Sentiment Analysis of Customer Reviews in Zomato Bangalore Restaurants

Aarushi Agarwal
CSE Dept.
PES University
Electronic City Campus
Bangalore, India
aarushi.agarwal11@gmail.com

Arjun Harish
CSE Dept.
PES University
Electronic City Campus
Bangalore, India
itsmearjun2001@gmail.com

Anwesha Kelkar
CSE Dept.
PES University
Electronic City Campus
Bangalore, India
anweshakelkar@gmail.com

Mohnish Srikanth
CSE Dept.
PES University
Electronic City Campus
Bangalore, India
mohnishsrikanth@gmail.com

Abstract: With the popularity of social media and e-commerce websites, sentiment analysis has become an active area of research. Sentiment analysis tries to understand the public opinion about a specific product, topic or trends from reviews or tweets. It plays an important role in understanding customer opinions and extracting social/political trends.
In this work, we perform sentiment analysis on customer reviews found on the zomato website. For this we make use of a long short-term-memory (LSTM) model to classify the review as positive, negative or neutral.

Keywords:
Natural Language processing (NLP), Sentiment Analysis, Tokenization, Customer reviews, Long short-term-memory (LSTM) model

INTRODUCTION

Reviews on Zomato are still in the form of text and can be classified with positive, negative, or neutral ratings. Zomato does not have an analysis of how users interact with the reviews and which words indicate that the customers enjoyed their restaurant experience or not. We need to extract these words from the reviews and analyse them, so we know how users interact in the Zomato review section.
Natural Language Processing (NLP) is a subfield of Artificial Intelligence that deals with understanding and deriving insights from human languages such as text and speech. Sentiment Analysis is an application of Natural Language Processing. Sentiment Analysis is a machine learning tool that analyses texts for polarity. By training machine learning tools with examples of emotions in text, machines automatically learn how to detect sentiment without human input.
Sentimental Analysis is performed by various businesses to understand their customer behaviour towards the products well. It gives them automatic feedback from the customer that helps them to take actions accordingly.

Recurrent Neural Networks (RNNs) are one of the most prevalent architectures because of the ability to handle variable-length texts. Though RNNs are capable of modelling long sequential data theoretically they fail to represent long sequences in real time applications. Long short-term-memory or LSTM model is an updated version of Recurrent Neural Network to

overcome the vanishing gradient problem.

Dataset

A. About the dataset

Bengaluru has more than 50,000 restaurants serving dishes from all over the world. The basic idea of the Zomato dataset is to summarise all the details of these restaurants such as name, address, phone number, website URL etc in a tabular form. It also contains customer feedback in the form of ratings, most liked dish, customer reviews etc. The data is in a csv format and is approximately 547MB in size.

B. Review Data

The dataset contains 17 variables all of which were scraped from the Zomato website. The dataset contains details of 51,717 restaurants in Bengaluru.

C. Link to the dataset: Zomato Bangalore Restaurants

LITERATURE REVIEW

- *Sentiment Analysis of Customer Reviews in Zomato Bangalore Restaurants Using Random Forest Classifier*

Bern Jonathan, Jay Idoan Sihotang, Stanley Martin, Department of Technology, Female Daily Network, Department of Information Technology, Universitas Advent Indonesia

The paper conducts analysis on sentiment of customer reviews in Zomato Restaurants located in Bangalore. A Random Forest Classifier is used to classify the sentiments of users based on their review. In addition to this, the specific words that affect the model classifier are also found. We find this paper very insightful for the fact that they have used Random Forest to create as many trees as possible on the subset of the data and combine the output of all the trees. This way, it reduces overfitting problems in decision trees and reduces the variance and therefore improves the accuracy. We can use the word data of sentiments to find the sentiments rather than see their ratings. The issue with this research is that the least percentage of recall is neutral. It is indicating the machine model is less neutral than the others.

