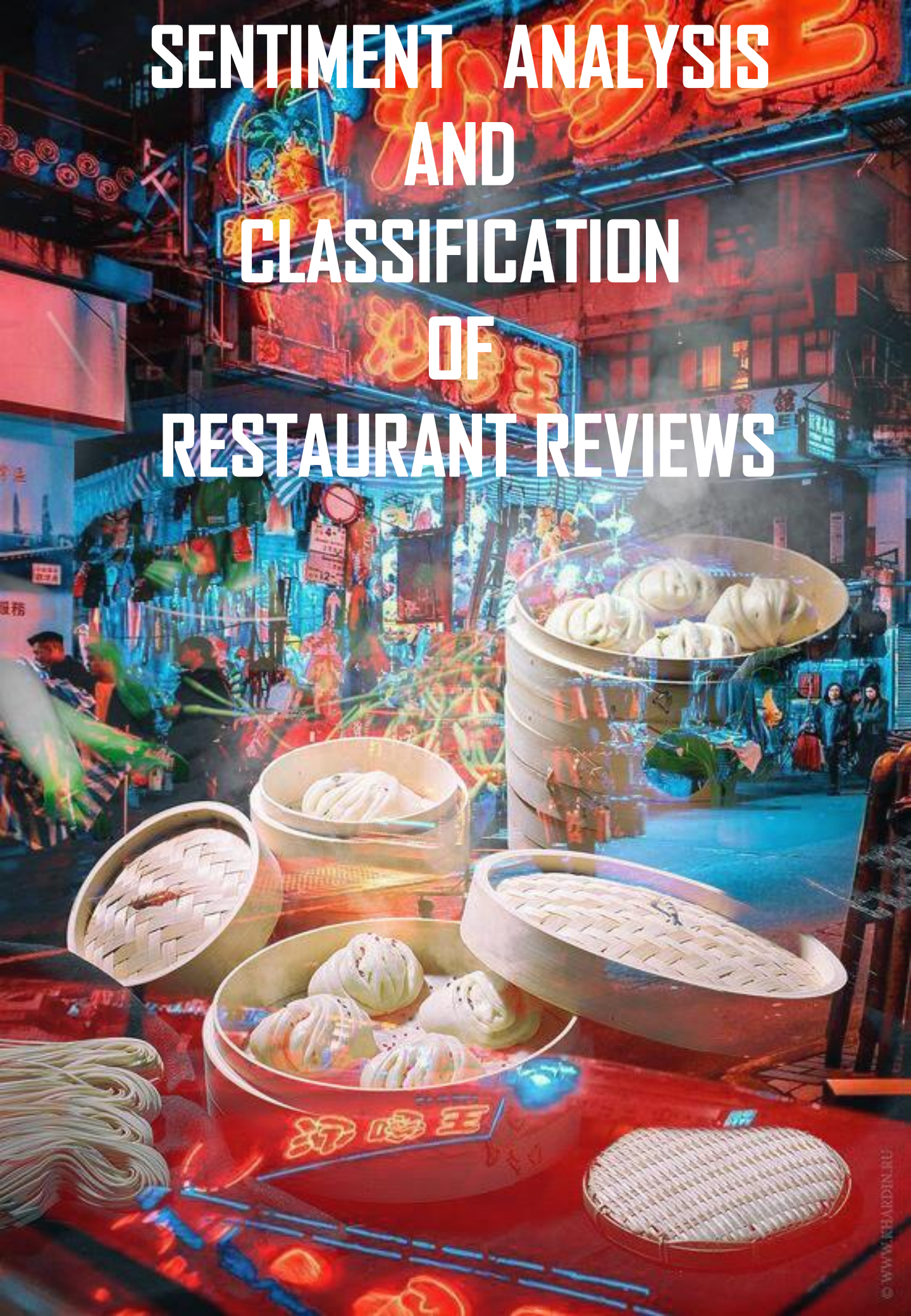
SENTIMENT ANALYSIS AND CLASSIFICATION OF RESTAURANT REVIEWS



CONTENTS

INTRODUCTION

A review is an evaluation of a publication, service, or company such as a movie, video game, musical composition, book; a piece of hardware like a car, home appliance, or computer; or an event or performance.

Online reviews are a great source of information for consumers. From the sellers' point of view, online reviews can be used to gauge the consumers' feedback on the products or services they are selling. However, since these online reviews are quite often overwhelming in terms of numbers and information, an intelligent system, capable of finding key insights (topics) from these reviews, will be of great help for both the consumers and the sellers. This system will serve two purposes:

- Enable consumers to quickly extract the key topics covered by the reviews without having to go through all of them
- Help the sellers/retailers get consumer feedback in the form of topics (extracted from the consumer reviews)

It has been found that for nearly 9 in 10 consumers, an online review is as important as a personal recommendation.

.

ABOUT THE STUDY

OBJECTIVE:
This project is on analysing the reviews of the Restaurant, Pan Asian in Chennai.

The analysis helps us understand whether majority of the customers are satisfied with the services or not. It also helps us to find the reasons why the restaurant must have been unfavourable to some. Thereby helping the company to make improvements to increase profits.
We analysed the sentiments in the reviews based on two types of lexicons and then built a classifier to classify these sentiments.

ABOUT THE DATA:
The data of 878 reviews was scraped from the tripadvisor website using R.
The scraped data was then converted to csv format with the following columns:

review: review given by the customer.

star: rating given to the product by the customer.

