A Business Relevant Sentimental Analysis Project

Kapil Rajoria (1BC18CS008)

Bangalore College of Engineering and Technology
Chandapura, Bengaluru, 560099
rajoria.kapil456@gmail.com

***Abstract*— This document gives full details on how to implement a business sentiment-based binary classification (positive / negative) model for customer reviews received on business' facebook page. This model can be used to predict whether a review is a positive review or a negative review. We are given two files by a restaurant, one is the historical reviews which is the past data in .tsv format with 900 entries and with each review in the table there is a liked value. liked = 0 means a negative review and liked = 1 means a positive review. Another file is the fresh reviews in .tsv format with 100 entries, for which we have to do the predictions i.e finding out the liked value using our model. We will be using Gaussian Naive Bayes machine learning classifier algorithm for our model which is a supervised machine learning algorithm. We will be using purely python programming language to implement our model. On completion of this model, the business will be able to resolve the problems by seeing the negative reviews which are automatically classified by the model to improve their business and make changes accordingly.**

***Keywords*— Sentiment analysis, Python, Machine learning, reviews, predictions**

I. UNDERSTANDING THE PROBLEM

[1]One of the greatest entrepreneur of all time, Bill Gates has said "Your most unhappy customers are your greatest source of learning" which means listening to customers is rewarding for the business. With the power of internet today businesses get a lot of reviews through their website, apps and social media handles from their customers. These feedback can be used to improve the performance of the business and which can further be used to gain more profit. But the problem is most of the businesses do not use the user feedback. In fact the big companies like Google, Amazon, Zomato uses structured feedback. In the structured feedback there is a rating bar of stars along with a text field for the review which makes it easy to classify a review as negative or positive.

The real volume lies in the unstructured feedback which are often neglected. These are the plain text feedback which are being shared by a lot of people on facebook and instagram. The problem is these feedback are unstructured. Since, there is no rating bar one has to go through each review to determine the sentiment of the review.

To work with these unstructured feedback, we can take the help of machine learning. NLP (Natural Language Processing) based sentiment analysis technique can predict sentiments for such unstructured reviews at large scale. It allows businesses to analyze customer feedback at large scale and resolve customer complaints so that they make assure that a unhappy customer also revisits.

II. CLIENT'S BUSINESS CASE DATA SET

A restaurant intends to build a binary classification model (positive / negative) for customer reviews on their facebook page. Positive reviews are appreciations and negative reviews are criticism for the restaurant.

Business intends to build an in house customer support team to call back customers who gave negative feedback and resolve their issues and complaints and offer them discounts, ensuring they revisit.

Restaurant has shared following datasets :

1. Historical Reviews along with labels
2. Fresh Reviews without labels

Restaurant intends to generate labels for the fresh reviews.

[1]Historical Review Data Set :
A file in the .tsv format containing a table made up of 3 columns. First is index column, second one is review column which contains the reviews and the third one is label column named as liked which consists of zeroes and ones. A zero means a negative review and one means a positive review. It contains a total of 900 entries. The head of the file looks something like this (Ref. Figure 1). The name of the file is "a1_RestaurantReviews_HistoricDump.tsv"

	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday of...	1
4	The selection on the menu was great and so wer...	1

Figure 1: Head of the Historical Review Data set file

[1]Fresh Review Data set:
A file in the .tsv format containing a table made up of 2 columns. First is index column and second one is review column which contains the reviews. For every review here, we need to classify it in 0 or 1 category that is a positive review or a negative review. It contains a total of 100 entries. The head of the file looks something like this (Ref. Figure 2). The nameof the file is "a1_RestaurantReviews_HistoricDump.tsv"

	Review
0	Spend your money elsewhere.
1	Their regular toasted bread was equally satisf...
2	The Buffet at Bellagio was far from what I ant...
3	And the drinks are WEAK, people!
4	-My order was not correct.

Figure 2: Head of the Fresh Review Data set file

III. SOLUTIONING INTUITION

First step towards any machine learning model is data preparation i.e cleaning of the data so that computer can easily understand it. It is done in 5 major steps:

A. The original reviews

It consists of the original reviews which are yet to be cleaned. Let's consider the first 6 reviews (Ref. Figure 3).

1. Wow... Loved this place.
2. Crust is not good.
3. Not tasty and the texture was just nasty.
4. Also there are combos like a burger, fries, and beer for 23 which is a decent deal
5. I would definitely recommend the wings as well as the pizza.
6. Highly recommended.

Figure 3: The original reviews

B. Dropping the special characters like , : % ! etc.

From every review we will drop the special characters. It is done using the re module in python which stands for regular expressions. Take a look at the figure 4. The parts are highlighted in red which will be dropped from the reviews (Ref. Figure 4).

1. Wow... Loved this place.
2. Crust is not good.
3. Not tasty and the texture was just nasty.
4. Also there are combos like a burger, fries, and beer for 23 which is a decent deal.
5. I would definitely recommend the wings as well as the pizza.
6. Highly recommended.

Figure 4: Dropping special characters

C. Converting all letters to lower case

In this step, we will convert all the letters in each and every review to lower case. Since, computer understands lowercase and uppercase differently this step becomes very important. For a computer "not" is not equal to "Not" (Ref. Figure 5). For this, we will be using lower( ) function in python.

