Yelp Reviews Prediction System

Big Data Analytics
CSCI 59000

Final Project Report

Ankita Gupta, Debraj Dey
Department of Computer Information and Science,
Indiana University Purdue University, Indianapolis, USA

Contents

1 ABSTRACT

Our aim is to develop a recommendation system which help users by recommending restaurant based on the rating of it predicted using reviews given to it by previous customers. The principal data source that we chose to work with as the project is to analyze and predictions of review given in Yelp Dataset for year 2013. Using this dataset, we extract customer, restaurant profiles and rating given for suggesting recommendations. In particular we have implemented Naïve Bayes algorithm to actualize this system. This is basically Naïve Bayes text classification. This report is an attempt to document and discuss the entire implementation details and result analysis. I start the discussion with an introduction to the problem statement, the motivation behind this project and detailed explanation of the work and results. Further, we have listed some of the possible future work and the challenges I came across while actualizing the project.

2 INTRODUCTION

A vast database of reviews, ratings and general information provided by the community about businesses, Yelp provides consumers with a myriad of options and information even when searching for an especially specific service or goods niche. However, although all required information may be present to make an informed choice, it is often still difficult by just looking at the raw data. Reading all the reviews of a single business alone is time consuming and requires more effort than the average user is willing to expand. As a result, we believe users could greatly benefit from a recommendation system.

Recommendation system have historically been created for Machine Learning applications in numerous disciplines. One such example is social networking sites such as Facebook that utilize recommendation systems to suggest friendships to users. Magic and media applications such as iTunes and Spotify also utilize similar machine learning and recommendation logic to suggest various songs, videos, movies, etc. to users based off their previous choice and taste. Given this general theme, our project focuses on creating a recommendation system for yelp users in application to potential food choices they could make.

The rise of the popular review site Yelp has led to an influx in data on people's preferences and personalities when it comes to be a modern consumer. Recommendation systems that can identify a user's preferences and identify other similar users' and/or restaurants that match his/her preferences can make this problem easier. Specifically, we aim to build a recommendation system that will enable us to make sophisticated food recommendations for yelp users by applying learning algorithms to develop a predictive model of customers' restaurant ratings.

3 Methods and Data

Data: We have used yelp data available at https://www.kaggle.com/yelp-dataset/yelp-dataset
Data files which we have utilized for the project is reviews.csv.

4 Background Work

For implementation of this project we had to think of the platform where we could deal with this huge dataset and come up with technologies. The first challenge was to design it and to decide technologies for each phase of the system. For this we have gone through many sources and deicide how we shall start working.
In this project we have tried to justify the Naïve Bayes prediction model on Text Classification i.e. the reviews user has given. Also, we have visualized the output using matplotlib library of the python.

Some background on Naïve Bayes Predictive model:

5 Implementation

We have spent good time in getting familiar with writing the scripts in python.

5.1 Preprocessing

The yelp dataset is provided on Kaggle. The data is divided into 5 sets: businesses, reviews and user's datasets. To further reduce the size of data we only worked on restaurant reviews dataset. To further reduce the size of data we only worked on restaurant reviews and neglected all other as of now. However, there was still lot of problems with the dataset. Data was too sparse; more than 90% of data was empty. This was a major issue when building recommendation system which led to other problems such as cold start. Cold start is a situation when a recommender system does not have any historical information about user or item and is unable to make personalized recommendations. Another problem that surfaced was there were large number of grey sheep unpredictable ratings. There were many users whose profile could not match other users. So, in order to resolve these issues, we spent lot of time on preprocessing of data.
Most of the important sub-tasks in pattern classification are feature extraction and selection. Prior to fitting the model and using machine learning algorithms for training, we need to think about how to best represent a text document as a feature vector. A commonly used model in Natural Language Processing is the so-called bag of words model. The idea behind this model really is as simple as it sounds. First comes the creation of the vocabulary – the collection of all different words that occur in the training set and each word is associated with a count of how it occurs.

