Sentiment Analysis and Machine Learning on Yelp Reviews

Eric John Pozholiparambil
Northeastern University, Boston
pozholiparambil.e@husky.neu.edu

Rishi Raj Dutta
Northeastern University, Boston
dutta.r@husky.neu.edu

ABSTRACT

We have performed sentiment analysis on reviews given by user on different businesses that are on Yelp. On using the Yelp dataset, we encountered that there were many reviews that did not have star ratings and filled with null values. This is not just the case with Yelp dataset but we see the presence of null values in all types of dataset across the world. We planned to tackle the null value exception by creating a machine learning model by training the model with the remaining dictionary in our dataset and generating a prediction for the sentiment of the reviews given by the users that contained null star ratings. In order to implement this, we used Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes and Logistic Regression to train out model. Each model will be used to predict the sentiment of the user text for null values based on the train and test datasets created. Our approach towards this led us to refer a research paper published by Boya Yu, Jiaxu Zhou, Yi Zhang and Yunong Cao who proposed a method called Support Vector Machine (SVM) model to decipher the sentiment tendency of each review from word frequency. Word scores generated from the SVM models are further processed into a polarity index indicating the significance of each word for special types of restaurant. We inferred that customers overall tend to express more sentiment regarding service offered in a restaurant that the food.

INTRODUCTION

Yelp is an American multinational corporation founded in 2004 which aimed at helping people locate local business based on social networking functionally and reviews. The main purpose of Yelp is to provide a platform for customers to write review along with providing a star-rating along with an open-ended comment. Yelp data is reliable, up-to-date and has a wide coverage of all kinds of businesses. Millions of people use yelp and empirical data demonstrated that Yelp restaurant reviews affected consumers' food choice decision-making; a one-star increase led to 59% increase in revenue of independent restaurants (Luca, 2011). With the rapid growth of visitors and users, we see great potential of Yelp restaurant reviews dataset as a valuable insights repository.

Microblogging websites have evolved to become a source of varied kind of information. This is due to nature of microblogs on which people post real time messages about their opinions on a variety of topics, discuss user experience, complain, and express positive sentiment for products and services they use in daily life. In fact, companies manufacturing such products have started to poll these microblogs to get a sense of general sentiment for their product.

Yelp serves as a platform to express the user's sentiment and opinion. With the increasing popularity of Yelp and the impact it incurs on the various businesses associated with Yelp due to the plethora of reviews. It is challenging to build a technology to detect and summarize an overall sentiment of all the reviews. We will take an approach to create a machine learning model based on the reviews in our dataset and then try to predict the sentiment of a review in real time based on the learning potential of the model.

CODE OF DOCUMENTATION

The Dataset that we have used is Yelp Dataset - A trove of reviews, businesses, users, tips, and check-in data from kaggle.com. The dataset contains six comma separator value files containing details of the users, businesses, business attributes, business hours, check-in, tips and reviews. The dataset contains about five million rows. Link to the dataset: https://www.kaggle.com/yelp-dataset/yelp-dataset. Below is a data model of the project flow which describes the steps taken to reach the sentiment prediction for the null values.

