Estimate review ratings from review contents

Akhil Gudavalli[1]

Abstract— **The paper's main aim is to predict the ratings of the reviews using the review text. We also intend to detect spam or inconsistent reviews using sentiment analysis. Multinomial regression is applied on the sentiment score of the review and the features extracted from the bag-of-words approach to predict the rating of the review.**

I. INTRODUCTION

The growth of the World Wide Web has resulted in tonnes of reviews for products we wish to purchase, destinations we may want to travel to, and decisions we make on a day to day basis. User experience would be greatly improved if the structure of the content in reviews was taken into account, i.e., if review parts pertaining to different product features (e.g., food, ambience, price, service for a restaurant), as well as the sentiment of the reviewer towards each feature (e.g., positive, negative or neutral) were identified. This information, coupled with the metadata associated with a product (e.g., location or cuisine for restaurants), can then be used to analyze and access reviews. However, identifying structured information from free-form text is a challenging task as users routinely enter informal text with poor spelling and grammar.

One of the most important opinion-mining tasks is sentiment classification, whereby the opinion documents are categorized into two sentiment categories: positive and negative. In this paper, we focus on a finer-grained task, where we consider the object of our prediction can be among a finite range of integers. We call this task the rating-inference task; It determines an authors polarity evaluation within a multi-point scale (e.g. one to five stars).

We explore solutions for this task in the context of product or service reviews, which are one of the most important opinion resources and widely used by costumers and companies. We observe that in many real-world scenarios, it is important to provide numerical ratings rather than binary decisions, especially when a customer compares several candidate products, all of them are positive in a binary classification, to make a purchase decision, since customers not only need to know whether a product is good or not, but also how good the product is. A recent study pointed out that many consumers are willing to pay at least 20more for an excellent product (with 5-star rating) than a good product (with 4-star rating)

II. DATASET

The dataset consists of the reviews of the below mentioned applications in the Google play store.
a) AccuWeather
b)PayPal
c)Swiggy
d)Lyft
I have collected around 640 reviews of each application mentioned above.

A. Data Collection

Data is collected using a node.js module named Google-play-scraper. The available methods in the module are app,list,search,developer,suggest,reviews,similar and permissions. The methods used for data collection are search and reviews.

The description of the methods:
search: Retrieves a list of apps that results of searching by the given term. This gives us the appId of the application we are going to retrieve the reviews.
reviews: Retrieves a page of reviews for a specific application. appId is the main parameter for this method. The sort parameter is used to sort the reviews useful in retrieving the latest reviews. The page parameter is used to retrieve the reviews of the specific page. Every page has 40 reviews at most. For our study, we have taken around 16 pages[0-15] to retrieve 640 reviews.

The result consists of many fields such as the reviewId, reviewUrl, title, score, date and the user details. I have considered only the title and the score fields for our analysis
title: the review text
score: the rating given by the user

B. Data Pre-processing

One of the first steps in working with text data is to pre-process it.. Majority of available text data is highly unstructured and noisy in nature. To achieve better insights or to build better algorithms, it is necessary to play with clean data. social media data is highly unstructured with the presence of unwanted content like Stopwords, Expressions etc.

Review text is preprocessed to achieve better results. For the analysis only the alphabetical words are considered. Hence, every other expression other than the alphabet is replaced by a space. All the words are converted into lowe case.Then the stop words are removed from the review text.

*This work was not supported by any organization

[1] Akhil Gudavalli is a Masters Student in Computer Science Dept,Utah State University,Logan,Utah

Hence the review text is now formed by the meaningful words.

III. METHODOLOGY

To predict the rating of the review based on the review content, a regression model is applied on the review content and the rating of the review. But regression can't be directly applied on the review content. So, the necessity of converting the review content into another representation arises. Hence, the review content is converted into some numeric representation for the analysis. The method we have used to convert to the numeric representation is Bag-of-words.

A. Bag-of-Words

The Bag of Words model learns a vocabulary from all of the documents, then models each document by counting the number of times each word appears.The user takes the text to be classified and counts the frequencies of the words in each object, followed by some sort of trimming to keep the resulting matrix of a manageable size.

For example, consider the following two sentences:
Sentence 1: ”The cat sat on the hat”
Sentence 2: ”The dog ate the cat and the hat”

From these two sentences, our vocabulary is as follows: ”the, cat, sat, on, hat, dog, ate, and ”

To get our bags of words, we count the number of times each word occurs in each sentence.

In Sentence 1, ”the” appears twice, and ”cat”, ”sat”, ”on”, and ”hat” each appear once.