Vectorization:

The vocabulary can be used to construct the d-dimensional feature vectors for the individual documents where the dimensionality is equal to the number of different words in the vocabulary (a=|V|d=|v|). This process id called vectorization.

Tokenization:

Tokenization describes the general process of breaking down a text into individual elements that serve as input for various NLP algorithms. Usually, tokenization is accompanied with other processing steps, such as the removal of stop words and punctuation characters, stemming or lemmatizing and the construction of n-grams.

Stop Words:

Stop words are words that are particular common in a text and thus considered as rather un-informative (e.g., words such as so, and, or, the, ….”). One approach to stop word removal is to search against a language-specific stop word dictionary. An alternative approach is to create a stop limit by sorting all words in the entire text by frequency. The stop list – after conversion into a set of non-redundant words – is then used to remove all those words from the input documents that are ranked among the top n words in this stop list.

Lemmatization:

Lemmatization aims to obtain the canonical (grammatically correct) form of the words, the so-called lemmas. It has little impact on the performance of text classification.

5.2 Naïve Bayes Classifier

Naïve Bayes classifiers are linear classifiers that are known for being simple yet very efficient. The probabilistic model of naïve Bayes classifier is based on Bayes’ Theorem and on the assumption that the features in the dataset are mutually independent. It is a two-class problem. Below is the formula used for the same.

$$\text{posterior probability} = \frac{\text{conditional probability} * \text{prior probability}}{\text{evidence}}$$

$$P(\omega_j \mid \mathbf{x}_i) = \frac{P(\mathbf{x}_i \mid \omega_j) \cdot P(\omega_j)}{P(\mathbf{x}_i)}$$

Decision Rule for Classification

In terms of good or bad review classification the decision rule of a Naïve Bayes classifier based on the posterior probabilities can be expressed as

if p(w=5|x) >= p(w=1|x) classify as good

else classify as bad.

As we discussed the posterior probability is the product of the class- conditional probability and the prior probability; the evidence term in the denominator can be dropped since for both classes.

$$p(w = 5|x) = p(x|\ w = 5).\ p(5)$$
$$p(w = 1|x) = p(x|\ w = 1).\ p(1)$$

The prior probabilities can be obtained via the maximum – likelihood estimate based on the frequencies of 5-star and 1-star reviews in the training dataset:

$$p(w = 5) = \frac{\#\text{ of 5-star rating}}{\#\text{ of ratings}}$$

$$p(w = 1) = \frac{\#\text{ of 1-star ratings}}{\#\text{ of ratings}}$$

Assuming that the words in every document are conditionally independent (as per the naïve assumption), two different models can be used to compute the class – conditional probabilities: The Multi – variate Bernoulli model and the Multinomial model.

For our project we are considering cool, useful, reviews, funny, stars feature from the yelp dataset.

Multi-variate Bernoulli Naïve Bayes

It is based on binary data. Every feature in the document is associated with the value 1 or 0; the value 1 means that the word occurs in the particular document, and 0 means that the word does not occur in it.

Multinomial Naïve Bayes

Another approach to characterize text documents – than binary values - is the term frequency (tf (t, d)). It is basically defined as the number of times a text appears in that document. The 2nd different approach is to calculate tf-idf which is called as term frequency – inverse document frequency.

We wrote python script for all the concepts explained above using python libraries such as pandas, nltk, sklearn, matplotlib, seaborn etc. After that we trained for model with 70% of data we had, and we kept rest 30% for testing purpose.

5.3 Visualization

For the purpose of visualizing the results obtained from the above model we have used python libraries itself such as matplotlib and seaborn. This gave us really good visualization and helped us to get good insights about them.

6 Results

In this section we talk about the results so obtained from the model we built. As we said all the visualizations have been done by using python libraries only.

