

# Restaurant Recommendation Systems

1<sup>st</sup> Basavaraj Harvi

*Dept. Of Computer Science and Engg.*

*PES University*

Bengaluru, India

basavaraj456vc2@gmail.com

2<sup>nd</sup> Bhavikha Chopra

*Dept. Of Computer Science and Engg.*

*PES University*

Bengaluru, India

bhavikachopra1821@gmail.com

3<sup>rd</sup> Gitika Jain

*Dept. Of Computer Science and Engg.*

*PES University*

Bengaluru, India

gitikavinaykiya@gmail.com

4<sup>th</sup> Numan Mohammed Abbas

*Dept. Of Computer Science and Engg.*

*PES University*

Bengaluru, India

mohd123456789numan@gmail.com

**Abstract**—In modern day it causes a lot of stress for a person to decide on what to eat and where to eat since the number of restaurants and the variety of foods served by them is enormous. One wants to eat good quality food, which is served hot at a reasonable price. This project aims to analyse the user's past history and get insightful information about user's behaviour. A recommender model is proposed where user's food ordering behaviour is recorded for that given time of the day and analysed. Based on the analysis user is recommended restaurants based on his past orders and taste.

**Index Terms**—Zomato Recommender, Cosine Similarity, TF-IDF, Sentiment Analysis, LSTM, Content based filtering, Knowledge based filtering.

## I. INTRODUCTION

Bengaluru is one of the best places with a variety of food available from all over the world. Currently, there are about 12,000+ restaurants. With such a high number of restaurants. This field is not yet complete. And new restaurants are being opened every day. However, it has been difficult for them to compete with existing restaurants. This paper will help newly opened restaurants in discovering menus, cuisine, cost, etc. for a particular location.

The number of online reviews for products and services has grown considerably that often making it infeasible to read all of them. The method proposed by this paper considers the user comments and suggests restaurants based on how alike their menus are to the preferences. It also considers that based on the dishes that the user has chosen previously, the amount he/she spent, the system provides him personalized recommendations with a similar menu.

## II. BACKGROUND

Today we live in a world where a new restaurant/cafe pops up every day. This gives the public a lot of options to choose from. There's something for every taste bud. But one downfall to this is that having too many options can confuse the masses and they may end up going to the same place every time. There's always understandable hesitation to try these places without dependable reviews. Although the web world is a source of much handy information, it does introduce many concerns and can make the decision-making process tiresome and complicated. Therefore, the information must be filtered and personalized about a particular user. Some of these restaurants also have many outlets in the same city and the quality of the food may vary depending on the location of the store. This is where the Restaurant Recommendation System comes into play. Our system helps the user to choose a restaurant based on his taste, location, and estimated cost. Therefore, the information must be filtered and personalized about a particular user.

## III. RELATED WORK

### A. Aspect-Based Opinion Mining and Recommendation System for Restaurant Reviews (I)

The proposed system aims to extract the key features from reviews, along with the polarity weight of reviews. Polarity weight is done using POS tagging and Opinion mining. Opinion lexicon is used to create a list of negation words like not, nothing, never, etc. which negate the sentiment of the review, while positive words like too, most, very, etc. increase the magnitude of the sentiment. A user-based collaborative filtering technique is used where past history of the user is considered. The user-user similarity is calculated using PCC (Pearson correlation coefficient). Each review coexists with

some metrics showing the subjectivity and polarity of the review, as well as the user-user similarity. The end-user can also further filter the list of reviews by filtering only those that derive from his/her friends. In this work, a system was proposed that personalizes the order in which the reviews are shown and provides an inherent UI that permits the users to see the essential aspects of each review in a quick look.

### **B. *Fiducia: A Personalized Food Recommender System for Zomato (II)***

In the proposed system, they aim to plan a method that reflects customer choice in an environment where they can make the best choices. Inputs are taken here from previous user history; this paper also provides a look at some of the side dishes that best fit user preferences. The authors used the Stanford Dependency Parser to identify the words intended for each item in the review. They form a vector of these pieces and use a sentiment analysis on them to classify the sentiment of each object into categories. For sentiment analysis using 3 methods namely, Naive Bayes Classifier, Bag of Words (BoW), Long Short-Term Memory (LSTM), side food item recommendation is made based on topic modelling and community detection, making recommendations using collaborative filtering. The recommendations made give us a precision of 0.74