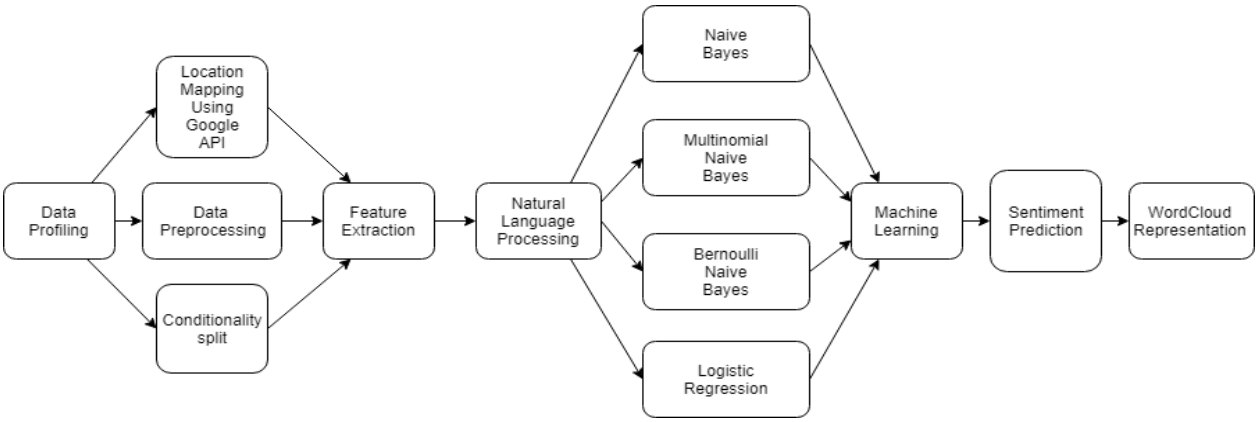

Fig:1

A pictorial process flow model of what we are doing, (as per Fig:1): -

Data Profiling - Cleaning of the data and Exploratory Data Analysis.

Data preprocessing - Understanding the numeric and textual components of the dataset with distribution histogram and scatter plots.

Conditionality Split - reviews with star ratings (3,2,1) --> neg [0] || [1] pos <-- (4,5). Helps us to convert the reviews into binary/Boolean values for data analysis.

Location Mapping - Gives us a demographic understanding of the negative reviews with the help of Google API.

Feature Extraction - Extracting all the key words that contribute to sentiment analysis excluding stop words, upper cases and conjunction words.

Natural Language Processing - Generate sentiment confidence of feature words by assigning weights according to their occurrence.

Classifier - We have used four classifiers to generate sentiment accuracy i.e. Naive Bayes Classifier, Multinomial Naive Bayes Classifier, Bernoulli Naive Bayes Classifier, Logistic Regression.

Machine Learning - From the sentiment accuracy derived we train the model to predict sentiment of a review.

Sentiment Prediction - The machine learning model predicts the sentiment of a review given by a user in real time on the model.

WordCloud - Gives us a representation of all the positive and negative words predicted by the model.

We performed exploratory data analysis with the help of open python three environment tool kit. This approach generated distribution of all the numeric values that are there in the dataset as these numeric values were used in the classifier.

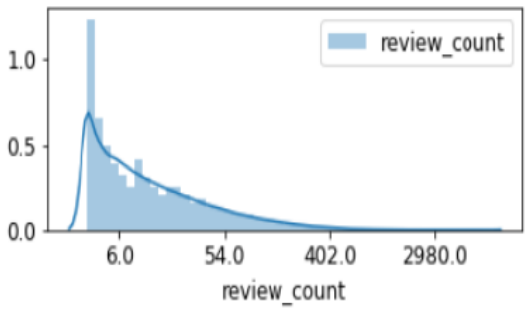

Fig:2

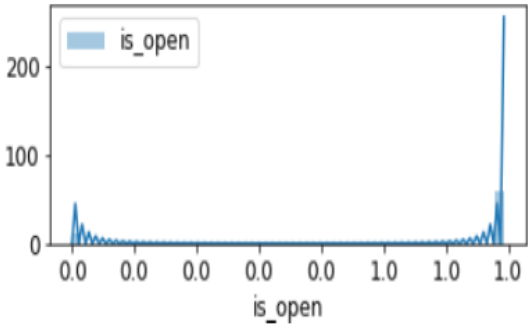

Fig:3

The above diagrams (Fig:2) and (Fig:3) gives us an understanding of all the users who are coming back again and giving their reviews over all the first-time new users because repeating users reviews create an increase and decrease in revenue and impacts the performance of the business.

METHODS:

Data Description

The Dataset that we have used is Yelp Dataset - A trove of reviews, businesses, users, tips, and check-in data from kaggle.com. The dataset contains six comma separator value files containing details of the users, businesses, business attributes, business hours, check-in, tips and reviews.

The dataset contains about five million rows and contains 1,100,000 tips by 1,300,000 users. Over 1.2 million business attributes like hours, parking, availability, and ambience. Aggregated check-ins over time for each of the 174,000 businesses

In the below diagram (as per Fig:4), we have extracted the latitudes and longitude values from the given dataset and used Google API maps to display the highly negative reviews.