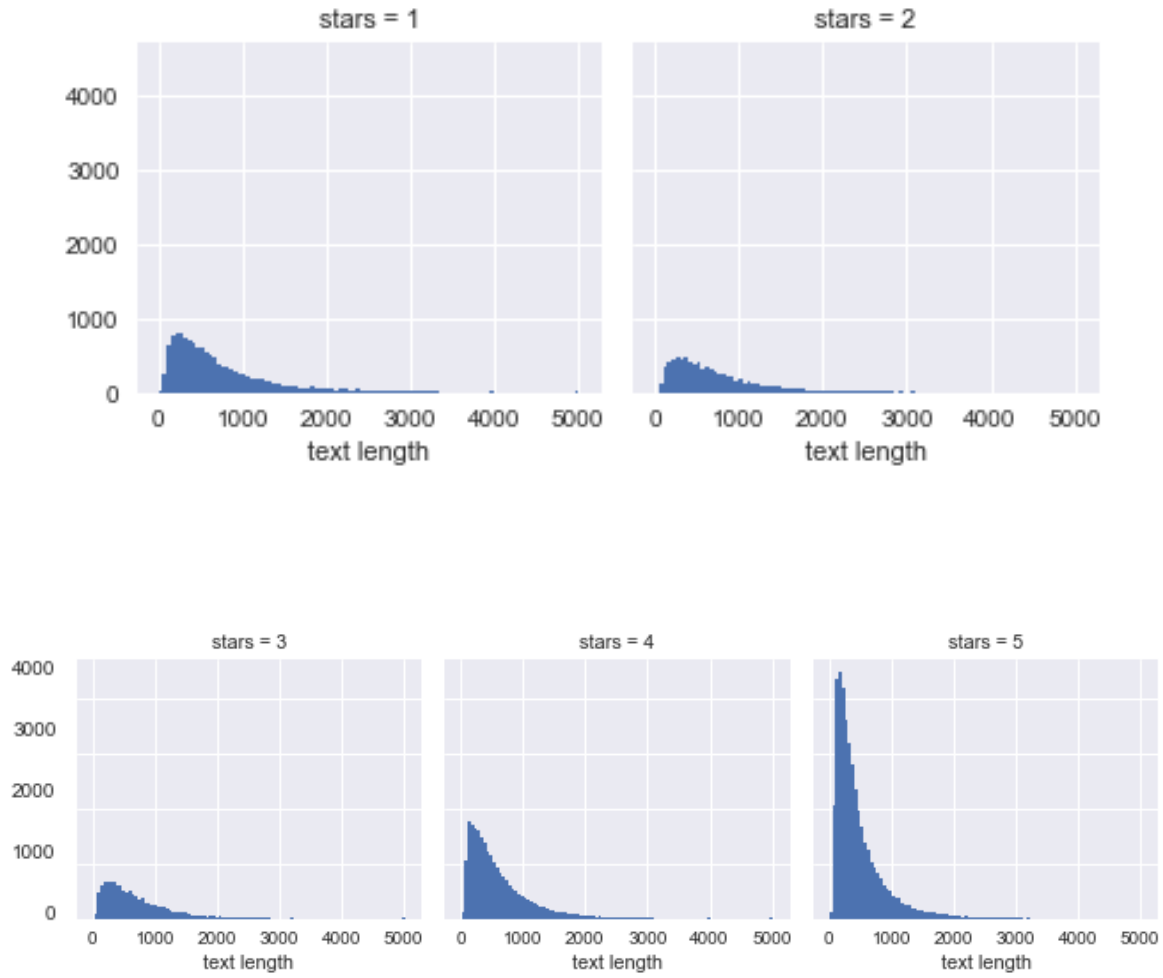

Figure 1: Length of reviews for each kind of stars

Graph in Figure 1 shows us that customer has written more length of reviews when given 5-star rating for the service. Also, the 1-star reviews are smaller in length. But, 2-star and 3-star reviews are smaller in length because less number of people given 2-star and 3-star reviews.