METHODOLOGY

SENTIMENT ANALYSIS

INTRODUCTION:

Sentiment analysis is a text analysis method that detects polarity (e.g. a *positive* or *negative* opinion) within text, whether a whole document, paragraph, sentence, or clause.
Understanding people's emotions is essential for businesses since customers are able to express their thoughts and feelings more openly than ever before. By automatically analysing customer feedback , from survey responses to social media conversations, brands are able to listen attentively to their customers, and tailor products and services to meet their needs.

THERE ARE 5 STEPS TO ANALYZE SENTIMENT DATA AND HERE'S THE GRAPHICAL REPRESENTATION OF THE METHODOLOGY TO DO THE SAME.

METHODS OF SENTIMENT ANALYSIS

Data Collection

Consumers usually express their sentiments on public forums like the blogs, discussion boards, product reviews as well as on their private logs – Social network sites like Facebook and Twitter. Opinions and feelings are expressed in different way, with different vocabulary, context of writing, usage of short forms and slang, making the data huge and disorganized. Manual analysis of sentiment data is virtually impossible. Therefore, special programming languages like 'R' are used to process and analyse the data.

Text Preparation

Text preparation is nothing but filtering the extracted data before analysis. It includes identifying and eliminating non-textual content and content that is irrelevant to the area of study from the data.

Sentiment Detection

At this stage, each sentence of the review and opinion is examined for subjectivity. Sentences with subjective expressions are retained and that which conveys objective expressions are discarded. Sentiment analysis is done at different levels using common computational techniques like Unigrams, lemmas, negation and so on.

Sentiment Classification

Sentiments can be broadly classified into two groups, positive and negative. At this stage of sentiment analysis methodology, each subjective sentence detected is classified into groups- positive, negative, good, bad, like, dislike

Presentation of Output

The main idea of sentiment analysis is to convert unstructured text into meaningful information. After the completion of analysis, the text results are displayed on graphs like pie chart, bar chart and line graphs.

RULE-BASED APPROACH:

Here's how it works:

There are two lists of words. One of them includes only the positive ones, the other includes negatives.

The algorithm goes through the text, finds the words that match the criteria.

After that, the algorithm calculates which type of words is more prevalent in the text. If there are more positive words, then the text is deemed to have a positive polarity.

MULTINOMIAL NAIVE BAYES

INTRODUCTION:

Naive Bayes is a family of algorithms based on applying Bayes theorem with a strong(naive) assumption, that every feature is independent of the others, in order to predict the category of a given sample. They are probabilistic classifiers, therefore will calculate the probability of each category using Bayes theorem, and the category with the highest probability will be output. Naive Bayes classifiers have been successfully applied to many domains, particularly Natural Language Processing(NLP).
Bayes theorem calculates probability P(c|x) where c is the class of the possible outcomes and x is the given instance which has to be classified, representing some certain features.
P(c|x) = P(x|c) * P(c) / P(x)
Naive Bayes predict the tag of a text. They calculate the probability of each tag for a given text and then output the tag with the highest one.

ALGORITHM:

First, we apply Removing Stopwords and Stemming in the text.
Removing Stopwords: These are common words that don't really add anything to the classification, such as an able, either, else, ever and so on.
Stemming: Stemming to take out the root of the word.

STEP 1: FEATURE ENGINEERING

In the first step, feature engineering, we focus on extracting features of text. We need numerical features as input for our classifier. So an intuitive choice would be **word frequencies**, i.e., counting the occurrence of every word in the document.

STEP 2: BEING NAIVE

In the non-naive Bayes way, we look at sentences in entirety, thus once the sentence does not show up in the training set, we will get a zero probability, making it difficult for further calculations. Whereas for Naive Bayes, there is an assumption that every word is independent of one another. Now, we look at individual words in a sentence, instead of the entire sentence.

STEP 3: CALCULATING THE PROBABILITIES

In the final step, we simply calculating the probabilities and compare which has a higher probability.
If probability comes out to be zero then By using Laplace smoothing: we add 1 to every count so it's never zero. To balance this, we add the number of possible words to the divisor, so the division will never be greater than 1.

BOOLEAN FEATURE MULTINOMIAL NAIVE BAYES

Here we use a variation of the multinomial Naive Bayes algorithm known as binarized (boolean feature) Naive Bayes. In this method, the term frequencies are replaced by Boolean presence/absence features. The logic behind this being that for sentiment classification, word occurrence matters more than word frequency.

