SENTIMENT ANALYSIS ON RESTAURANT REVIEWS FROM YELP DATASET

2/22/2021

By Priyanka Ball & Shubhneet Grover
pball@saintpeters.edu, sgrover@saintpeters.edu

Contents

Abstract 2

Introduction 2

Data 3

Methodology: Sentiment Analysis 3

Analysis 4

- Step by Step workflow 4
- Results from Exploratory Data Analysis 5
  - N Gram Analysis 6
  - Feature Engineering 7

Regression models 7

- Naïve baes 7
- Random forest 8
- Logistic regression 8
- The Training and Prediction Processes 8
- Best Model 9

Conclusion 10

References 10

Abstract

In the world of the internet, customers tend to research and analyze reviews before deciding which restaurant to visit. For businesses, it’s hard for them to gauge customer satisfaction, as customer’s prefer providing reviews online and due to the large prevalence of the Internet, the number of reviews can sometimes be unfathomable for a business to review individually, especially from websites like Yelp.

Reviews are a vital source of feedback loop that the business utilizes to stay current and heed to the customer suggestions, which can attract an even larger customer base. This can lead to higher profit for the business and help them grow. For this project, we have analyzed Yelp reviews of restaurants and find out the factors that are leading to positive and negative views. We have then performed different classification techniques to train the model to predict if a review is positive or negative. Finally we have deployed the model using Flask.

Introduction

Customers are increasingly providing reviews online and have become very important for customers to read reviews from other customers before choosing a restaurant. As the frequency of the reviews have increased, businesses have faced difficulty analyzing every review for their qualitative information. Reviews help business in the following ways:

- Social Proof Drives Purchases.
- Reviews make the business more visible.
- Reviews make business look trustworthy.
- They expand the conversation about your business.
- Reviews are increasingly essential to decision making.
- Reviews have a clear impact on sales for a business.
- Provides an open line to consumers.

The project tried to make the process easy for the businesses to analyze the reviews. The project will begin with preprocessing the data and analyzing the key factors affecting restaurant reviews.Classification models were utilized to predict if a review is good or bad. Model was trained on Yelp Reviews that had ratings from 1-5. Reviews that have 4 or greater stars out of 5 will be considered good while reviews below 4 will be considered as bad.

Data

Data was collected through Yelp. Business and Review datasets were downloaded from the site. The dataset was downloaded as a json file so we had to turn it into a dataframe. We found the dataset to be very big so decided to use the first 10000 rows of the data. The business and review datasets were merged based on business_id. We decided to filter for restaurant data only and include reviews of business that are currently open. After joining both datasets, we decided to leave out unnecessary columns that would not add any value to our analysis. The final dataset included "text", "business_id", "name", "state", "business_stars", "categories", "review_id", "review_stars". We have mostly used text data for this analysis, but have used other information to have a better understanding of where the data is coming from.

Methodology

Sentiment analysis is the process of detecting positive or negative sentiment in text. It's often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers. Since customers express their thoughts and feelings more openly than ever before, sentiment analysis is becoming an essential tool to monitor and understand that sentiment. Automatically analyzing customer feedback, such as opinions in survey responses and social media conversations, allows brands to learn what makes customers happy or frustrated, so that they can tailor products and services to meet their customers' needs.

Sentiment analysis models focus on polarity (positive, negative, neutral) but also on feelings and emotions (angry, happy, sad, etc), urgency (urgent, not urgent) and even intentions (interested v. not interested). Depending on how you want to interpret customer feedback and queries, you can define and tailor your categories to meet your sentiment analysis needs.

Sentiment analysis, otherwise known as opinion mining, works thanks to natural language processing (NLP) and machine learning algorithms, to automatically determine the emotional tone behind online conversations. There are different algorithms you can implement in sentiment analysis models, depending on how much data you need to analyze, and how accurate you need your model to be. Sentiment analysis algorithms fall into one of three buckets:

- ○ Rule-based: these systems automatically perform sentiment analysis based on a set of manually crafted rules.
- ○ Automatic: systems rely on machine learning techniques to learn from data.
- ○ Hybrid: systems combine both rule-based and automatic approaches.

For this project we have mostly used classification algorithms which fall under automatic. Before using the models, we have cleaned and pre-processed the text data using various techniques available in NLP. Once the data was cleaned, we tested out with Naive Bayes, Logistic Regression and Random Forest model. After picking the best model, we used hyper parameter tuning techniques to improve the model further. The model was also tested by increasing the sample size.