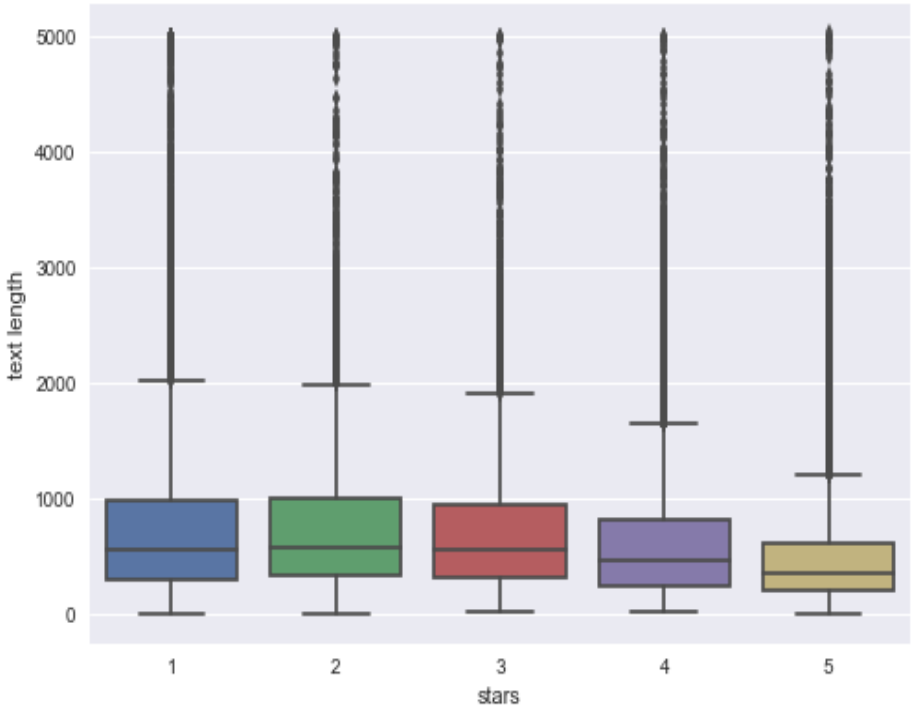

Figure 2: Boxplot of reviews length for each kind of rating

Figure 2 explains the boxplot for reviews of each stars. Also, we have given different colors to different stars.