### **C. *Restaurant Recommendation System for User Preference and Services Based on Rating and Amenities: (III)***

The proposed system in this paper uses machine learning algorithms for personalized recommendation systems depending upon tripadvisor.com search data. All the information related to every hotel is stocked into the database. The reviews are parsed, tagged, and analysed to get the necessary insights obtained using Natural Language Processing (NLP). A lexicon-based approach is applied to distinguish between positive and negative reviews. Accordingly, the features are combined according to the user's viewpoint and a score is calculated for each sentence. Then we calculate the overall score for each review and subsequently use the database to store this sentimental end result. The system then uses reviews and comments to examine the restaurant's negative and positive sense. At last, the restaurants are sorted from best to worst and then suggested to the user. Various ML algorithms along with NLP are being suggested to determine personalized restaurant systems. We can observe that The NLP yields greater accuracy compared to other ML approaches. The Results depict 92%

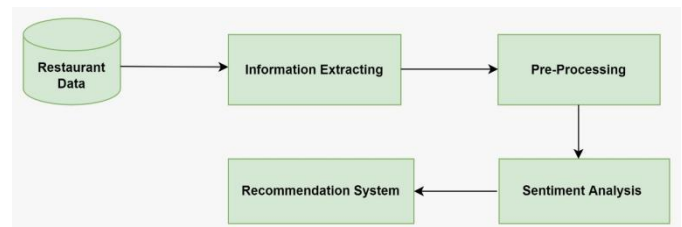
### **D. *Restaurant recommender system based on sentiment analysis: (IV)***

This research paper is divided into 2 parts extracting the user preferences and recommendations. It first extracts the

user's preferences from past reviews of restaurants then it recommends restaurants relevant to these preferences. The reviews are first pre-processed, which includes tokenization, stop word removing, POS tagging, stemming, noun extraction, and filtering. Since there are a large number of nouns, only nouns related to the food domain are filtered from the WordNet. If a noun is not found in food vocabulary and its synonyms of the WordNet, the noun is removed. Clustering techniques (WuPalmer method) like hierarchical and partitioning are used to categorize the nouns. Lastly, only those sentences which include words present in the cluster are considered and transferred to that cluster. Each sentence is then examined and classified as positive and negative based on sentiment. For sentiment analysis, the SentiWordNet dictionary is used. The recommendation is done based on the cluster that has the most score, which indicates which cluster the user is most likely to belong to. By applying these methods, the authors achieved 92.8%

## **IV. PROPOSED WORK**

In this paper, we attempt to make a personalized recommendation system that utilizes as much information from the dataset and provides the best restaurant recommendations. The important features considered for our analysis include review, ratings, dish\_liked, cuisines, cost, and location. The proposed Recommendation system is divided into four sections namely Information extraction, Pre-processing, Sentiment Analysis and Recommendation



**Fig1.1**

### **A. INFORMATION EXTRACTION**

This phase comprises of collection of information i.e., user's past history which contains the name of the restaurant, review given, dishes ordered, approximate cost spent. The user's current location and order time are also extracted to make personalized recommendations.

Since our Zomato dataset has reviews\_list which contains the reviews of all the users for each restaurant. So, for better analysis of the reviews, we made a separate CSV file named ratings.csv which contains individual reviews and ratings given by each user for each restaurant. This ratings.csv file has restaurant name, review, and the rate of each user.

### **B. PREPROCESSING**

Following issues were observed while cleaning the dataset.

**Inconsistent data:** Numerical columns were found to be inconsistent. For example, the cost column was expressed as “1,500” rather than 1500.00 which we cannot use for analysis, as well as the rating column was expressed in string format.

**Duplicate rows:** There was 28,375 number of duplicates present in our data, which were dropped later to avoid redundancy.

#### Missing values:

- **Dish liked:** The missing values present in this column were around 28,078. So, we filled the missing value with the menu item of that particular restaurant. If the menu item was also missing, then it was filled with dish liked of other franchises of the same restaurant.
- **Ratings:** It was observed that there were around 7,775 missing values out of 55,000 data rows. In order to handle it, we extracted mean ratings from the review’s column.
- **Approximate cost:** The missing values present in this column were around 346. As approx. the cost had outliers, we replaced missing values with a median.
- **Rest type:** The missing values present in this column were around 160. If the rest type is null, then its corresponding listed in type is taken then we group the dataset w.r.t listed in type, and then we take the mode of the rest type for the group-by dataset.

**Encoding of discrete and categorical variables:** The categorical variables in the dataset like online ordering and book table column have binary variables (‘Yes’ and ‘No’) which can be encoded to integer values (0, 1).

The cuisines, dish liked which are stored as string format for all the restaurants have now been split up and transformed into a list of strings format which can be analysed easily. (Example: “South Indian, Thai” is transformed as [ ‘South Indian’, ‘Thai’]).