So the feature vector for Sentences are:
Sentence 1: ” 2, 1, 1, 1, 1, 0, 0, 0 ”
Sentence 2 : ” 3, 1, 0, 0, 1, 1, 1, 1 ”

In the data, we have a very large number of reviews, which will give us a large vocabulary. To limit the size of the feature vectors, we should choose some maximum vocabulary size. Below, we use the 500 most frequent words (remembering that stop words have already been removed).

We have used the feature extraction module from scikit-learn to create bag-of-words features. CountVectorizer Object in the feature extraction module is used. CountVectorizer converts a collection of text documents to a matrix of token counts. The analyzer is specified as a ”word” analyzer. If you do not provide an a-priori dictionary and you do not use an analyzer that does some kind of feature selection then the number of features will be equal to the vocabulary size found by analyzing the data. Fit-transform method is called on the CountVectorizer to learn the vocabulary dictionary and return term-document matrix.

Now every review is converted into a vector of the most of 500 most frequent words. This vector is used as features to build a regression model to predict the rating of the review.

B. Training and Testing data

To perform regression on the dataset and predict the reviews, we have to divide the dataset into training and testing data. In our analysis, I have randomly sampled 80 percent of the data as training data and the remaining as testing data. This is done using the cross validation module of the scikit-learn.

The train-test-split method of the cross validation module is used to split the data into training and testing data. We have to provide the data to predict as Y and the features useful for the prediction as X and the percent of testing data(in our case 0.2) to split the data. The output of the method is X-train, X-test, Y-train and Y-test.

C. Sentiment Analysis

A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. Advanced, ”beyond polarity” sentiment classification looks, for instance, at emotional states such as ”angry,” ”sad,” and ”happy.”Generally speaking, sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. The attitude may be a judgment or evaluation , affective state (that is to say, the emotional state of the author or speaker), or the intended emotional communication.

The Sentiment of the review is an important aspect in predicting the review. Hence, Sentiment score is calculated for each and every review and that is used as an additional feature in the regression model.

Sentiment Score is calculated using the Textblob library. TextBlob is an open source text processing library written in Python. It can be used to perform various natural language processing tasks such as part-of-speech tagging, noun-phrase extraction, sentiment analysis, text translation, and many more.

To calculate a sentiment score of a review, firstly a textblob is created on the review. The blob.sentiment method gives the polarity and the subjectivity of the review. Since we consider only polarity, we take blob.sentiment.polarity of the review. Using this technique sentiment scores of all the reviews is calculated.

D. Regression Model

Regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables that is, the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, the estimation target is a function of the independent variables called the regression function

The output we predict is review rating which is a nominal variable of 5 categories. The review rating can be only among the following: 1, 2, 3, 4 and 5. Hence a linear regression model can't be used to perform analysis. A multi-class prediction analysis, Multinomial Logistic Regression is used to predict the rating into one of the 5 categories.

Multinomial Logistic Regression is the linear regression analysis to conduct when the dependent variable is nominal with more than two levels. Thus it is an extension of logistic regression, which analyzes binary dependents. Since the output of the prediction analysis is somewhat different to the logistic regression's output, multinomial regression is sometimes used instead.

Like all linear regressions, the multinomial regression is a predictive analysis. Multinomial regression is used to describe data and to explain the relationship between one dependent nominal variable and one or more continuous-level(interval or ratio scale) independent variables.

Multinomial Logistic regression is implemented using the linear model module of the scikit-learn library. The LogisticRegression object is applied to implement this regression. We have to specify the multi-class parameter to be "multinomial" to apply Multonimial Logistic regression else a binary regression occurs. The weight parameter is set to be "balanced" since the balanced mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data.

The fit method is applied on the regression object to fit the model according to the given training data. The parameters passed are X-train and Y-train obtained after splitting data into training and testing.
The predict method labels for samples in X. The parameter passed is the X-test. This gives us the predicted ratings for the reviews in the X-test. This is our final output.

E. Spam/Inconsistent reviews

Online review can help people getting more information about store and product. The potential customers tend to make decision according to it. However, driven by profit, spammers post spurious reviews to mislead the customers by promoting or demoting target store.The opinion spam identification task has great impacts on industrial and academia communities. For sentiment analysis companies, if the opinion provided services contain large number of spams, they will affect the users experience. Furthermore, if the user is cheated by the provided opinion, he will never use the system again. For academic researchers, they have conducted various research studies on sentiment analysis tasks. If their acquired opinion resources contain many opinion spams, it is meaningless to provide any sentiment analysis results. Therefore, it is an essential task to identify and filter out the opinion spam. Previous studies mainly utilize rating as indicator for the detection. However, these studies ignore an important problem that the rating will not necessarily represent the sentiment accurately. We incorporate the sentiment analysis techniques into review spam detection.