Analysis

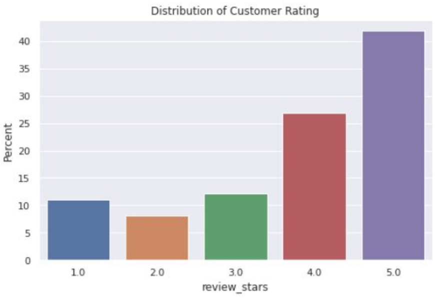
Most of the analysis of the data was on the text data. However, we have also explored other things in the datasets to have more context around where the review data was coming from.

Step by Step workflow

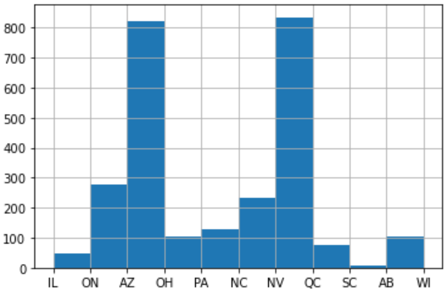
- **STEP 1**: Downloaded business and review data from Kaggle YELP Review. Joined both datasets. Data was filtered to include reviews from restaurants that are open.
- **STEP 2**: Selected columns that are necessary for our analysis.
- **STEP 3**: Filtered data to include reviews written in English. We found the data to include reviews from 6 different languages.
- **STEP 4:** Labelled reviews as positive and negative based on review rating.
- **STEP 5**: Explored text data to find differences between positive and negative reviews, common words (bi-gram, tri grams), common adjectives and nouns.
- **STEP 6**: Explored text data to find patterns in the data and identify necessary steps for cleaning data and standardizing the data. We are standardizing the data by lowercasing, removing stop words, replacing contracted words.
- **STEP 7**: Tokenized each word and used lemmatization methods.
- **STEP 8**: Use Bag of Words transformation to incorporate into Classification ML Models. We have also tested models with TF-IDF transformation
- **STEP 9:** Segment data into test and train
- **STEP 10**: Test with different models: Naive-Bayes, Random Forest, Logistic Regression.
- **Step 11:** Select model with best output and use hyper parameter tuning techniques to improve the model further.
- **Step 12:** Deploy model using Flask

Results from Exploratory Data Analysis

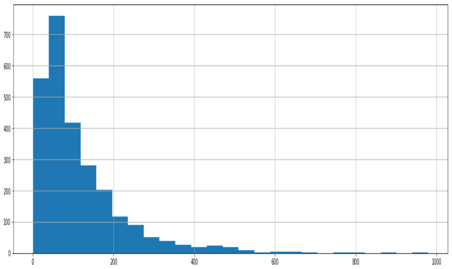
In this Section, the project explores data and gathers more information about the data set. The aim is to identify and understand underlying information.



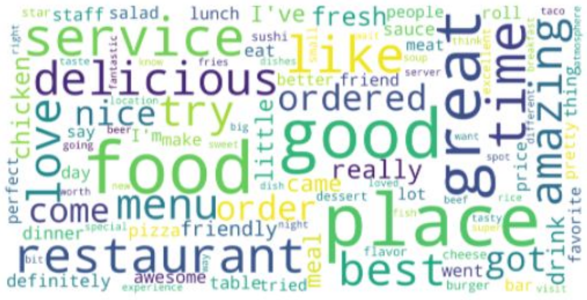
The dataset is skewed as it has more positive review stars.



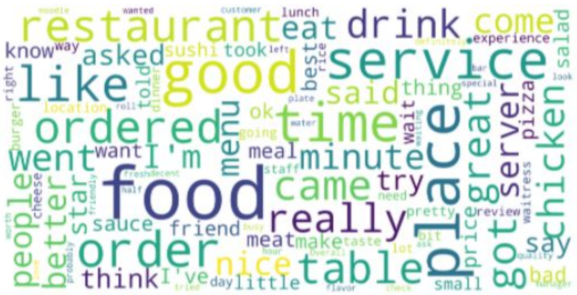
Dataset included reviews from 11 different regions