- *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova 2018

The paper gives an idea of a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabelled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. This paper has been insightful on the simplicity and empirical power of BERT while trading off computational resources. The main drawbacks of using BERT and other big neural language models is the computational resources needed to train/fine-tune and make inferences and possible bias added.

- *Analysing Scientific Papers Based on Sentiment Analysis*

Doaa Mohey El-Din Mohamed Hussein

This paper has provided a great understanding on SAOOP which is a natural language processing, text analysis and opinion mining in the sentiment analysis evaluation. We also learnt that the main disadvantages of this classifier are the large size of the dictionary, and its size should not exceed a certain value at which it will be resistant to noise. Also, the bag of words does not consider spatial information about the object in any way, which, if there are similar points in descriptors of different points of different objects in the image, their descriptions may coincide. The main disadvantage of this classifier is the large size of the dictionary, and

its size should not exceed a certain value at which it will be resistant to noise.

- *Location based restaurant preferences in Bangalore*

Dr. Anupam Bhatia, Ms. Sneha

In this work, the impact of restaurant type and location on the restaurant's success is investigated. A restaurant is chosen and is empirically investigated to analyse how the success of this restaurant can be affected by its location and address. The main problem with this method is that the complex joins may take time to execute due to shuffling of data, it is slower than Apache Spark by almost 100 times and it is dependent on external memory and storage to execute.

- *The effects of pictures, review credibility and personalization on users' satisfaction of using restaurant recommender apps: Case study: Zomato*

From this study we understand the effects of pictures, review credibility and personalization features on users' satisfaction in accessing restaurant recommender apps. They have used a quantitative approach with a total of 419 collected respondents. The data gathered within this study is analysed using multiple regressions. The results of this study indicate that pictures and review credibility are affecting user satisfaction. It is seen that providing personalization on a restaurant recommender apps may also improve its user satisfaction in accessing the apps. The main problems with this method is that it doesn't consider the meaning behind social phenomena, every answer provided in this research method must stand on its own, quantitative research sometimes creates unnatural environments and there is no access to specific feedback.

PROPOSED SOLUTION:

To solve the challenge of gaining insights from customer reviews on the Zomato website, our team performed a sentiment analysis using LSTM model.
This process can be broken down into the following stages:

i) pre-processing: data pre-processing, visualisations & text pre-processing

ii) building a model

iii) evaluation

DATA PRE-PROCESSING:
Data attributes and corresponding data types before exploratory data analysis:

```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   name          51717 non-null  object
 1   online_order  51717 non-null  object
 2   book_table    51717 non-null  object
 3   rate          43942 non-null  object
 4   votes         51717 non-null  int64
 5   location      51696 non-null  object
 6   rest_type     51490 non-null  object
 7   cuisines      51672 non-null  object
 8   approx_cost   51371 non-null  object
 9   reviews_list  51717 non-null  object
 10  type          51717 non-null  object
 11  city          51717 non-null  object
```

Data Cleaning:
Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

- We check for duplicate values using the duplicated () function and drop them using the drop_duplicates() function.

- We remove NaN values from the dataset using dropna() function.

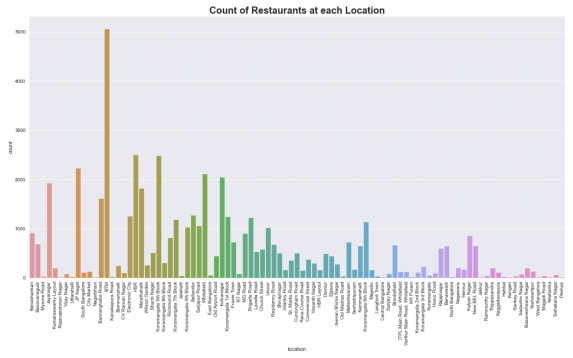
- We predict the importance of columns and drop/keep them accordingly, mainly retaining only text data. We remove certain columns that are not required for our study such as: 'url', 'address', 'phone', 'dish_liked', 'menu_item'.

