

A PROJECT ON

“RESTAURANT RECOMMENDATION SYSTEM”

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN BIG DATA ANALYTICS FROM CDAC



SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

Phase 2, Hinjewadi Rajiv Gandhi Infotech Park,
Hinjewadi, Pune – 411057.
MH-INDIA

SUBMITTED BY:

Praful Singhal

UNDER THE GUIDANCE OF:

Mr. Shubham More
Faculty Member
Sunbeam Institute of Information Technology, PUNE.



CERTIFICATE

This is to certify that the project work under the title 'Restaurant Recommendation System' is done by Praful Singhal in partial fulfillment of the requirement for award of Diploma in Big Data Analytics Course.

Mr. Shubham More
Project Guide

Mrs . Pradnya Dindorkar
Course Coordinator

Date : 30-JAN-2020

ACKNOWLEDGMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mrs. Pradnya Dindorkar (Course Coordinator, SIIT, Pune) and Project Guide Mr. Shubham More.

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Praful Singhal
DBDA August 2019 Batch,
SIIT Pune

TABLE OF CONTENTS

1. Introduction of Project

- 1.1. Statement of the problem
- 1.2 Data Analysis
- 1.3 Goal
- 1.4 Project Scope & Limitations
- 1.5 Overview of proposed solution approach

2. Product Overview and Summary

- 2.1 Purpose
- 2.2 Scope
- 2.3 Dataset

3. Requirements and feasibility Study

- 3.1 Collaborative Filtering
- 3.2 Content-based Filtering
 - 3.2.1 Limitation of Content-Based Filtering
- 3.3 Software Requirement

4. System Design and Architecture

- 4.1 Use Case Diagram
- 4.2 System Flow diagram
- 4.3 Proposed Algorithm
 - 1. NLTK
 - 2. TextBlob
- 4.4 User Interface Snapshots
 - 4.4.1 Home Page
 - 4.4.2 Recommendation Input Page
 - 4.4.3 Restaurant Recommendation Page
 - 4.4.4 Tableau Visualization Graphs
 - 4.4.5 About Us

5. CONCLUSIONS

6. BIBLIOGRAPHY

1. Introduction of Project

As modern consumers, we greatly benefit from Restaurant Recommendation Applications. It is so convenient to get a list of restaurants that match our preferences without much clicking, comparing, and browsing through a long list of reviews for each single business.

In this project, we want to reduce human effort of filtering a lot of things and provide them a fine User Interface with less filtering and selection options so that every customer can find a better restaurant in couple of seconds.

1.1. Statement of the problem

In the past, people obtained suggestions for restaurants from friends or other conventional sources or sites. Although this method is straightforward and user-friendly, it has some severe limitations. First, the recommendations from friends or other common people are limited to those places they have visited before. Thus, the user is not able to gain information about places less visited by their friends. Besides that, there is a chance of users not liking the place recommended by their friends. Second, the information provided by the site can often be biased; thus the information provided cannot always consider being accurate.

1.2 Data analysis

These kind of analysis can be done using the data:

- Approx Price of food
- Location of the restaurant
- Theme based restaurant or not
- Which locality of that city serves that cuisines with maximum number of restaurants
- The needs of people who are striving to get the best cuisine of the neighborhood
- Is a particular neighborhood famous for its own kind of food.

1.2.1 File Format

The data is in csv format. For best results read the data using Python.

1.2.2 Content

The dataset contains 17 variables all of which were scraped from the zomato website. The dataset contains details of more than 50,000 restaurants in Bengaluru in each of its neighborhood. Detailed explanation about the variables are also available in section 6. The data is correct to the best of my knowledge, to that available on the zomato website until 15 March 2019.

1.2.3 Size

The total size of the dataset is approximately 547MB. The dataset examined has the following dimensions:

Feature	Result
Number of Observations	51,717
Number of Variables	17

1.3 Goal

The main goals of the application are:

- To Sentimental Analyze user reviews of different restaurants in Bangalore.
- To recommend restaurants to users based on Sentiment of previous user reviews using TextBlob Algorithm & Location and Cuisine of user's choice using MongoDB Database.

1.4 Project Scope & Limitations

The scope of the applications are as follows:

- The restaurants and hotels within the Bangalore City will be listed in the application.

The limitations of the application are as follows:

- Only the registered restaurants will be listed in the application.