ANALYSIS

EXPLORATORY ANALYSIS ON THE DATA

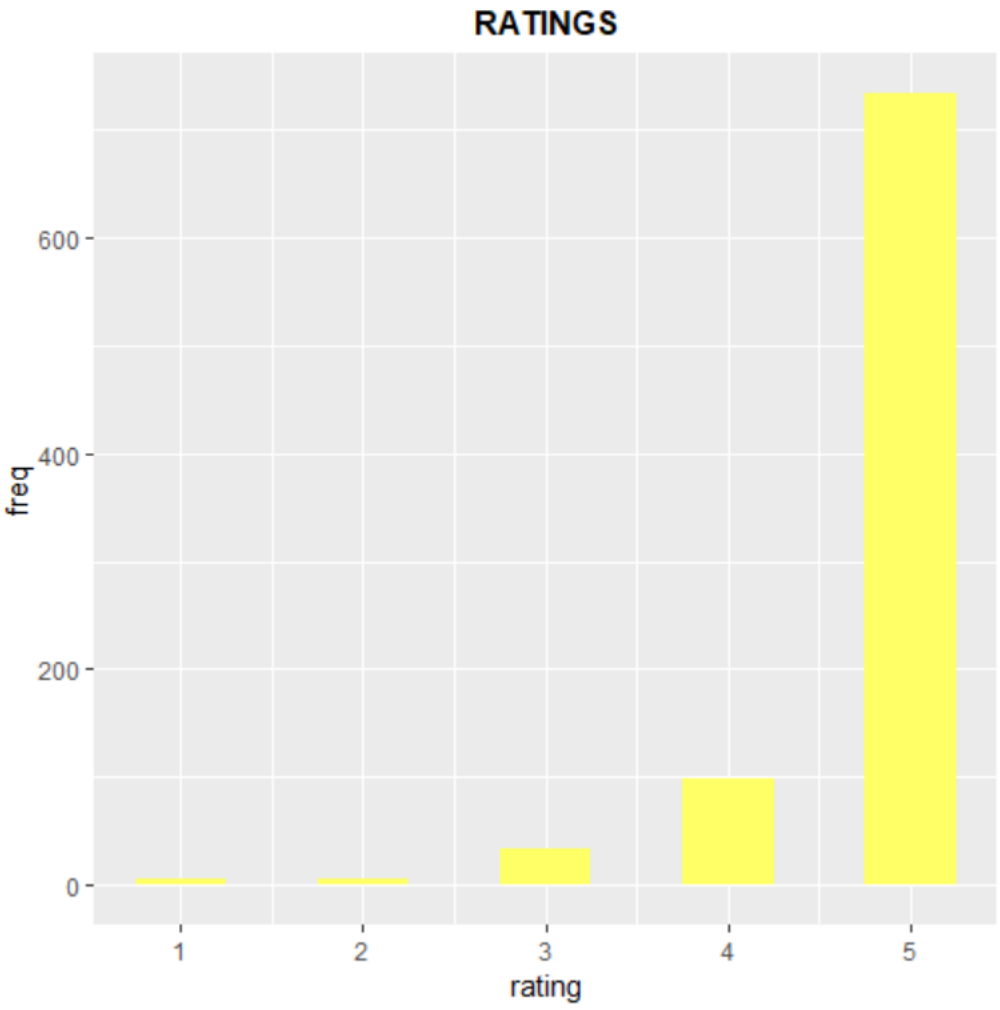


FIG 1

FIG 2

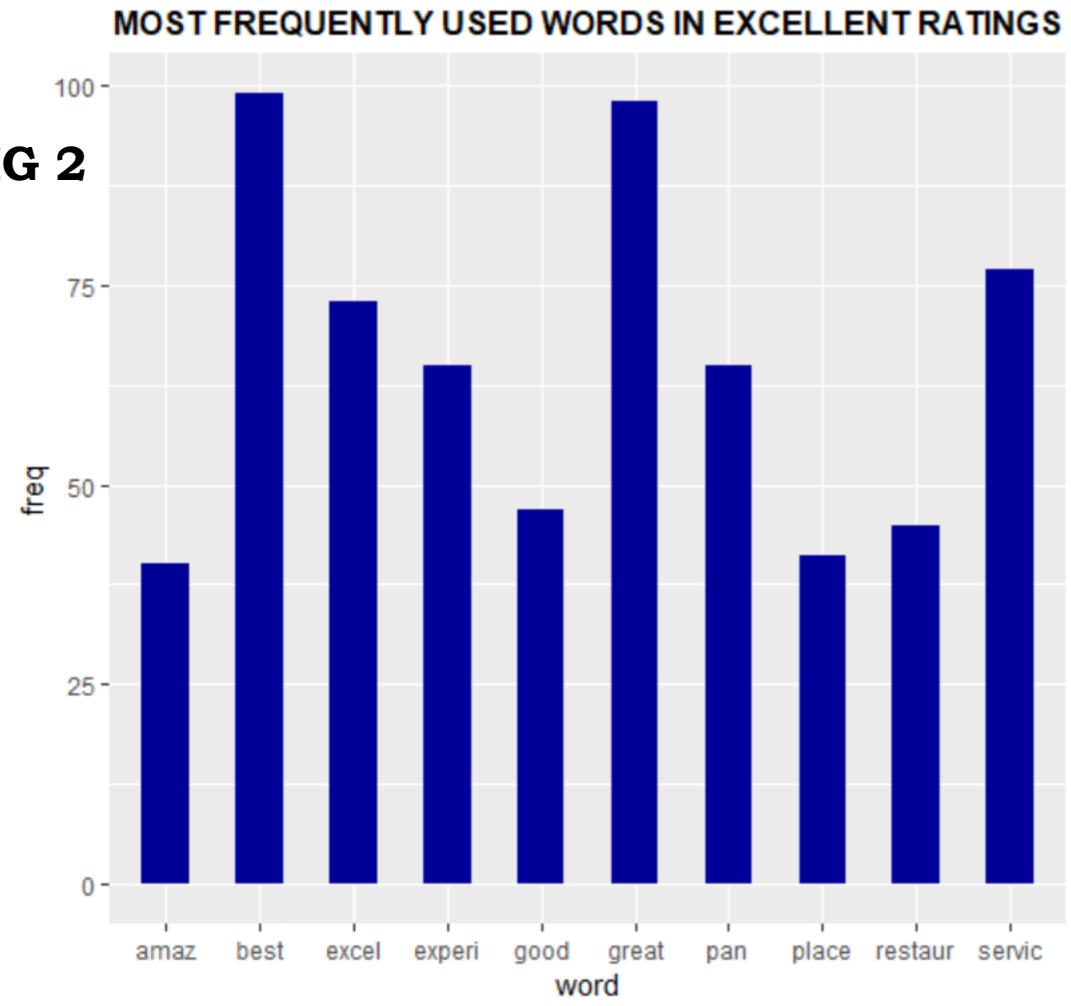
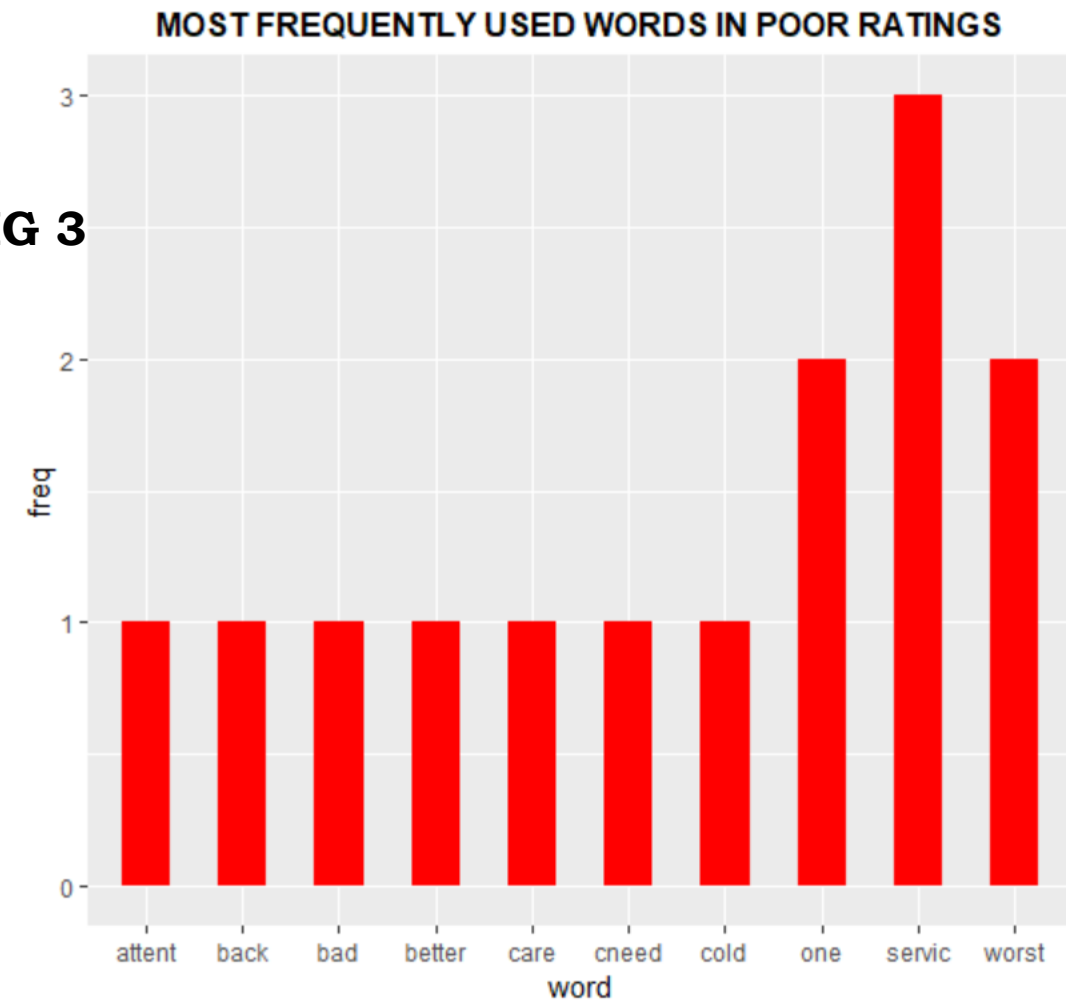

MOST FREQUENTLY USED WORDS IN EXCELLENT RATINGS
freq
100
75
50
25
0
amaz
best
excel
experi
good
great
pan
place
restaur
servic
word

FIG 3


MOST FREQUENTLY USED WORDS IN POOR RATINGS
freq
3
2
1
0
attent
back
bad
better
care
cneed
cold
one
servic
worst
word