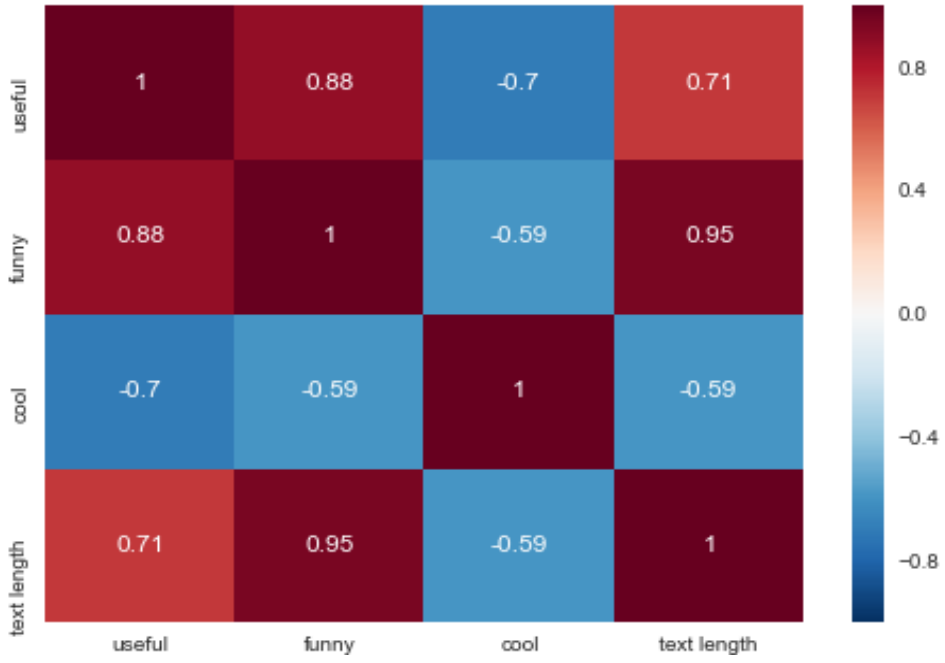

Figure 3.1: Heatmap in respect to stars

Heatmap in figure 3.1 shows that useful and funny are highly and positively correlated. This Heatmap is represented on basis of stars. Also, in figure 3.2 describes the heatmap of the matrix in respect to cool attribute.

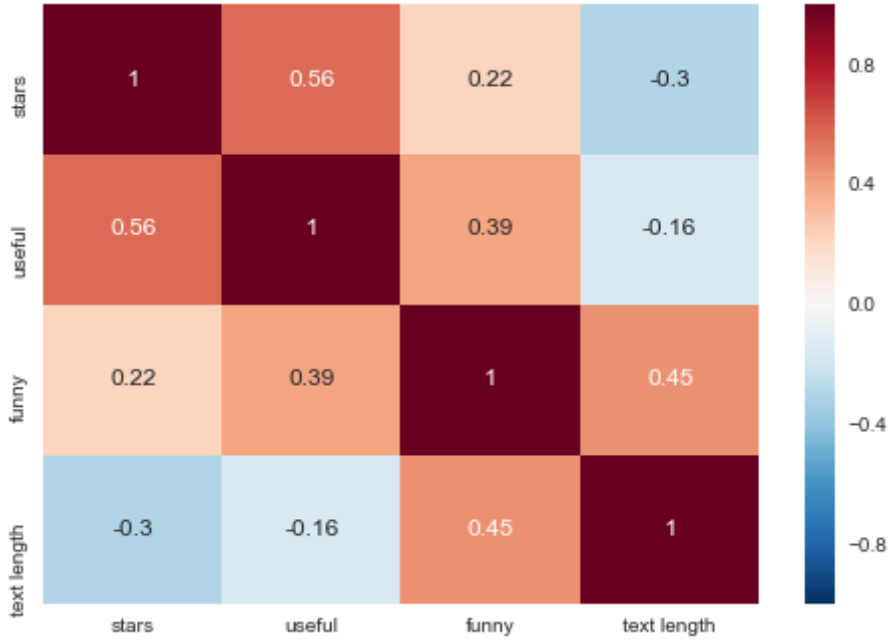

Figure 3.2: Heatmap in respect to cool

In figure 4 tells us the score of Sparse Matrix which is (55986, 103863) and non-zero occurrences which is 2565874. As the dataset is bigger, there is a lot of non-zero occurrences. Also, the accuracy of 1-star and 5-star reviews are shown in figure 5. We have reached an accuracy of 94% in our model.

```
In [29]: X = bow_transformer.transform(X)
    ...:
    ...: print('Shape of Sparse Matrix: ', X.shape)
    ...: print('Amount of Non-Zero occurrences: ', X.nnz)
Shape of Sparse Matrix:  (55986, 103863)
Amount of Non-Zero occurrences:  2565874
```

Figure 4: Example of prediction by our model

```
In [54]: print(confusion_matrix(y_test, preds))
    ...: print('\n')
    ...: print(classification_report(y_test, preds))
[[ 3474   477]
 [  550 12295]]


             precision    recall  f1-score   support

          1       0.86      0.88      0.87      3951
          5       0.96      0.96      0.96     12845

avg / total       0.94      0.94      0.94     16796
```