Below is our own private Google API key generated from Google:

Link to Google API key - "https://developers.google.com/maps/documentation/javascript/get-api-key"

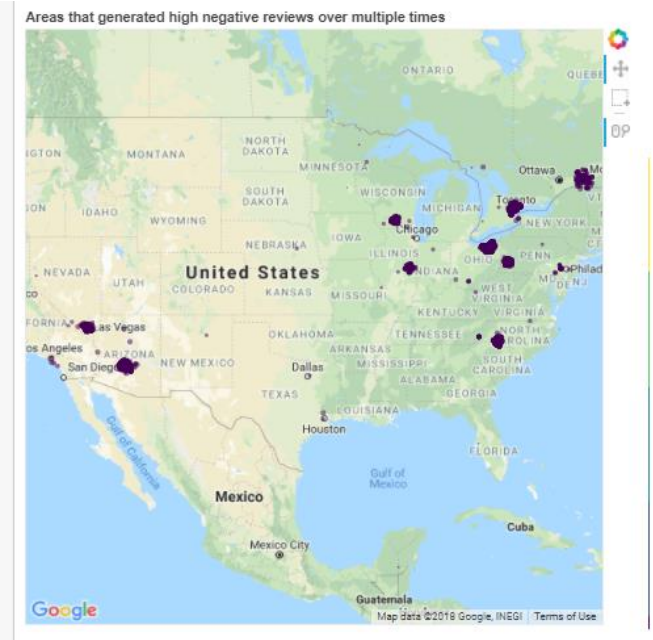

Fig:4

1] Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. Given a class variable y and a dependent feature vector $x_1$through x_n, Bayes' theorem states the following relationship:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

After applying conditionality split, which converting stars 1,2 and 3 to negative and 4 and 5 to positive, we perform cleaning on the two variables {(senti), (check)}. Further we created Test and Train datasets for the cleaned-up attribute (Summary_Clean) and we apply Feature Extraction which will be used in NLTK Naïve Bayes Classifier. Using the above-mentioned equation, we calculated an accuracy of 0.60 which is equivalent

to 60% accuracy through Naïve Bayes.

Further we built count vector and tf-idf vector for matrix generation of train, test and check data.

tf - term frequency: It is computed by dividing the number of times a term occurs in the document by the total number of terms in the document. This division by the document length prevents a bias towards longer document by normalizing the raw frequency of the term into a comparable scale.

idf - inverse document frequency: It is computed by taking the logarithmic of the total number of documents in the corpus divided by the number of documents where the term occurred. This normalization is to up-weigh the rare terms in the corpus).

2] Multinomial Naïve Bayes

MultinomialNB implements the Naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). The distribution is parametrized by vectors $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θ_{yi} is the probability $P(x_i \mid y)$ of feature i appearing in a sample belonging to class y. The parameters θ_y is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

where $N_{yi} = \sum_{x \in T} x_i$ is the number of times feature i appears in a sample of class y in the training set T, and $N_y = \sum_{i=1}^{|T|} N_{yi}$ is the total count of all features for class y.

Applied Multinomial Naive Bayes Classifier to predict the sentiment for check which contained null ratings. It divides the extracted feature words into individual tokens and calculates the occurrence count of each token then generates the sentiment result for each token. Advantages are such that it runs faster than Naive Bayes. This works well for data which can easily be turned into counts, such as word counts in texts. Gives a higher sentiment accuracy model. Using the above-mentioned equation, we calculated an accuracy of 0.64 which is equivalent to 64% accuracy through Multinomial Naïve Bayes.

3] Bernoulli Naïve Bayes

BernoulliNB implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, Boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a BernoulliNB instance may binarize its input (depending on the binarize parameter).

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

The decision rule for Bernoulli naive Bayes is based on which differs from multinomial NB's rule in that it explicitly penalizes the non-occurrence of a feature i that is an indicator for class y, where the multinomial variant would simply ignore a non-occurring feature. Here we implement Bernoulli Naive Bayes Classifier for sentiment prediction on check data which has null ratings. The difference is that while MultinomialNB works with occurrence counts, BernoulliNB is designed for binary/Boolean features.