- We also change the names of certain columns to more meaningful ones: approx_cost(for two people)' to 'approx_cost', 'listed_in(type)' to 'type' and 'listed_in(city)' to 'city'.

- Changing data types of ‘cost’ and ‘rate’ attributes to float.

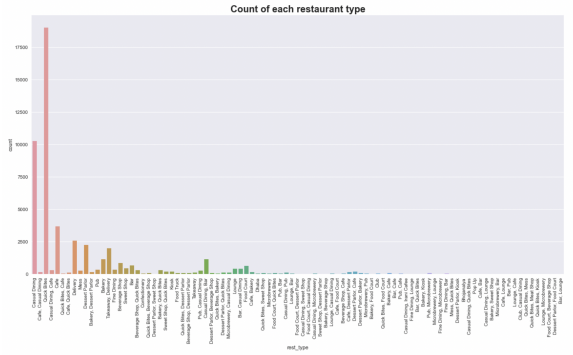
VISUALIZATIONS:

1. Count plot of location:



Observation: There are more than 5000 restaurants in BTM. Quick Google Search shows us that BTM is a posh residential area so because of that there are quite a lot of restaurants, and it is famous for cafes. JP Nagar HSR, Koramangala 5th block, Whitefield, Indiranagar have more than 2000 restaurants and Jayanagar, Marathahalli, Bannerghatta Road have more than 1000 restaurants.

2. Bar plot of rest_type:

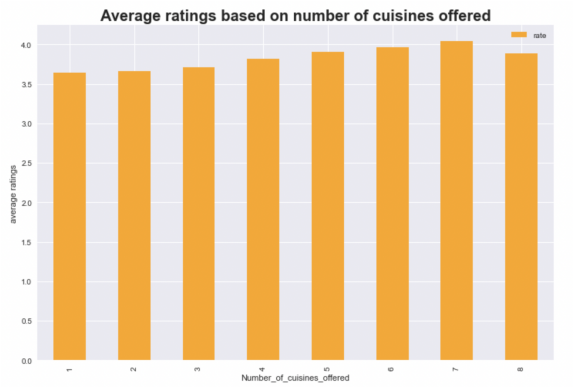


Observation: Quick Bites is the most popular restaurant type as its count is more than 19000. It is followed by Casual Dining which has more than 10,000 records. Cafe, Delivery, Dessert Parlour are some other common restaurant types.

3. Grouping by number of cuisines offered and finding the mean of approx_cost for each cuisine:

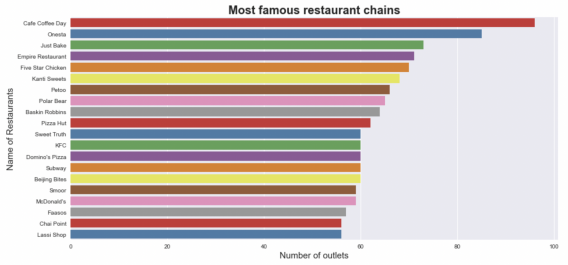
Observation: The restaurants which offer 7 cuisines have the highest approx_cost above Rs 800. It is followed by restaurants offering 6 cuisines with approx_cost more than Rs 800 and restaurants offering 5 cuisines with approx_cost close to Rs 800. Here, the approx_cost where the number of cuisines offered is 8 is less as there might be fewer restaurants offering these many cuisines. We can conclude that restaurants which offer more cuisines have a higher approx_cost value compared to restaurants which offer less than 5 cuisines.

4. Grouping by number of cuisines offered and finding the mean of approx_cost for each cuisine



Observations: Restaurants offering 7 cuisines do have higher ratings of around 4.0 but there is no significant change in the ratings given to others based on the number of cuisines offered.