Figure 5: Accuracy of the model

6.1 Prediction

In our model we have only predicted positive, negative and biased reviews. In figure 6 we have shown an example of positive review and it is giving the actual review which is 5-star. In Figure 7 we are predicting the negative review and it is giving the accurate results. In our next prediction we are predicting biased reviews shown in figure 8. Here our prediction is 5-star but in the dataset we are getting a 1-star.

```
In [41]: positive_review
Out[41]: "This restaurant is incredible, and has the best pasta carbonara and the best tiramisu I've had in my life. All
the food is wonderful, though. The calamari is not fried. The bread served with dinner comes right out of the oven, and
the tomatoes are the freshest I've tasted outside of my mom's own garden. This is great attention to detail.\n\nI can no
longer eat at any other Italian restaurant without feeling slighted. This is the first place I want take out-of-town
visitors I'm looking to impress.\n\nThe owner, Jon, is helpful, friendly, and really cares about providing a positive
dining experience. He's spot on with his wine recommendations, and he organizes wine tasting events which you can find out
about by joining the mailing list or Facebook page."

In [42]: positive_review_transformed = bow_transformer.transform([positive_review])
    ...: nb.predict(positive_review_transformed)[0]
Out[42]: 5
```

Figure 6: Predicting Positive Review

```
In [39]: negative_review
Out[39]: 'Still quite poor both in service and food. maybe I made a mistake and ordered Sichuan Gong Bao ji ding for what
seemed like people from canton district. Unfortunately to get the good service U have to speak Mandarin/Cantonese. I do
speak a smattering but try not to use it as I never feel confident about the intonation. \n\nThe dish came out with
zichini and bell peppers (what!??)  Where is the peanuts the dried fried red peppers and the large pieces of scallion. On
pointing this out all I got was " Oh you like peanuts.. ok I will put some on" and she then proceeded to get some peanuts
and sprinkle it on the chicken.\n\nWell at that point I was happy that atleast the chicken pieces were present else she
would probably end up sprinkling raw chicken pieces on it like the raw peanuts she dumped on top of the food. \n\nWell
then  I spoke a few chinese words and the scowl turned into a smile and she then became a bit more friendlier. \n
\nUnfortunately I do not condone this type of behavior. It is all in poor taste...'

In [40]: negative_review_transformed = bow_transformer.transform([negative_review])
    ...: nb.predict(negative_review_transformed)[0]
Out[40]: 1
```

Figure 7: Predicting Negative Review

```
In [43]: another_negative_review
Out[43]: "Other than the really great happy hour prices, its hit or miss with this place. More often a miss. :(\n\nThe
food is less than average, the drinks NOT strong ( at least they are inexpensive) , but the service is truly hit or miss.
\n\nI'll pass."

In [44]: another_negative_transformed = bow_transformer.transform([another_negative_review])
    ...: nb.predict(another_negative_transformed)[0]
Out[44]: 5
```

Figure 8: Predicting Biased Review

7 Challenges Faced

As we mentioned earlier that we had theoretical background knowledge of Naïve Bayes Algorithm, but we wanted to have practical experience of implementing it on a real data. Some of the challenges that we faced are listed below:

- The main challenge was to choose the appropriate dataset that will help us achieve the results we wanted out of this project.
- The next challenge that we faced was to choose the appropriate features to that it contributes towards the prediction.
- Preprocessing of the data was again a challenge for us to overcome.

8 Future Work:

As we know that there is always a room for improvement. So, I would like to implement below ideas in the future if get to work on it again.

- Implement non-linear classifiers
- Compare results of both
- Predict ratings using the better one

9 Conclusion:

While working on this project for past three months, we have tried to put my best efforts to learn new big data analytics techniques and also have made attempts to improve our model by changing the features that we are using in the model. It was a good opportunity to enhance our knowledge with implementation of machine learning algorithms and improve my skill set. So, it was good learning experience for us to learn something we thought is really challenging to actualize.

10 Contributions:

Data Preprocessing was done by Ankita Gupta.
Finding the best script, Implementation of Code in pandas was done by Debraj Dey.
Contribution in project report were done by both.

11 References:

https://www.kaggle.com/yelp-dataset/yelp-dataset
http://www.nltk.org
https://seaborn.pydata.org
http://scikit-learn.org/stable/