1.5 Overview of proposed solution approach

The proposed solution approach for the given problem statement is Content based Filtering Recommendation System. Content-based filtering, also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analysing the content of items which have been seen by the user.

So, In this application, we retrieve the data of bangalore restaurant from zomato and take the reviews of previous and then Follow these steps:

Step 1: Tokenize all Reviews of Previous User Row-wise.

Step 2: Remove Punctuation From Lists of Tokenized Reviews.

Step 3: Remove Stop Words From Lists of Tokenized Reviews.

Step 4: Perform Sentimental Analysis on the Cleaned Reviews of Each Restaurant.

Step 5: Add New Column of Sentimental Analysis to the Data.

This Data is dumped into the MongoDB Database. So, the User can Just Select his/her Location & Cuisine. Then this application take that data from the user and fire the query to the MongoDB Database including Sentiment to be Positive and Fetch the Result and Display it to the User. This application is far better than asking a friend about any Restaurant because it is not Limited to any particular person.

2. Product Overview and Summary

2.1 Purpose

In the past, people obtained suggestions for restaurants from friends or other conventional sources or sites. Although this method is straightforward and user-friendly, it has some severe limitations. First, the recommendations from friends or other common people are limited to those places they have visited before. Thus, the user is not able to gain information about places less visited by their friends. Besides that, there is a chance of users not liking the place recommended by their friends. Second, the information provided by the site can often be biased; thus, the information provided cannot always consider being accurate.

So, the purpose of this application is to make a recommendation that is not limited to any particular person or a friend who have visited that particular restaurant and ordered any particular dish because that person is only know about that dish but this system contains all the reviews of that restaurant and have the sentiment of that restaurant that is very helpful in recommending a better restaurant based on user's requirement.

2.2 Scope

The restaurants and hotels within the Bangalore City will be listed in the application. Only the registered restaurants will be listed in the application. This application only contains restaurant which are updated by the Admin.

2.3 Dataset

2.3.1 Initial Dataset

```
In [3]: df = pd.read_csv('/home/sunbeam/Downloads/zomatoRestaurant.csv')
```

```
In [4]: df
```

address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_liked	cuisines	approx_cost(for two people)	reviews
942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	080 42297555n+91 9743772233	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	North Indian, Mughlai, Chinese	800.0	["Rated 4 'RATED' beautiful pl t
2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	080 41714161	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chinese, North Indian, Thai	800.0	["Rated 4 'RATED' in I been here d
1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91 9663487993	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Cafe, Mexican, Italian	800.0	["Rated 3 'RATE' Ambie nc not tha
1st Floor, Annakuteera, 3rd Stage	Addhuri Udupi	No	No	3.7	88	+91 9620009302	Banashankari	Quick Bites	Masala Dosa	South Indian, North Indian	300.0	["Rated 4 'RATE' Great food : p

```
In [ ]:
```


2.3.2 Null Values in Initial Dataset

```
In [6]: df.isnull().sum()
```

```
Out[6]: address          0
        name             0
        online_order     0
        book_table       0
        rate             737
        votes            0
        phone            407
        location         0
        rest_type        96
        dish_liked       5248
        cuisines         5
        approx_cost(for two people) 136
        reviews_list     0
        listed_in(type)  0
        menu             0
        dtype: int64
```

2.3.3 Cleaned Dataset

```
In [11]: df.head()
```

```
Out[11]:
```

	address	name	online_order	book_table	rate	phone	type	dish_liked	cuisines	cost	city	Reviews
0	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	080 42297555, +91 9743772233	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	North Indian, Mughlai, Chinese	800	Banashankari	beautiful place dine interiors take back mugha...
1	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	080 41714161	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chinese, North Indian, Thai	800	Banashankari	dinner family turned good choose suitable ages...
2	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	+91 9663487993	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Cafe, Mexican, Italian	800	Banashankari	ambience good enough pocket friendly cafe quan...
3	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7	+91 9620009302	Quick Bites	Masala Dosa	South Indian, North Indian	300	Banashankari	great food proper karnataka style full meals t...
4	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8	+91 8026612447, +91 9901210005	Casual Dining	Panipuri, Gol Gappe	North Indian, Rajasthani	600	Banashankari	good restaurant neighbourhood buffet system pr...