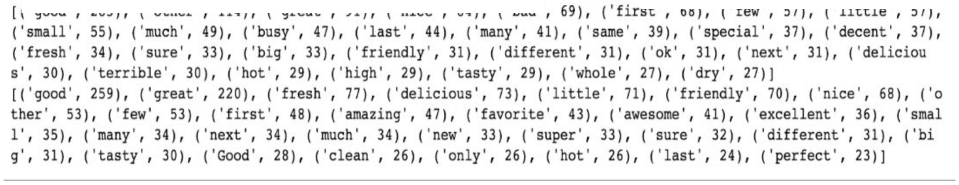
Most reviews were around 100 words



Word Cloud with 4- and 5-star rating reviews

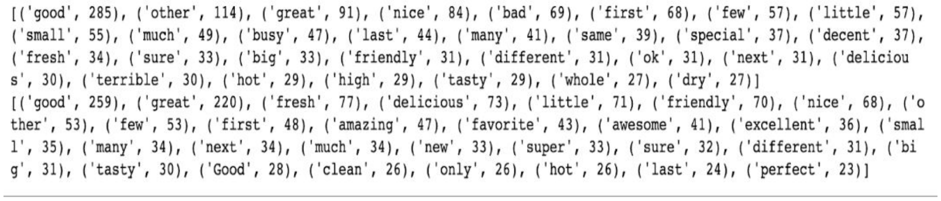


Word Cloud with 1, 2, 3-star rating reviews



```
[('good', 205), ('other', 114), ('great', 91), ('nice', 84), ('bad', 69), ('first', 68), ('few', 57), ('little', 57),
('small', 55), ('much', 49), ('busy', 47), ('last', 44), ('many', 41), ('same', 39), ('special', 37), ('decent', 37),
('fresh', 34), ('sure', 33), ('big', 33), ('friendly', 31), ('different', 31), ('ok', 31), ('next', 31), ('deliciou
s', 30), ('terrible', 30), ('hot', 29), ('high', 29), ('tasty', 29), ('whole', 27), ('dry', 27)]
[('good', 259), ('great', 220), ('fresh', 77), ('delicious', 73), ('little', 71), ('friendly', 70), ('nice', 68), ('o
ther', 53), ('few', 53), ('first', 48), ('amazing', 47), ('favorite', 43), ('awesome', 41), ('excellent', 36), ('smal
l', 35), ('many', 34), ('next', 34), ('much', 34), ('new', 33), ('super', 33), ('sure', 32), ('different', 31), ('bi
g', 31), ('tasty', 30), ('Good', 28), ('clean', 26), ('only', 26), ('hot', 26), ('last', 24), ('perfect', 23)]
```

Negative Reviews



```
[('good', 285), ('other', 114), ('great', 91), ('nice', 84), ('bad', 69), ('first', 68), ('few', 57), ('little', 57),
('small', 55), ('much', 49), ('busy', 47), ('last', 44), ('many', 41), ('same', 39), ('special', 37), ('decent', 37),
('fresh', 34), ('sure', 33), ('big', 33), ('friendly', 31), ('different', 31), ('ok', 31), ('next', 31), ('deliciou
s', 30), ('terrible', 30), ('hot', 29), ('high', 29), ('tasty', 29), ('whole', 27), ('dry', 27)]
[('good', 259), ('great', 220), ('fresh', 77), ('delicious', 73), ('little', 71), ('friendly', 70), ('nice', 68), ('o
ther', 53), ('few', 53), ('first', 48), ('amazing', 47), ('favorite', 43), ('awesome', 41), ('excellent', 36), ('smal
l', 35), ('many', 34), ('next', 34), ('much', 34), ('new', 33), ('super', 33), ('sure', 32), ('different', 31), ('bi
g', 31), ('tasty', 30), ('Good', 28), ('clean', 26), ('only', 26), ('hot', 26), ('last', 24), ('perfect', 23)]
```

Positive Reviews

Finding most common nouns and adjectives for positive and negative reviews did not give a lot of insight into the data types

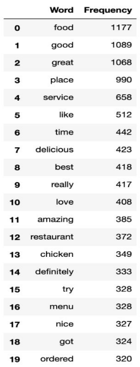
N Gram Analysis

In Natural Language Processing, N-grams as strings of words, where n stands for an amount of words that you are looking for. N-grams are contiguous sequences of n-items in a sentence

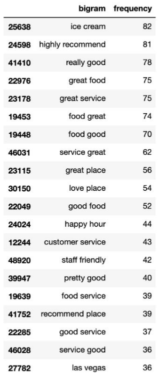
The following types of N-grams are usually distinguished:

- **Unigram**: An N-gram with simply one string inside.
- **2 Gram or Bigram**: Typically a combination of two strings or words.
- **3 Gram or Trigram**: An N-gram containing up to three elements that are processed together.

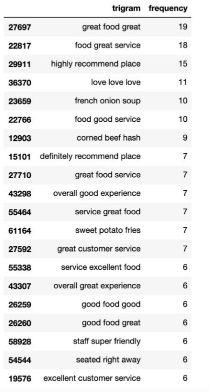
For our analysis we did n-gram analysis to have a better understanding of the type of data we are working with. We looked at positive and negative reviews separately.