- **Dish liked:** After the transformation, the dish names were pre-processed by removing numbers (like 450ml, 8 inches) and words (like special, delight, royal) which is irrelevant for our analysis. Then different types of similar dish items were grouped together. (Example: All different types of Maggi were grouped together).
- **Cuisine:** Since the cuisine’s column had some dish items, we replaced those items with their cuisine name.
- **Reviews:** Before analysing text for sentiment analysis, it should be cleaned and prepared first. Since the dataset had reviews in mixed languages, we encoded it using UTF-8 and then decoded it in ASCII.

We also added a new column named list\_review\_rate to store all the ratings given in the review column (where each review contains both the rating and review) given by different users which will be used later for analysis. Later, all the stop words, URLs, punctuations, and numerical data (i.e. rating) were removed.

#### • For Rating:

i) For Well established restaurants: (votes > 100)

$Avg = \text{Mean}(\text{list\_review\_rate})$

$X = \text{no. of votes}$

$Y = \text{Total No. of reviews}(\text{length of list\_review\_rate})$

$Z = X/2Y$

$W1 = Z/(Z+Y)$

$W2 = Y/(Z+Y)$

**Final rating = (W1\*Rate + W2\*Avg)**

ii) For newly established restaurants:

$Avg = \text{Mean}(\text{list\_review\_rate})$

$X = \text{no. of votes}$

$Y = \text{Total No. of reviews}(\text{length of list\_review\_rate})$

if  $X < Y$ ,

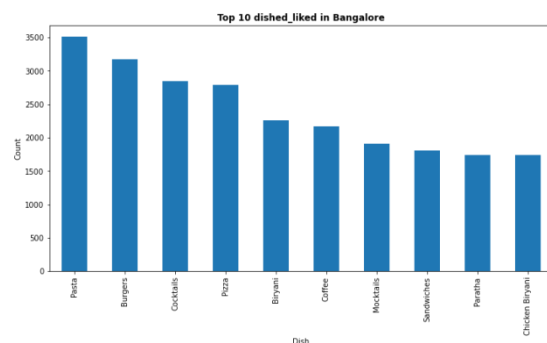
rating = rate.

if  $X > Y$

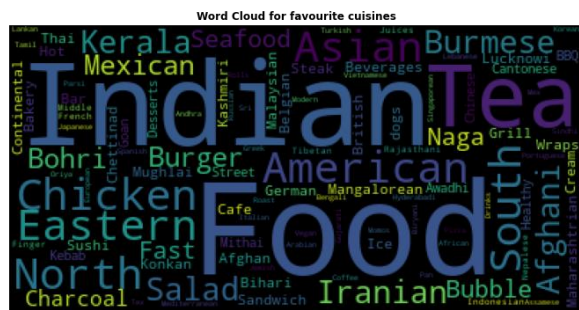
$W1 \text{ and } W2 = 0.5$

**Final rating = (W1\*Rate + W2\*Avg).**

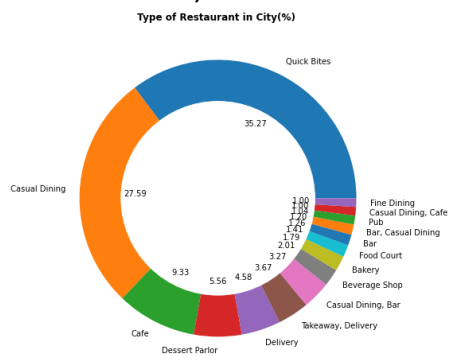
- **Ratings.csv:** All the duplicates based on two columns i.e., name and address were removed. Before analysing reviews for sentiment analysis, it should be cleaned first i.e., all the stop words, URLs, punctuations were removed. Then applied lemmatization which is a pre-processing technique that analyses the intended meaning of the word rather than its base form. Cleaned reviews were now lemmatized.



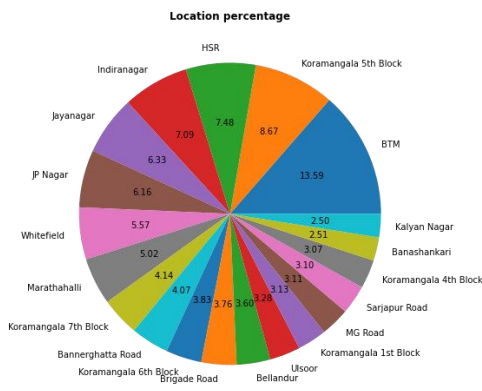
**Fig2.1) Observation: We can observe that top dishes preferred by Bangaloreans are Pastas, Burgers, etc.**



