summary(PCA)
screeplot(PCA, npcs = 450, type = "lines")

subset3 <- data.frame(PCA$x[,(1:300)], subset$Score)
colnames(subset3)[301] <- 'Score'
set.seed(1)
myIndex<- createDataPartition(subset3$Score, p=0.7, list=FALSE)
trainSet <- subset3[myIndex,]
TestSet <- subset3[-myIndex,]
#QualityLog <- glm(Score ~., data = trainSet, family=binomial)

myCtrl <- trainControl(method="cv", number=10)
set.seed(1)
glm_fit <- train(Score ~., data = trainSet, method = "glm", trControl = myCtrl)
glm_class <- predict(glm_fit, newdata = TestSet)
confusionMatrix(glm_class, TestSet$Score, positive = 'Pos')
glm_class_prob <- predict(glm_fit, newdata = TestSet, type = 'prob')
TestSet$Score <- as.numeric(TestSet$Score)
gains_table <- gains(TestSet$Score, glm_class_prob[,2])
gains_table

plot(c(0, gains_table$cume.pct.of.total*sum(TestSet$Score)) ~ c(0, gains_table$cume.obs), xlab
= '# of cases', ylab = "Cumulative", type = "l")
lines(c(0, sum(TestSet$Score))~c(0, dim(TestSet)[1]), col="red", lty=2)
barplot(gains_table$mean.resp/mean(TestSet$Score), names.arg=gains_table$depth,
xlab="Percentile", ylab="Lift", ylim=c(0,1.5), main="Decile-Wise Lift Chart")
roc_object <- roc(TestSet$Score, glm_class_prob[,2])
plot.roc(roc_object)
auc(roc_object)
```