	Word	Frequency
0	food	1177
1	good	1089
2	great	1068
3	place	990
4	service	658
5	like	512
6	time	442
7	delicious	423
8	best	418
9	really	417
10	love	408
11	amazing	385
12	restaurant	372
13	chicken	349
14	definitely	333
15	try	328
16	menu	328
17	nice	327
18	got	324
19	ordered	320



	bigram	frequency
25638	ice cream	82
24598	highly recommend	81
41410	really good	78
22976	great food	75
23178	great service	75
19453	food great	74
19448	food good	70
46031	service great	62
23115	great place	56
30150	love place	54
22049	good food	52
24024	happy hour	44
12244	customer service	43
48920	staff friendly	42
39947	pretty good	40
19639	food service	39
41752	recommend place	39
22285	good service	37
46028	service good	36
27782	las vegas	36

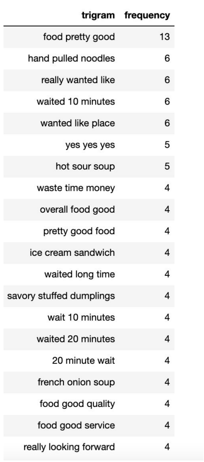


	trigram	frequency
27697	great food great	19
22817	food great service	18
29911	highly recommend place	15
36370	love love love	11
23659	french onion soup	10
22766	food good service	10
12903	corned beef hash	9
15101	definitely recommend place	7
27710	great food service	7
43298	overall good experience	7
55464	service great food	7
61164	sweet potato fries	7
27592	great customer service	7
55338	service excellent food	6
43307	overall great experience	6
26259	good food good	6
26260	good food great	6
58928	staff super friendly	6
54544	seated right away	6
19576	excellent customer service	6

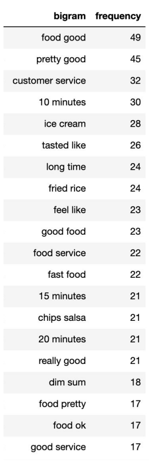
+ve Reviews

Bi-grams and trigrams gives the most context into the text data we are working with.

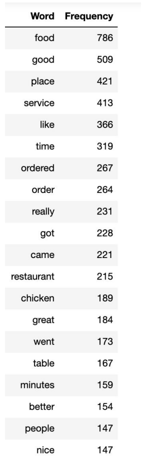
Most positive reviews seem to be around good food and service



trigram	frequency
food pretty good	13
hand pulled noodles	6
really wanted like	6
waited 10 minutes	6
wanted like place	6
yes yes yes	5
hot sour soup	5
waste time money	4
overall food good	4
pretty good food	4
ice cream sandwich	4
waited long time	4
savory stuffed dumplings	4
wait 10 minutes	4
waited 20 minutes	4
20 minute wait	4
french onion soup	4
food good quality	4
food good service	4
really looking forward	4



bigram	frequency
food good	49
pretty good	45
customer service	32
10 minutes	30
ice cream	28
tasted like	26
long time	24
fried rice	24
feel like	23
good food	23
food service	22
fast food	22
15 minutes	21
chips salsa	21
20 minutes	21
really good	21
dim sum	18
food pretty	17
food ok	17
good service	17



Word	Frequency
food	786
good	509
place	421
service	413
like	366
time	319
ordered	267
order	264
really	231
got	228
came	221
restaurant	215
chicken	189
great	184
went	173
table	167
minutes	159
better	154
people	147
nice	147

-ve Reviews

Similar to positive reviews, bi-grams and tri-grams also give more context to negative reviews.

Most customer seems to leave negative reviews because of late service.

Feature Engineering

Most of the feature engineering on the dataset was on text data. Some of the techniques that were used are:

- **Word Tokenization**: Tokenization is a way of separating a piece of text into smaller units called tokens.
- **Lemmatization**: Method of removing the suffix of the word and bringing it to a base word.
- **Bag of Words Transformation**: The bag-of-words model is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.
- **TF-Idf Transformation**: TF-IDF (term frequency-inverse document frequency) works by increasing proportionally to the number of times a word appears in a document but is offset by the number of documents that contain the word. So, words that are common in every document, such as this, what, and if, rank low even though they may appear many times, since they don't mean much to that document.

Results

After cleaning the text data, we trained the data using three different models: Naive Bayes, Random Forest and Logistic Regression. Each model was also tested using two different transformation techniques: Bag of Words Transformation and TF-Idf Transformation. The best model was also tuned using hyper parameter tuning techniques.