SENTIMENT ANALYSIS ON THE DATA

LEXICON: AFINN

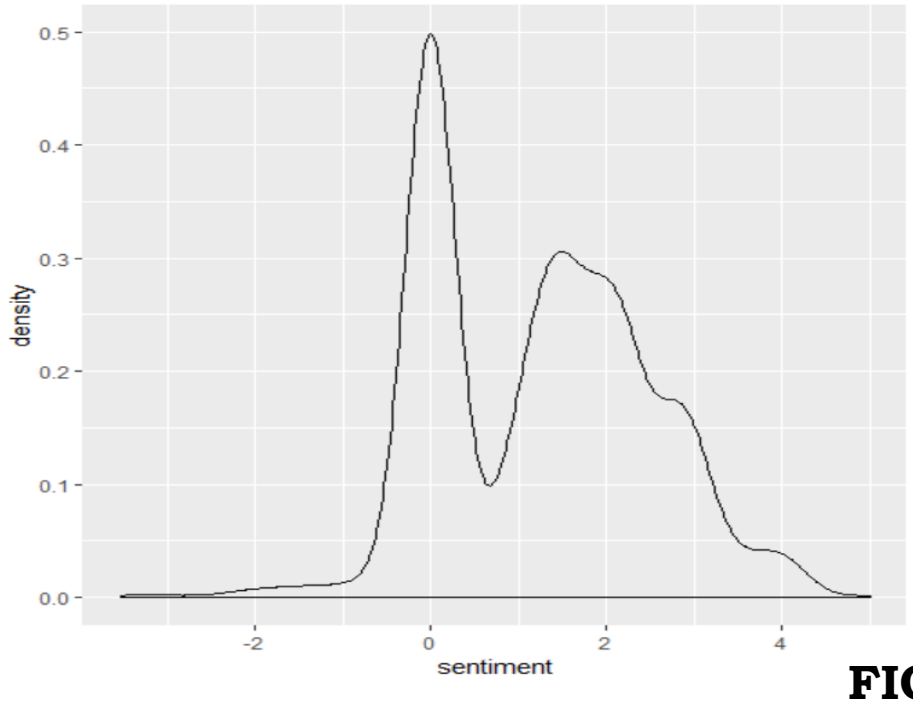


FIG 4

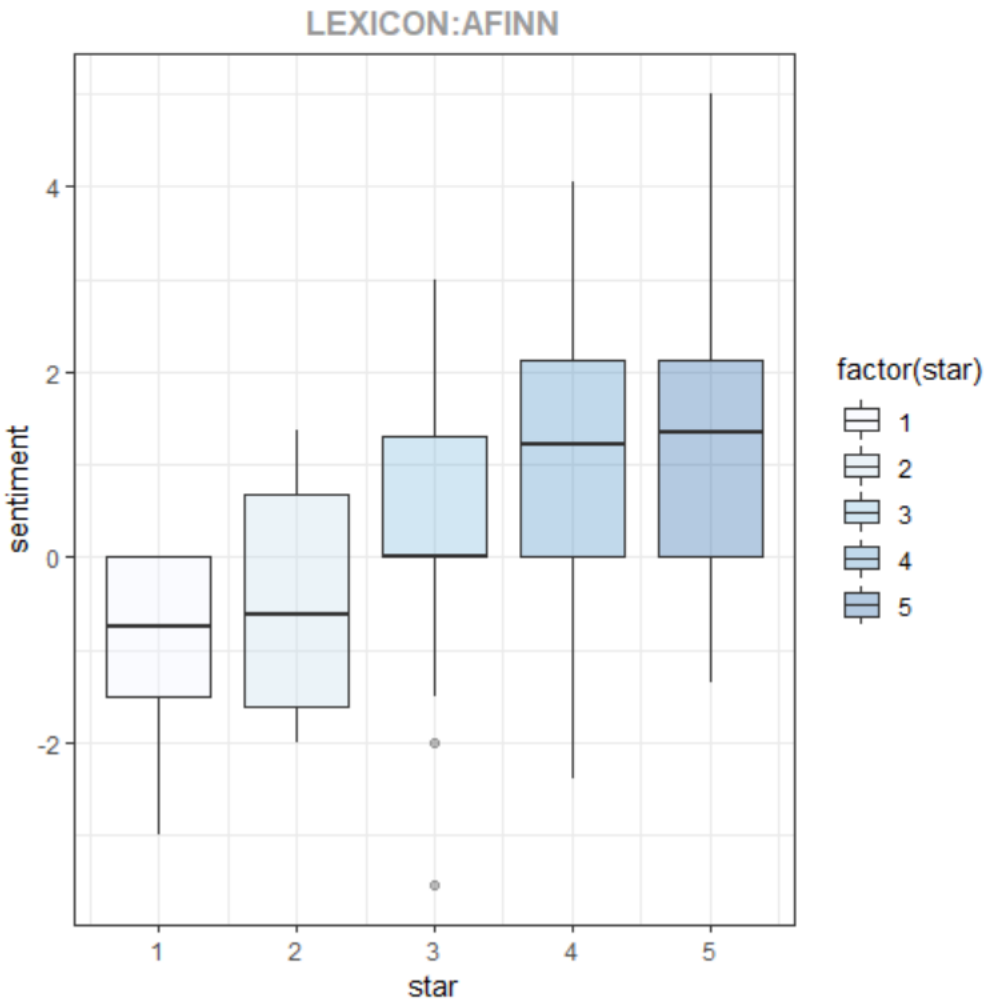


FIG 5

LEXICON: JOKER'S

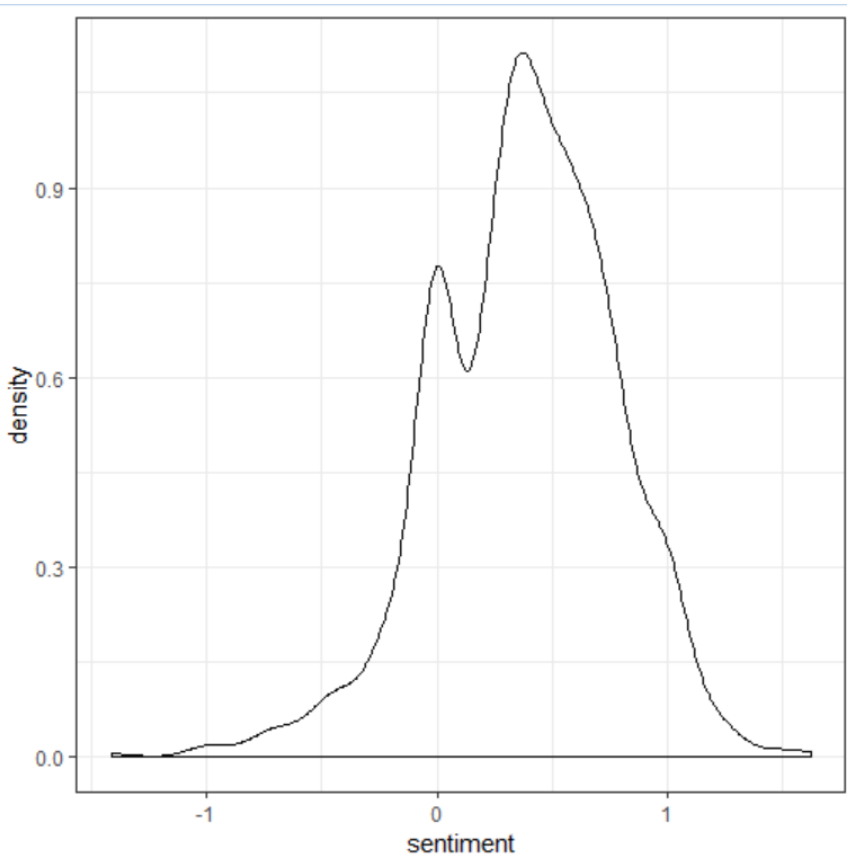


FIG 6

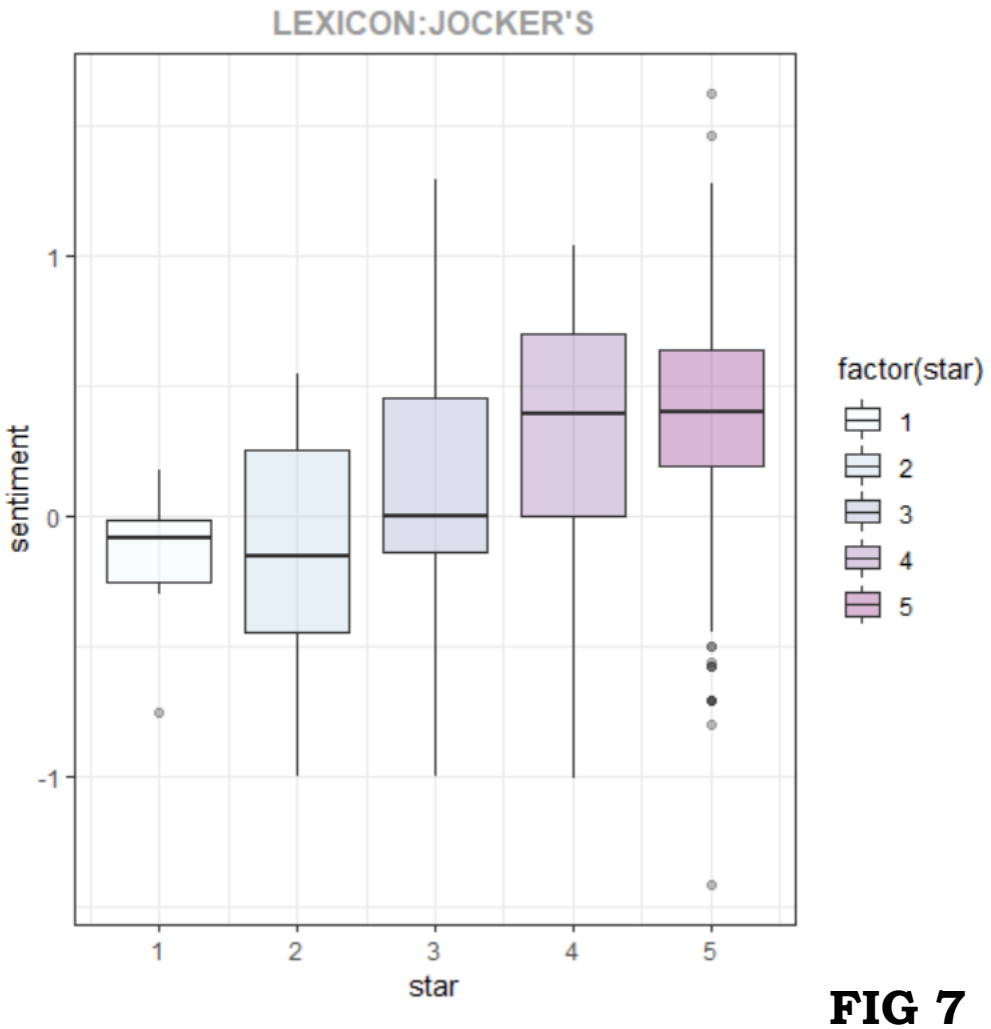


FIG 7

NAÏVE BAYE'S CLASSIFICATION

LEXICON: AFINN

PREDICTION	REFERENCE: NEGATIVE	REFERENCE: POSITIVE
NEGATIVE	110	2
POSITIVE	2	106

ACCURACY	**0.9818**
95% CI	(0.9541,0.995)
NO INFORMATION RATE	0.5091
P-VALUE	<2e-16
KAPPA	0.9636
McNEMAR'S TEST P-VALUE	1
SENSITIVITY	0.9821
SPECIFICITY	0.9815
POS PRED VALUE	0.9821
NEG PRED VALUE	0.9815
PREVALENCE	0.5091
DETECTION RATE	0.5
DETECTION PREVALENCE	0.5091
BALANCED ACCURACY	0.9818
'POSITIVE' CLASS	Negative