**Fig2.2) Most loved cuisines by Bangaloreans are North and South Indian, Fast Foods etc.**



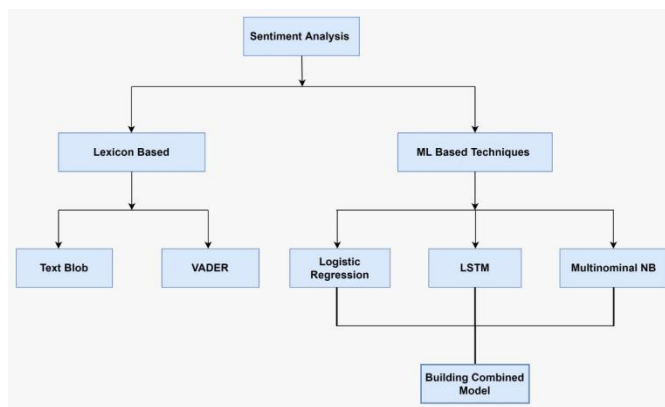
**Fig2.3) Observation: No doubt about this as it is known as the tech capital of India, people having modern and busy life will prefer Quick Bites.**



**Fig 2.4) Observation:** From the graph, we can conclude that Maximum no of restaurants are in BTM followed by Koramangala 5th block, HSR, IndiraNagar, so on.

### C. SENTIMENT ANALYSIS

In this section, we attempt to compare different sentiment analysis approach:



**Fig 3.1**

### Lexicon-based approach: Textblob and VADER

- **Text blob** is an open-source library for processing textual information. It performs different operations on data such as classification, POS tagging, translation, and sentiment analysis.
- **VADER** - Valence Aware Dictionary and Sentiment Reasoner, not only gives us positive or negative sentiment but also gives us polarity of it. The accuracy obtained from the text blob is 72%. Therefore, we can observe that both the approaches do not perform well to classify the reviews.

- **VADER** - Valence Aware Dictionary and Sentiment Reasoner, not only gives us positive or negative sentiment but also gives us polarity of it. The accuracy obtained from the text blob is 72%. Therefore, we can observe that both the approaches do not perform well to classify the reviews.

**Sentiment analysis using ML techniques:** There are number of Machine learning algorithms that are used to perform sentiment analysis. Here we have applied Logistic Regression, Random Forest Classifier, Decision Tree, Multinomial Naive Bayes, and LSTM models.

- **Logistic Regression:** This classifier utilizes the weighted combination of the information feature and after that applies the sigmoid function to it. The fundamental thought here is to partition the training dataset into positive and negative reviews. So, it counts all the words and makes a dictionary of their frequencies in positive and negative reviews.
- **Multinomial Naive Bayes:** It is one of the quickest and most straightforward classifiers for a large chunk of information. Rather than keeping the frequencies of each word with the positive and negative marks, we take the proportion of their recurrence in that label by the total number of frequencies. This will give the likelihood of the event of that word given the review is positive/negative.
- **Decision tree classifier:** It is a tree-organized classifier, given a set of records (for example addressed as TF IDF vectors) along with their names, the calculation will ascertain how much each word corresponds with a specific name.

- **Multinomial Naive Bayes:** It is one of the quickest and most straightforward classifiers for a large chunk of information. Rather than keeping the frequencies of each word with the positive and negative marks, we take the proportion of their recurrence in that label by the total number of frequencies. This will give the likelihood of the event of that word given the review is positive/negative.

- Decision tree classifier:** It is a tree-organized classifier, given a set of records (for example addressed as TF IDF vectors) along with their names, the calculation will ascertain how much each word corresponds with a specific name.

- **Adaboost:** It is a significant ensemble learning algorithm, which could upgrade a weak classifier which is better compared to a random guess to a strong classifier.
- **LSTM:** Long short-term memory is an artificial RNN engineering utilized in the field of deep learning. Dissimilar to standard feedforward neural networks, LSTM has feedback connections. It never keeps the whole information like standard RNN, LSTM keeps momentary memory of information.

ML algorithm	Accuracy
Logistic Regression	87%
Multinomial Naive Bayes	85%
Decision Tree Classifier	70%
AdaBoost	82%
LSTM	84%

Fig 3.2

It is observed that Logistic Regression performs better as compared to all the other models. But only considering Logistic Regression will not provide good results. Hence, we take the mode of results obtained from the best three models i.e. Logistic Regression, Multinomial Naive Bayes, and LSTM to provide the sentiment for the user data.

#### D. RECOMMENDATION

Cosine similarity constructs a matrix, which gives us the similarity between 2 vectors, it is computed by finding the cosine angle between the vectors. If the two vectors are directed in the same direction, then they are most relatable. Cosine similarity is applied on dish liked, cuisines as well as reviews, the matrix for dish\_liked and cuisines is constructed using count vectorizer.