Naïve baes

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Naïve Bayes models are commonly used as an alternative to decision trees for classification problems. When building a Naïve Bayes classifier, every row in the training dataset that contains at least one NA will be skipped completely. If the test dataset has missing values, then those predictors are omitted in the probability calculation during prediction.

Random forest

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks. The implementation starts with picking N random records from the dataset.

- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
- In case of a classification problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

Logistic regression

Logistic regression is a classification technique where the independent variable is used to predict the dependent variable.

The Training and Prediction Processes

In the training process (a), our model learns to associate a particular input (i.e. a text) to the corresponding output (tag) based on the test samples used for training. The feature extractor transfers the text input into a feature vector. Pairs of feature vectors and tags (e.g. positive or negative) are fed into the machine learning algorithm to generate a model.

In the prediction process (b), the feature extractor is used to transform unseen text inputs into feature vectors. These feature vectors are then fed into the model, which generates predicted tags (again, positive or negative).

The precision score of the different algorithms and transformation techniques when using 1-gram are as follows. The analysis showed that Naive Bayes was the best model.

Algorithm	BOW Transformation	Tf-Idf Transformation
Naive Bayes	0.8373	0.8106
Random Forest	0.8132	0.8132
Logistic Regression	0.6861	0.6861

We tested out the model using other n-gram analysis but did not see any improvement. The precision score when using different-gram analysis on Naive Bayes model is:

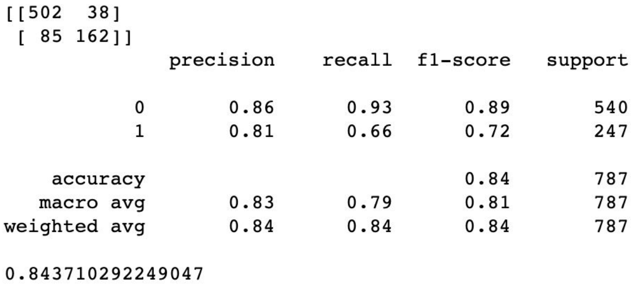
n-gram	BOW	Tf-Idf
1-gram	0.837	0.8106
2-gram	0.777	0.770
3-gram	0.699	0.682

Best Model

The best model happens when we replace contractions with long form, convert words to lowercase, remove unwanted characters, remove stop words and use Bag of Words transformation.

Using 1-gram and Naive Bayes model gives the highest accuracy score. The model also used smoothing parameters of 0.9 and a uniform prior. The precision score of the final model was 0.8437. From the analysis, we can conclude the following

- Using tf-idf transformation lowers the accuracy score.
- Using more than 1-gram lowers accuracy score.
- Using logistic regression or random forest lowers the score too.
- Increasing the training data have shown to improve model performance



```
[[502  38]
 [ 85 162]]
              precision    recall  f1-score   support

           0       0.86      0.93      0.89       540
           1       0.81      0.66      0.72       247

    accuracy                           0.84       787
   macro avg       0.83      0.79      0.81       787
weighted avg       0.84      0.84      0.84       787

0.843710292249047
```

Results from final Naive Bayes model.

Conclusion

The best model for predicting the reviews appear to be Using 1-gram and Naive Bayes model. While working on this project, preprocessing, and cleaning steps mentioned earlier are crucial to the accuracy of Sentimental Analysis. Future scope for this project will be to compare Naive Bayes Classification model against models like VADER. Also, Deep learning algorithms can be used for sentiment analysis and the results can be compared to select the most accurate models. One of the problems that this project faced was to train the model to differentiate between positive words and the positive words used in a negated context (example- not good). The project can also increase data size and explore how the model precision changes. Additionally, train the model on equal number of negative and positive reviews to understand the effect of positive skewed data on the accuracy of prediction.

References

https://www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-python

https://towardsdatascience.com/develop-a-nlp-model-in-python-deploy-it-with-flask-step-by-step-744f3bdd7776

https://towardsdatascience.com/a-complete-exploratory-data-analysis-and-visualization-for-text-data-29fb1b96fb6a

https://jonathanhsiao.com/blog/sentiment-classification-with-yelp-reviews

https://towardsdatascience.com/text-mining-and-sentiment-analysis-for-yelp-reviews-of-a-burger-chain-6d3bcfcab17b

https://medium.com/analytics-vidhya/applying-text-classification-using-logistic-regression-a-comparison-between-bow-and-tf-idf-1f1ed1b83640

https://www.kaggle.com/yelp-dataset/yelp-dataset