FIG 9

NAÏVE BAYE'S CLASSIFICATION

LEXICON: JOCKER'S

PREDICTION \ REFERENCE	NEGATIVE	POSITIVE
NEGATIVE	81	9
POSITIVE	31	99

ACCURACY	0.8182
95% CI	(0.7608,0.8668)
NO INFORMATION RATE	0.5091
P-VALUE	<2e-16
KAPPA	0.6376
McNEMAR'S TEST P-VALUE	0.0008989
SENSITIVITY	0.7232
SPECIFICITY	0.9167
POS PRED VALUE	0.9
NEG PRED VALUE	0.7615
PREVALENCE	0.5091
DETECTION RATE	0.3682
DETECTION PREVALENCE	0.4091
BALANCED ACCURACY	0.8199
'POSITIVE' CLASS	Negative

FIG 10

CONCLUSION

EXPLORATORY ANALYSIS

From the exploration of the data we can see that majority of the customers have given the restaurant five stars and the customers who have given poor ratings are very small.
The customers who have given poor ratings feel that the restaurant is expensive and probably serves cold food and probably has poor service.
From figures 1,2 and 3.

SENTIMENT ANALYSIS

Sentiment analysis was performed on sentimentr package using two type of lexicons (A dictionary or glossary relating to a particular language or sphere of interest.)

AFINN:
The AFINN lexicon assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment.

JOCKER'S:
(lexicon::hash_sentiment_jockers):
The lexicon assigns words with a score that runs between -1 and 1, with negative scores indicating negative sentiment and positive scores indicating positive sentiment.

AFINN:

The density curve in Fig 4 suggests that the distribution is bimodal because of the presence of two peaks.
With the main peak at around 0 and the lower peak in the range (1,3).

Since majority of the star ratings have been found in favour of the restaurant, reviews with sentiment score greater than or equal to zero have been assigned as “Positive” and reviews with sentiment score less than zero have been assigned “Negative”.

STAR	INTERPRETATION
1	Here we see that the absence of the upper whisker, indicates that the upper quartile is equal to the maximum value. All observations have a negative sentiment score, indicating that it is in accordance with the star rating.
2	75% of the reviews have a negative score including the median. However about 25% of the scores are positive.
3	Here we can see that the first quartile is equal to the second quartile that is the median. Also 75% of the of the reviews have a positive score whereas 25% have negative scores with two outliers.
4	Here 75% of the reviews have a positive score whereas 25% of the reviews have a negative score. The variance for this boxplot is comparatively high compared to other box plots.
5	Here 75% of the reviews have a positive score and 25% have a negative score. Variation here is comparatively less.

JOCKER'S:

The density curve in Fig 6 suggests that the distribution is bimodal because of the presence of two peaks.
With the main peak at around 0 and the lower peak around 0.5.

Since majority of the star ratings have been found in favour of the restaurant, reviews with sentiment score greater than or equal to zero have been assigned as "Positive" and reviews with sentiment score less than zero have been assigned "Negative".

STAR	INTERPRETATION
1	75% of the reviews have a negative sentiment score while 25% of the reviews have a positive score.
2	50% of the reviews have a negative score and a little less than 50% of the reviews have a positive score.
3	Most variation was observed in this boxplot with 50% of the reviews with positive score and the remaining 50% of the reviews with negative score.
4	75% of the reviews have positive score and 25% of the reviews have a negative score.
5	A little less than 25% of the reviews have negative score and 75% of the reviews have positive score.

NAÏVE BAYES CLASSIFICATION

The current state of the dataset with just two columns; review and polarity, the table of polarity showed that out of the 878 reviews 857 of them are positive and 21 of them are negative. Clearly the dataset is imbalanced.

A Naïve Bayes algorithm on an imbalanced dataset would have very high accuracy and a very high no information rate.
In this case an accuracy of 0.99 and no information rate of 0.90 indicating that 90% of the time the classifier would choose the majority class.

Hence the dataset was balanced using the ROSE (Random Over Sampling Examples)package in R which helps us to generate artificial data.
Here the minority class was oversampled with replacement and majority class was undersampled without replacement, with a probability of 0.5 for “positive” class in the newly generated sample.

Naïve Bayes algorithm was applied on the two lexicons, and the following results were interpreted from the confusion matrix obtained.

AFINN:

Accuracy (number of correct predictions divided by total number of predictions) is 98.18% with a **95% confidence interval** of 0.9541 and 0.995 meaning that there is a 95% likelihood that the true accuracy for this model lies within this range.

The **No-Information Rate** is 0.5091. This is the accuracy achievable by always predicting the majority class label. In this case if asked to predict whether a review will be positive or negative, by always choosing “Positive” we can achieve nearly 50% accuracy on the test set.

The low **p-value** indicates that we reject the Null Hypothesis :Acc=NIR and accept the Alternative Hypothesis :Acc>NIR.

The **Kappa** statistic shows how well our classifiers predictions matched the actual class labels while controlling for the accuracy of a random classifier. Kappa for this model is 0.9636 which represents excellent agreement between our classifier and the true class labels once random accuracy is controlled for.

McNemar's test is about whether the row and column marginal are equal, or, equivalently, whether the "off-diagonal" elements are equal. Specifically, the Negative/Positive and Positive/Negative cells in the confusion matrix. The test checks if there is a significant difference between the counts in these two cells.
Null Hypothesis: Classifier has a similar proportion of errors on the test set.
Alternate Hypothesis: Classifier has a different proportion of errors on the test set.
Since here the **p value** is 1, we cannot reject the null that they are equal.