The proposed method compute sentiment score from the natural language text by using textblob sentiment analysis. We propose 2 cases for the review to be spam/inconsistent:
Case1: If the sentiment polarity of the review is positive and the given rating is negative(1-3)
Case2: If the sentiment polarity of the review is negative and the given rating is positive(4-5)

IV. Results

We have developed the regression model on the features extracted from the bag-of-words approach and predicted the ratings of the reviews. We have also included sentiment score as a feature in predicting the reviews. There is a significant difference including the sentiment scores in many datasets.

We have evaluated our model using 3 measures:
Accuracy: Accuracy is often the starting point for analyzing the quality of a predictive model, as well as an obvious criterion for prediction. Accuracy measures the ratio of correct predictions to the total number of cases evaluated.
Mean Absolute Error: In statistics, the mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes.
Ranked Correlation Coefficient: The Spearman's rank-order correlation is the nonparametric version of the Pearson product-moment correlation. Spearman's correlation coefficient, (, also signified by rs) measures the strength and

direction of association between two ranked variables.

The Accuracies of the model without including sentiment as a feature are mentioned in Table1.

TABLE I
ACCURACY TABLE WITHOUT INCLUDING SENTIMENT SCORE

dataset	accuracy
AccuWeather	57.8125
Swiggy	63.28125
PayPal	61.71875
Lyft	54.6875

The Accuracies of the model using sentiment as a feature are mentioned in Table2.

TABLE II
ACCURACY TABLE WITH INCLUDING SENTIMENT SCORE

dataset	accuracy
AccuWeather	51.5625
Swiggy	61.71875
PayPal	63.28125
Lyft	54.6875

The MAE of the model without including sentiment as a feature are mentioned in Table3.

TABLE III
MAE TABLE WITHOUT INCLUDING SENTIMENT SCORE

dataset	MAE
AccuWeather	0.6875
Swiggy	0.703125
PayPal	0.8125
Lyft	0.953125

The MAE of the model including sentiment as a feature are mentioned in Table4.

TABLE IV
MAE TABLE INCLUDING SENTIMENT SCORE

dataset	MAE
AccuWeather	0.71875
Swiggy	0.734375
PayPal	0.7421875
Lyft	0.890625

The Ranked Correlation values of the model without including sentiment as a feature are mentioned in Table5.

The Ranked Correlation values of the model including sentiment as a feature are mentioned in Table6.

In the Spam/Inconsistent reviews detection, we have discussed the cases in which we have identified the review to be spam/inconsistent review. So performing above mentioned detection on the reviews of the the datasets, we have identified the inconsistent reviews in all the datasets.

TABLE V
CORRELATION TABLE WITHOUT INCLUDING SENTIMENT SCORE

dataset	Coefficient
AccuWeather	0.399
Swiggy	0.688
PayPal	0.401
Lyft	0.523

TABLE VI
CORRELATION TABLE INCLUDING SENTIMENT SCORE

dataset	Coefficient
AccuWeather	0.338
Swiggy	0.666
PayPal	0.502
Lyft	0.566

The percent of spam reviews in each of the datasets is mentioned in Table7.

TABLE VII
PERCENTAGE OF SPAM REVIEWS

dataset	spam percentage
AccuWeather	8.76
Swiggy	13.9
PayPal	9.53
Lyft	16.71

V. CONCLUSIONS

A regression model is built on the features extracted from vectorization of the review content. Sentiment Analysis is used to detect spam/inconsistent reviews. The importance of sentiment score in predicting the review is explained using sentiment score as a feature to predict the review.

From the results, we can say that sometimes the sentiment score inclusion has resulted in higher accuracy but in many cases, it reduced the accuracy of the prediction and may not be a correct feature in this model to be applied. The spam detection algo identified around 10 percent of the reviews to be spam.

ACKNOWLEDGMENT

I would like to thank Dr.Tung Nguyen for his patience and support in overcoming numerous obstacles I have been facing through my project in Software Analytics. The Assignments and the lectures of the Software Analytics course have helped me understand the concepts better and have helped me in completing the project successfully.

REFERENCES

[1] http://blog.christianperone.com/
[2] Wikipedia
[3] http://textminingonline.com/getting-started-with-textblob