1. Wow Loved this place
2. Crust is not good
3. Not tasty and the texture was just nasty
4. Also there are combos like a burger fries and beer for which is a decent deal
5. I would definitely recommend the wings as well as the pizza
6. Highly recommended

Figure 5: Converting letters to lower case

D. Dropping stop words and stemming

[3]Stop words are those words which are not useful in the prediction of sentiment or the words which doesn't have a meaning. For example: the, or, I, am etc. We will remove the stop words using the nltk module in python. The red highlighted words are stop words (Ref. Figure 6).

[3]Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as lemma. We will do the stemming of each and every review using the nltk module in python. For example: The green highlighted words need to be stemmed (Ref. Figure 6).

Stop words dropping and stemming are both important steps in machine learning.

1. wow loved this place
2. crust is not good
3. not tasty and the texture was just nasty
4. also there are combos like a burger fries and beer for which is a decent deal
5. i would definitely recommend the wings as well as the pizza
6. highly recommended

Figure 6: Dropping stop words and stemming

After dropping the stop words and stemming, our reviews data is prepared for the further processing. The final data looks like this (Ref. Figure 7).

1. wow love place
2. crust not good
3. not tasti textur nasti
4. also combo like burger fri beer decent deal
5. would definit recommend wing well pizza
6. highli recommend

Figure 7: Cleaned data

E. Preparing the data further for model

In the above steps we have cleaned the data and also the number of words are reduced in the data set. But computer is still unable to understand these. So we have to some how convert these words into numbers specifically 0 and 1. For that we will convert this cleaned data set to a bag of words representation through transformation.

Let's take the first 3 data sets (Ref. Figure 8) for transformation to bag of words representation. System would simply identify all unique words also called as tokens in the review column and for separate columns for each of tokens like this (Ref. Figure 9). All these tokens collectively are called model dictionary.

Review	Liked
wow love place	1
crust not good	0
not tasti textur nasti	0

Figure 8: taking first 3 cleaned reviews for bag of words conversion

wow	love	place	crust	not	good	tasti	textur	nasti	Liked
1	1	1	0	0	0	0	0	0	1
0	0	0	1	1	1	0	0	0	0
0	0	0	0	1	0	1	1	1	0

Figure 9: Bag of words representation

For every review, system puts 1 if that token is present in the review or 0 if that token is absent. Bag of words representation discards information on order and sequencing of words. Dropping tokens or we can say unique words that only reflects in a few reviews, reduces sparsity.

We will be using sklearn module of python to make a bag of words.

IV. MACHINE LEARNING MODEL

We will be using Gaussian Naive Byes classifier for our sentimental analysis model.

How Gaussian Naive Bayes works:

Let's say we have a new review - "This place is wow!!". After cleaning it, it becomes "place wow". If we add the new review in the bag of words representation (Ref. Figure 10).

wow	love	place	crust	not	good	tasti	textur	nasti	Liked
1	1	1	0	0	0	0	0	0	1
0	0	0	1	1	1	0	0	0	0
0	0	0	0	1	0	1	1	1	0
1	0	1	0	0	0	0	0	0	?

Figure 10: new example review added

For classification, Gaussian Naive Bayes uses conditional probability (Ref. Figure 11).

Model Prediction for Liked = Max (Prob(+ve | [101000000]), P(-ve| [101000000]))

Figure 11: conditional probability

[1]As we can see in the historic review data, whenever the token wow is present, the review is always positive. So if wow is present again, it predicts a positive sentiment I.e liked = 1. Similarly when token love is absent, review is always negative which can be used to predict a negative sentiment I.e liked = 0. For some words like crust, it is 50-50 chances that it is a negative sentiment or a positive sentiment. In some cases in historic data review having a crust token represents a positive sentiment while in some cases it represents a negative sentiment. Similarly we do this for all tokens analyzing their conditional probabilities.

Therefore, after analyzing the conditional probabilities of the tokens "wow" and "place" from the given historic data set we can predict that "wow place" is a positive sentiment i.e liked = 1 for it.

This is exactly how Gaussian Naive Bayes classifies reviews for us.

We will be using Sklearn module of python to directly import the Gaussian Naive bayes classifier and also for all the other operations used in machine learning like splitting the data into test data and training data, confusion matrix, making predictions, finding accuracy of the model.

RESULTS

After running the model an accuracy of 72.777% is achieved which implies that our model is quite good in classifying the reviews sentiment as negatives and positives.

DISCUSSIONS AND CONCLUSION

We came to know that the unstructured data which is often ignored can also be used to predict the sentiment of reviews. The business can now focus on those specific customers easily without the need of a human who gave negative reviews. Further, the business can improve by calling those customers and understand their issues and complaints and take actions accordingly. Business can give them discounts and assure that they revisit.

REFERENCES

[1] https://skillcate.com/
[2] https://statquest.org/studyguides/
[3] https://www.nltk.org/
[4] https://docs.python.org/3/library/re.html
[5] https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes
[6] https://numpy.org/
[7] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html
[8] https://www.nltk.org/api/nltk.corpus.html
[9] https://www.geeksforgeeks.org/python-stemming-words-with-nltk/
[10] *https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html*
[11] https://towardsdatascience.com/what-and-why-behind-fit-transform-vs-transform-in-scikit-learn-78f915cf96fe
[12] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.iloc.html
[13] https://docs.python.org/3/library/pickle.html#:~:text=%E2%80%9CPickling%E2%80%9D%20is%20the%20process%20whereby,back%20into%20an%20object%20hierarchy.
[14] *https://joblib.readthedocs.io/en/latest/*