Advantages using this method is that it runs faster than Naïve Bayes and Multinomial Naïve Bayes and it also generates a better accuracy as compared to Naïve Bayes and Multinomial Naïve Bayes. Using the above equation, we calculated an accuracy of 0.72 which is equivalent to 72% accuracy through Bernoulli Naïve Bayes.

4] Logistic Regression

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross- entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag' and 'newton-cg' solvers.)

This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag' and 'lbfgs' solvers. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

Logistic Regression turns out to be the most useful model as compared to the previous three models. Using Logistic Regression, we calculated an accuracy of 0.87 which is equivalent to 87% of accuracy.

we get the following accuracy, Naïve Bayes = 0.60, Multinomial Naïve Bayes = 0.64, Bernoulli Naïve Bayes = 0.72 and Logistic Regression = 0.87.

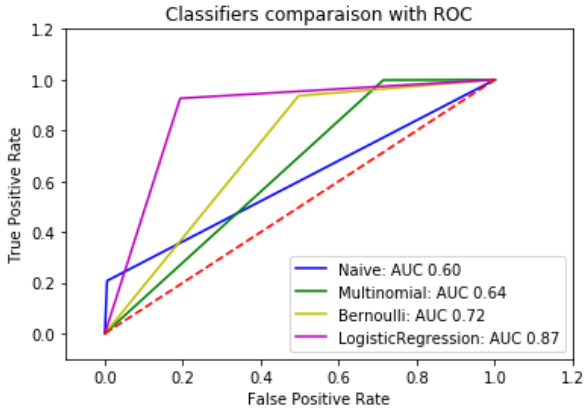

Fig:6

In the above diagram (Fig:5), we have created table having the following columns:
Stars- List of Null values in the dataset.
Text- List of Text reviews.
user_id- List of User IDs

Out[79]:

	stars	text	user_id	Summary_Clean	words	Naive	multi	Bill	log
0	NaN	Love coming here. Yes the place always needs t...	bv2nCi5Qv5vroFiqKGopiw	love coming here yes the place always needs th...	[love, coming, here, yes, the, place, always, ...	neg	pos	pos	pos
1	NaN	Came here with my girlfriends one Sunday after...	u0LXt3Uea_GidxRW1xcsfg	came here with my girlfriends one sunday after...	[came, here, with, my, girlfriends, one, sunda...	neg	pos	pos	pos
2	NaN	worse customer service ever. \r\nManager on du...	u0LXt3Uea_GidxRW1xcsfg	worse customer service ever manager on duty wa...	[worse, customer, service, ever, manager, on, ...	neg	neg	neg	neg
3	NaN	Small little Japanese restaurant in the Don Mi...	u0LXt3Uea_GidxRW1xcsfg	small little japanese restaurant in the don mi...	[small, little, japanese, restaurant, in, the,...	neg	pos	pos	pos
4	NaN	Visiting from SF. Checked yelp and found this...	_L2SZSwf7A6YSrIHy_q4cw	visiting from sf checked yelp and found this p...	[visiting, from, sf, checked, yelp, and, found...	pos	pos	pos	pos
5	NaN	After being scared away from Rock & Rita's, we...	nOTI4aPC4tKHK35T3bNauQ	after being scared away from rock rita s we en...	[after, being, scared, away, from, rock, rita,...	neg	neg	neg	neg
6	NaN	So, below is is original review, which was acc...	nOTI4aPC4tKHK35T3bNauQ	so below is is original review which was accom...	[so, below, is, is, original, review, which, w...	neg	pos	neg	pos
7	NaN	I've visited this place on and off since the 7...	tL2pS5UOmN6aAOi3Z-qFGg	i ve visited this place on and off since the s...	[i, ve, visited, this, place, on, and, off, si...	neg	pos	pos	neg
8	NaN	Stopped in here yesterday for lunch. I wasn't ...	tL2pS5UOmN6aAOi3Z-qFGg	stopped in here yesterday for lunch i wasn t e...	[stopped, in, here, yesterday, for, lunch, i, ...	neg	pos	neg	neg
9	NaN	This is one huge casino. I've been in here man...	tL2pS5UOmN6aAOi3Z-qFGg	this is one huge casino i ve been in here many...	[this, is, one, huge, casino, i, ve, been, in,...	neg	pos	pos	pos

