

# Naive Bayes to predict ratings based on Yelp restaurant reviews

Nisha Selvarajan

3 October 2020

- Objectives:
- Data Description
- Using Naive Bayes to validate the review and ratings
- Interpretation of the results and prediction accuracy achieved
- Overall insights obtained from the implemented project

## Objectives:

***Does the Rating and Review match in Yelp?*** On famous websites like Amazon and Yelp, many products and businesses receive tens or hundreds of reviews, making it impossible for readers to read all of them. Generally, readers prefer to look at the star ratings only and ignore the text. However, the relationship between the text and the rating is not obvious. In particular, several questions may be asked: why exactly did this reviewer give the restaurant 3/5 stars? In addition to the quality of food, variety, size and service time, what other features of the restaurant did the user implicitly consider, and what was the relative importance given to each of them? How does this relationship change if we consider a different user's rating and text review? The process of predicting this relationship for a generic user is called Review Rating Prediction

The main challenge which we will solve is building a good predictor which effectively extract useful features of the product from the text reviews and then quantify their relative importance with respect to the rating.

## Data Description

- 33K Rows, with 17 columns. You can download data on the link <https://www.kaggle.com/shikhar42/yelps-dataset>

Names	Description
business_id	ID related to each business
name	Name of the business
address	Street Address of the business
postal_code	zip code of the business
latitude	Latitude of the business
longitude	Longitude of the business
stars	Rating given by user to the business
review_count	Total number of reviews a user had posted at the time of data collection
is_open	Restaraunt open or closed
review_id	Unique Review Id
user_id	User Id of the reviewer

## Using Naive Bayes to validate if review and ratings matches.

- ***Step 1: import dataset***
- ***Step 2: Clean the Data***
  - Create a outcome variable positive which is a true or false indicator specifying if the sentiment corresponding to the review is positive or not. All the ratings which are greater than or equal to three is considered to be a positive review.
- ***Step 3: Features and Preprocessing***
  - Load the review data into a Corpus (a collection of documents) which is the main data structure used by tm.
  - Review texts were cleaned by tm package which provides several function to clean the text via the tm\_map() function.
  - Cleaning proces include removing format, punctuation and extra whitespace. All characters from the dataset are lowercase, so there is no need to preprocess uppercase letters. Word stemming was achieved using Porter stemming algorithm, which erased word suffixes to retrieve the root or stem. Stopwords, that is, words with no information value

but appear too common in a language, were also removed according to a list from [nltk.corpus](http://nltk.corpus).

- *Step 4: Making a document-term matrix*

- The document-term matrix is used when you want to have each document represented as a row.
- Create document term matrix from the corpus
- Bag of words is a way to count terms, n-grams, across a collection of documents.
- Create dataframe from cleaned corpus

- **Step 5: Build Word Cloud**

- Word cloud is a fun way to visualize popular words/phrases in group of text.
- This function takes a single parameter of review text and builds word clouds for words occurring with the highest frequencies in reviews for these restaurants



- **Step 6: Build Word Cloud For 5 Star Reviews**

- Build word cloud for 5 star ratings.



- **Step 7: Model Training and Testing**

- I used 25% to test data and 75% to data train.
- After obtaining training and testing data sets, then we will create a separate data frame which has values to be compared with actual final values

- **Step 8: Making predictions**

- We build Naive Bayes by using training & test data sets.
- We apply Laplace smoothing , which is a technique for smoothing categorical data.
- A small-sample correction, or pseudo-count, will be incorporated in every probability estimate. Consequently, no probability will be zero. this is a way of regularizing Naive Bayes, and when the pseudo-count is zero, it is called Laplace smoothing.

## Interpretation of the results and prediction accuracy achieved

- **Evaluate the model performance using confusionMatrix**

- The accuracy of our model on the testing set is 72%.
- We can visualise the model's performance using a confusion matrix.
- We can evaluate the accuracy, precision and recall on the training and validation sets to evaluate the performance of naive bayes algorithm.

### Confusion Matrix and Statistics

```

              Reference
Prediction FALSE  TRUE
  FALSE   46645  32147
  TRUE    51353 177256

      Accuracy : 0.7284
      95% CI   : (0.7268, 0.7299)
No Information Rate : 0.6812
P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.3402

McNemar's Test P-Value : < 2.2e-16

      Sensitivity : 0.4760
      Specificity : 0.8465
      Pos Pred Value : 0.5920
      Neg Pred Value : 0.7754
      Prevalence : 0.3188
      Detection Rate : 0.1517
      Detection Prevalence : 0.2563
      Balanced Accuracy : 0.6612

      'Positive' Class : FALSE

```

- **Evaluate the model performance using CrossTable**

#### Cell Contents

-----
N
N / Col Total
-----

Total Observations in Table: 307401

	actual		
predicted	FALSE	TRUE	Row Total
-----	-----	-----	-----
FALSE	46645	32147	78792
	0.476	0.154	
-----	-----	-----	-----
TRUE	51353	177256	228609
	0.524	0.846	
-----	-----	-----	-----
Column Total	97998	209403	307401
	0.319	0.681	
-----	-----	-----	-----

## Overall insights obtained from the implemented project

- Overall accuracy of the model is 72%. It is safe to assume that naive bayes models can be trained on to find the rating of the restaurant based on the reviews.
- Sensitivity for finding ratings is 0.4760.
- Specificity for finding ratings is 0.8465.
- Since the dataset was clean, and reviews are equally distributed between test & training set, adding laplace smoothing factor did not make much difference in the accuracy.