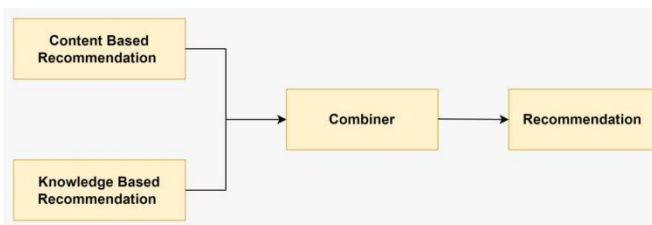


Fig 4.1

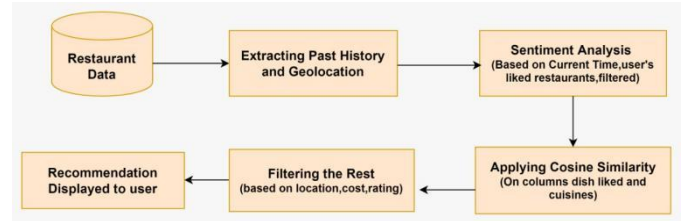


Fig 4.2

**Restaurant-Restaurant Similarity:** We used hybrid recommendation system which involve both content-based filtering and knowledge-based recommendation.

The final cosine similarity is the summation of the two cosine similarity matrix of cuisines and dish\_liked. We take the user's current order time into consideration and learn from the user's past behaviour and recommend restaurants relevant to it. We divide each day into 3 different timeframes (from 5:30 AM to 11:30 AM, from 11:30 AM to 4:00 PM, from 4:00 PM to 5:00 AM), and based on the current order time we extract all the restaurant names in that timeframe and apply sentiment analysis on the reviews of those restaurants to understand his taste better, now with the help of cosine similarity we find similar restaurants to it. Now we filter these restaurants' names based on the cost he usually spends in that timeframe and then filters out based on location (we take the users surrounding radius as 5KM) after that we considered rating (rating  $\geq 3$  considered as good) as the 3rd filter. And now finally while recommending, we sort it based on rating as well location.

**User-Restaurant Similarity:** This recommendation comes under instance-based learning, i.e., we apply a lazy algorithm. A lazy algorithm is one that requires zero training. Here we learn from the user's data, i.e., his dish liked and cuisines. We pre-process dish\_liked and cuisines and then calculate cosine similarity between the training data and users' history on dish\_liked and cuisines. After finding cosine similarities, we find the top 30 similar restaurants to users' data. Now we are applying filtering techniques as applied in the first recommendation system si.e. filtering based on cost, location, and rating.

#### V. RESULTS

**Restaurant-Restaurant Similarity:** From the below fig 5.1, we can observe various accuracies of different restaurants recommended by the model and the overall accuracy is found to be 74.2%

Restaurant	Accuracy
Rest 1	0.901
Rest 2	0.761
Rest 3	0.638
Rest 4	0.901
Rest 5	0.506

**Fig 5.1**

**User-Restaurant Similarity:** It is observed from the fig 5.2 that the food items preferred by the user based on his past history and order time included Biryani, Fries, Paratha, Rolls, etc. And the recommended restaurants by the model have these as their signature dishes.

Dish Name	Frequency
Biriyani	13/23
Fries	12/23
Paratha	8/23
Rolls	7/23
Butter Chicken	7/23

**Fig5.2**

## V. REFERENCES

- I. <https://dl.acm.org/doi/abs/10.1145/2645710.2645716>
- II. <https://arxiv.org/abs/1903.10117#:~:text=Fiducia%20is%20specific%20to%20popular,item%20and%20item%20Ditem%20similarity.>
- III. <https://www.semanticscholar.org/paper/Restaurant-Recommendation-System-for-User-and-Based-Gomathi-Ajitha/b799dc4a197947850c9c8cdabe5a139f52c89223>
- IV. [https://www.sciencedirect.com/science/article/pii/S2666827021000578?\\_cf\\_chl captcha tk \\_=l4.2Q0VDP46oSZUtdv.dBPqS79Dn.hhV6MZCv5WB5DE-1638618109-0-gaNycGzNDFE](https://www.sciencedirect.com/science/article/pii/S2666827021000578?_cf_chl captcha tk _=l4.2Q0VDP46oSZUtdv.dBPqS79Dn.hhV6MZCv5WB5DE-1638618109-0-gaNycGzNDFE)
- V. <https://ieeexplore.ieee.org/document/9498534>