Fig:5

RESULTS:

With respect to the below diagram (Fig:6), after applying Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes and Logistic Regression

Summary_Clean- Cleaned text reviews
words- Tokenize the cleaned words
Naïve- Sentiment generated by Naïve Bayes.
multi- Sentiment generated by Multinomial Naïve Bayes.

Bill- Sentiment generated by Bernoulli Naïve Bayes.
log- Sentiment generated by Logistic Regression.

CONCLUSION:

We derived that reviews with repetitively occurring feature words which have higher polarity index tend to generate an increasingly accurate prediction of star rating for rows which earlier had null values for star ratings. From our model we were able to infer that the sentiment value of logistic regression classifier produced the most accurate star rating prediction for the reviews based on weight frequency of the feature words presents in the form of tokens in the text. From figure, we infer how each sentiment column of the classifier corresponds to the review of the user and ideally it seems that Logistic Regression has a higher chance of predicting the text review accurately as compared to any other above-mentioned model. Although we found one drawback in using logistic regression, which is that it compromises on the execution time for better results, while the other models execute significantly faster than logistic regression.

Fig:7

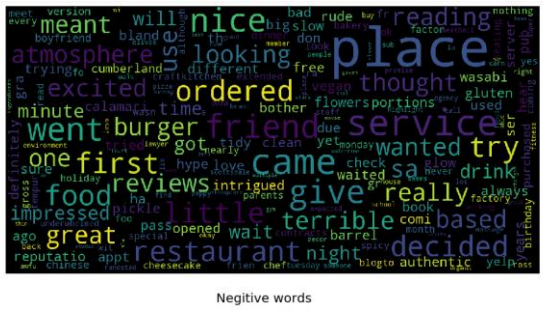

Fig:8

In the above diagrams (Fig:7 and Fig:8) we have used WordCloud to represent positive values and negative values respectively. This helps us to understand the final model that we generated in a better fashion.

DISCUSSIONS:

Since right now, the data set is not that large for this approach, therefore this data set can be increased to have more variation in sentiments which can improve the results. Also, more data preprocessing steps such as data stemming and word repetitions can be implemented to increase accuracy of the model. We are currently working on extending the sentiment values from 2 (pos, neg) to 5 (very neg, neg, neutral, pos, very pos). We are trying to incorporate an extra classifier model against our model to see the comparison in the accuracy and loss of prediction, thereby improving the machine learning experience for the user. Also, we would modify certain hyper-parameters within our model to see any significant difference within the accuracy and loss of our prediction.

REFERENCES:

[1] Boya Yu, Jiaxu Zhou, Yi Zhang, Yunong Cao (2017). Identifying Restaurant Features via Sentiment Analysis on Yelp Reviews. Center for Urban Science & Progress New York University New York, NY, the United States

[2] Hicks, A., Comp, S., Horovitz, J., Hovarter, M., Miki, M., & Bevan, J. L. (2012). Why people use Yelp. com: An exploration of uses and gratifications. Computers in Human Behavior, 28(6), 2274-2279.

[3] D. M. Freeman, “Using naive bayes to detect spammy names in social networks,” in Proceedings of the 2013 ACM workshop on Artificial intelligence and security. ACM, 2013, pp. 3–12.

[4] Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. S. (2013, July). What yelp fake review filter might be doing?. In ICWSM.

[5] Ariyasriwatana, W., Buente, W., Oshiro, M., & Streveler, D. (2014). Categorizing health-related cues to action: using Yelp reviews of restaurants in Hawaii. New Review of Hypermedia and Multimedia, 20(4), 317-340

[6] For Geo Map : http://www.bigendiandata.com/2017-06-27-Mapping_in_Jupyter/