5. 20 Most famous restaurant chains:

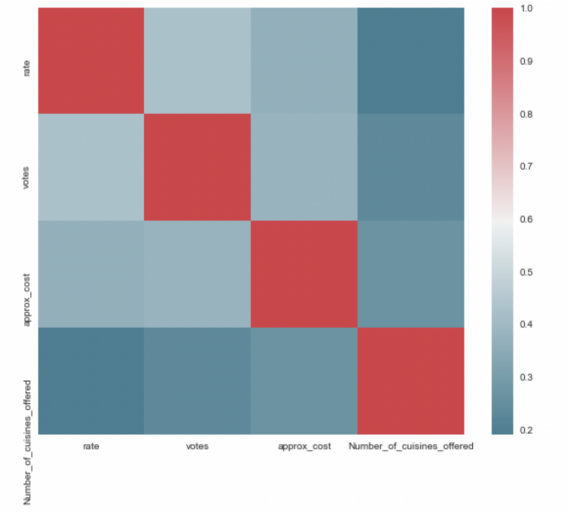


Observation: Cafe Coffee Day has maximum outlets in the city and is followed by Onesta with several outlets little more than 80. It can be seen that the most famous restaurant chains have more than 50 outlets.

6. Correlation: Finding the correlation between all numeric attributes in the Data frame

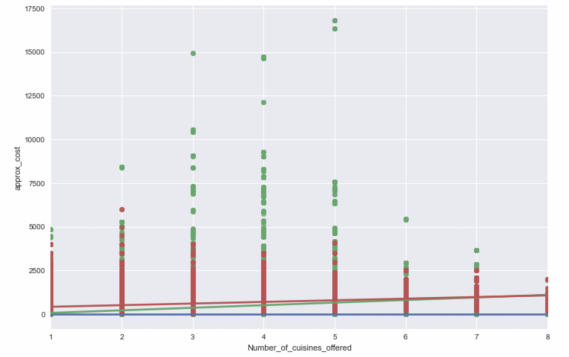
	rate	votes	approx_cost	Number_of_cuisines_offered
rate	1.000000	0.428843	0.366382	0.190368
votes	0.428843	1.000000	0.381648	0.232068
approx_cost	0.366382	0.381648	1.000000	0.268574
Number_of_cuisines_offered	0.190368	0.232068	0.268574	1.000000

Heatmap for the pairwise correlation of all the columns in the data frame.



Correlation plots showing the correlation of Rate, Votes and Approx. Cost respectively against number of cuisines offered

sns.regplot() is used to plot data and a linear regression model fit



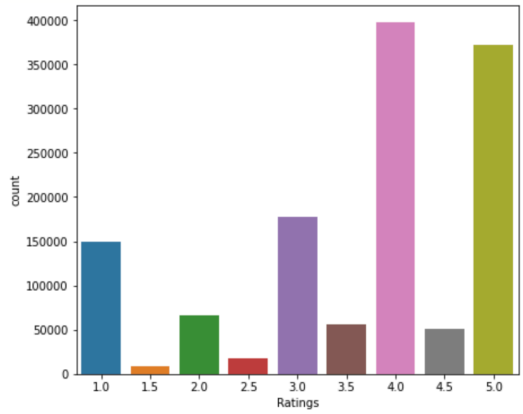
PRE-PROCESSING TEXTS:

Various types of noise are present in the text and the data is not readily usable without any pre-processing. The entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text pre-processing and is mandatory to get any coherent results.

- We replace any occurrence of '/5' with ' ' in the 'Rate' column using the lambda function and replace().
- We change any uppercase letters to lowercase letters found in the 'review' column using the lower()
- We remove any punctuation present in the text. For this, we define a function remove_punctuation(text) which replaces any punctuation present with a blank space.
- Stopwords are English words which do not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. We eliminate such words by importing stopwords from nltk.corpus and defining a function remove_stopwords(text).
- Removing any URLs that are present in the 'review' column.

- We then examine the ‘score’ column and encode the categorical data into binary labels using get_dummies. It converts categorical data into dummy or indicator variables.