From Fig 9 let
A=110, B=2, C=2, D=106 .

The confusion matrix in the figure shows that we correctly classified 110 reviews as negative and incorrectly classified 2 negative reviews as positive. We correctly classified 106 as positive reviews and incorrectly classified 2 positive reviews as negative.

Sensitivity: Also referred to as true positive rate or recall, shows the proportion of the positive class correctly predicted. Here this shows the proportion of negative reviews correctly predicted. Here it is calculated using the formula Cell A/(Cell A + Cell C) which gives 0.9821. 98% of the negative reviews were correctly predicted.

Specificity : Also referred to as true negative rate, shows the proportion of the negative class correctly predicted. Here this shows the proportion of positive reviews correctly predicted. Here it is calculated using the formula Cell D/(Cell B + Cell D) which gives 0.9815. 98% of the positive reviews were correctly predicted.

Positive Predictive Value: Also referred to as precision, shows the number of the positive class correctly predicted as a proportion of the total positive class predictions made.
Cell A/(Cell A + Cell B) which gives 0.9821.

Negative Predictive Value: Shows the number of the negative class correctly predicted as a proportion of the total negative class predictions made.
Cell C/(Cell C + Cell D) which gives 0.9815.

Prevalence: Shows how often the positive class actually occurs in our sample.
Here it is :
(Cell A + Cell C)/(Cell A + Cell B + Cell C + Cell D)
Which gives 0.5091.

Detection Rate : Shows the number of correct positive class predictions made as a proportion of all of the predictions made.
Cell A/(Cell A + Cell B + Cell C + Cell D) which gives 0.5.

Detection Prevalence : Shows the number of positive class predictions made as a proportion of all predictions.
Cell A + Cell B/(Cell A + Cell B + Cell C + Cell D) which gives 0.5091.

Balanced Accuracy : Essentially takes the average of the true positive and true negative rates i.e (sensitivity + specificity)/2.
Here it is 0.9818.

Positive: an optional character string for the factor level that corresponds to a "positive" result (if that makes sense for your data). If there are only two factor levels, the first level will be used as the "positive" result.
Here it is Negative ie Negative Reviews.

JOCKER'S:

Accuracy (number of correct predictions divided by total number of predictions) is 81.82% with a **95% confidence interval** of 0.7608 and 0.866 meaning that there is a 95% likelihood that the true accuracy for this model lies within this range.

The **No-Information Rate** is 0.5091. In this case if asked to predict whether a review will be positive or negative, by always choosing "Positive" we can achieve nearly 50% accuracy on the test set.

The low **p-value** indicates that we reject the Null Hypothesis :Acc=NIR and accept the Alternative Hypothesis :Acc>NIR.

The **Kappa** statistic for this model is 0.6 which represents good agreement between our classifier and the true class labels once random accuracy is controlled for.

McNemar's test p-value here the **p value** is 1, we cannot reject the null that they are equal.

From Fig 10 let
A=81, B=9, C=31, D=99 .

The confusion matrix in the figure shows that we correctly classified 81 reviews as negative and incorrectly classified 9 negative reviews as positive. We correctly classified 99 as positive reviews and incorrectly classified 31 positive reviews as negative.

Sensitivity: Here it is 0.7232. 72% of the negative reviews were correctly predicted.

Specificity : Here it is 0.9167. 91.6% of the positive reviews were correctly predicted.

Positive Predictive Value: Also referred to as precision, shows the number of the positive class correctly predicted as a proportion of the total positive class predictions made.
Cell A/(Cell A + Cell B) which gives 0.9.

Negative Predictive Value: Shows the number of the negative class correctly predicted as a proportion of the total negative class predictions made.
Cell C/(Cell C + Cell D) which gives 0.7615.

Prevalence: Shows how often the positive class actually occurs in our sample.
Here it is :
(Cell A + Cell C)/(Cell A + Cell B + Cell C + Cell D)
Which gives 0.5091.

Detection Rate : Shows the number of correct positive class predictions made as a proportion of all of the predictions made.
Cell A/(Cell A + Cell B + Cell C + Cell D) which gives 0.3682.

Detection Prevalence : Shows the number of positive class predictions made as a proportion of all predictions.
Cell A + Cell B/(Cell A + Cell B + Cell C + Cell D) which gives 0.4091.

Balanced Accuracy : Essentially takes the average of the true positive and true negative rates i.e (sensitivity + specificity)/2.
Here it is 0.8199.

Positive: an optional character string for the factor level that corresponds to a "positive" result (if that makes sense for your data). If there are only two factor levels, the first level will be used as the "positive" result.
Here it is Negative ie Negative Reviews.

REFERENCES

https://monkeylearn.com/sentiment-analysis/
https://www.edureka.co/blog/sentiment-analysis-methodology/
https://theappsolutions.com/blog/development/sentiment-analysis/
https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf
https://www.geeksforgeeks.org/applying-multinomial-naive-bayes-to-nlp-problems/
https://degreesofbelief.roryquinn.com/common-evaluation-measures-for-classification-models
https://stats.stackexchange.com/questions/129498/importance-of-mcnemar-test-in-caretconfusionmatrix
https://stats.stackexchange.com/questions/347292/mcnemars-test-p-value-output-in-r-confusion-matrix