Distribution of ratings:



BUILDING A MODEL:

At this point, we are ready to build our model. For this, we will go through the following steps:

1. Fit the Keras Tokenizer
2. Build an embedding matrix
3. Tokenize and pad our training data
4. Train the model

We first create the tokenizer. To do this, we first make a list of all symbols and pass it as an argument to the text.Tokenizer(). Then we fit the keras tokenizer to our data frame using the tokenizer.fit_on_texts().

Next, we create the embedded matrix where words are represented by dense vectors where a vector represents the projection of the word into a continuous vector space. The position of a word in the learned vector space is referred to as its embedding.

We then tokenize and pad the dataframe. texts_to_sequences() is used to transform each text in texts to a sequence of integers. Only words known by the tokenizer will be taken into account. pad_sequences() is used to ensure that all sequences in a list have the same length. By default, this is done by padding 0 in the beginning of each sequence until each sequence has the same length as the longest sequence.

Finally, we train the model. We used keras sequential function that allows to input and output sequences of data. For the activation function we made use of “softmax”. We also made use of word embeddings that represent words in an array and as continuous vectors. To train the model we use the fit() method.

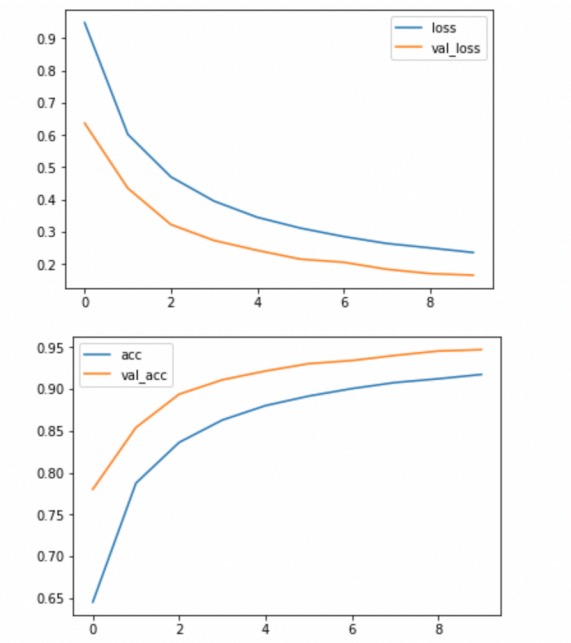
Summary of the model:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 112, 32)           96000

 lstm (LSTM)                 (None, 32)                8320

 dense (Dense)               (None, 2)                 66

=================================================================
Total params: 104,386
Trainable params: 104,386
Non-trainable params: 0
_________________________________________________________________
None
```

EVALUATION:

Keras can separate a portion of the training data into a validation dataset and evaluate the performance of your model on that validation dataset each epoch.

To evaluate the model, we used model.evaluate() method and passed the testing dataset as parameters. The output of this function gave us score and accuracy.



The above graph shows the accuracy and loss of the training model

EXPERIMENTAL RESULTS

The results obtained from the above model are:

```
score: 0.12
acc: 0.96
```

For this project we choose accuracy as our main evaluation criteria. The model shows overall accuracy as 0.96 and the score as 0.12.

CONCLUSIONS:

1. Contributions of each team member:
   - Literature Review Report (Phase 1):

     Each team member contributed equally towards this aspect of the project by reviewing three papers each.
     Total : 12 research papers

   - Data Pre-processing: Arjun Harish and Aarushi Agarwal
   - Data Visualisation: Anwesha Kelkar and Mohnish Srikanth
   - Pre-processing Text: Mohnish Srikanth and Aarushi Agarwal
   - Building a model: Anwesha Kelkar and Arjun Harish
   - Final evaluation and conclusion of project: Anwesha Kelkar, Mohnish Srikanth, Arjun Harish and Aarushi Agarwal.
   - Final Report (Phase 2): Each team member contributed equally towards this aspect of the project.

2. References:

   - https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentiment-analysis-af410fd85b47
   - https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/
   - https://scholar.google.com/
   - https://www.kaggle.com/datasets