2.3.4 Null Values in Cleaned Dataset

```
In [9]: df.isnull().sum()
```

```
Out[9]: address          0
        name             0
        online_order     0
        book_table       0
        rate             0
        phone            0
        type             0
        dish_liked       0
        cuisines         0
        cost             0
        city             0
        Reviews          0
        dtype: int64
```

2.3.5 Content of Review Column

```
In [13]: df['Reviews'][0]
```

```
Out[13]: 'beautiful place dine interiors take back mughal era lightings perfect went occasion christmas limited items avail
able taste service compromised complaint breads could better would surely like come dinner family weekday restaura
nt completely empty ambience good good old hindi music seating arrangement good ordered masala papad panner baby c
orn starters lemon corriander soup butter roti olive chilli paratha food fresh good service good good family hango
ut ncheers restaurant near banashankari bda along office friends visited buffet unfortunately provide veg buffet i
nquiring said place mostly visited vegetarians anyways ordered ala carte items took ages come food ok ok definitel
y visiting anymore went weekend one us buffet two us took ala carte firstly ambience service place great buffet lo
t items good good pumpkin halwa intm dessert amazing must try kulchas great cheers best thing place itâ\x83xã\x82x
ã\x83xã\x82xã\x83xã\x82xã\x83xã\x82xs ambience second best thing yummy food try buffet buffet food disappointed us
ntest nquality nservice staff professional friendly nnoverall experience excellent nnsubirmajumder wixsite com gre
at food pleasant ambience expensive coll place chill relax nnservice really good friendly staff nnfood nservice na
mbience noverall good ambience tasty food ncheese chilli paratha bhutta palak methi curry good combo nlemon chicke
n starters must try item negg fried rice also quite tasty nin mocktails recommend alice junoon miss canã\x83xã\x82
xã\x83xã\x82xã\x83xã\x82xã\x83xã\x82xt go wrong jalsa never fan buffet thus always order alacarteã\x83xã\x82xã\x83
xã\x82xã\x83xã\x82xã\x83xã\x82x service times slower side food worth wait overdelighted service food provided plac
e royal ethnic atmosphere builds strong essence india also quality taste food truly authentic would totally recomm
end visit place place nice comfortable food wise jalea outlets maintain good standard soya chaap standout dish cle
arly one trademark dish per must try nnthe concern parking congested limited cars basement parking steep makes cum
bersome place nice comfortable food wise jalea outlets maintain good standard soya chaap standout dish clearly one
trademark dish per must try nnthe concern parking congested limited cars basement parking steep makes cumbersome p
lace nice comfortable food wise jalea outlets maintain good standard soya chaap standout dish clearly one trademar
k dish per must try nnthe concern parking congested limited cars basement parking steep makes cumbersome '
```

2.3.6 All Locations Available in Dataset

```
In [16]: df['city'].unique()
```

```
Out[16]: array(['Banashankari', 'Bannerghatta Road', 'Basavanagudi', 'Bellandur',
'Brigade Road', 'Brookefield', 'BTM', 'Church Street',
'Electronic City', 'Frazer Town', 'HSR', 'Indiranagar',
'Jayanagar', 'JP Nagar', 'Kalyan Nagar', 'Kammanahalli',
'Koramangala 4th Block', 'Koramangala 5th Block',
'Koramangala 6th Block', 'Koramangala 7th Block', 'Lavelle Road',
'Malleshwaram', 'Marathahalli', 'MG Road', 'New BEL Road',
'Old Airport Road', 'Rajajinagar', 'Residency Road',
'Sarjapur Road', 'Whitefield'], dtype=object)
```

2.3.7 Shape of Final Dataset

```
In [10]: df.shape
```

```
Out[10]: (18656, 12)
```

2.3.8 Restaurant's Name in Dataset

```
In [23]: for i in range(len(df['name'].unique())):  
         print(df['name'].unique()[i])  
         i = i + 1  
i
```

```
Jalsa  
Spice Elephant  
San Churro Cafe  
Addhuri Udupi Bhojana  
Grand Village  
Timepass Dinner  
Onesta  
Penthouse Cafe  
Smaczneg  
Cafe Down The Alley  
Cafe Shuffle  
The Coffee Shack  
Caf-Eleven  
Cafe Vivacity  
Catch-up-ino  
Kirthi's Biryani  
T3H Cafe  
The Vintage Cafe  
Woodee Pizza  
Mr. Tea House
```

3. Requirement Analysis and feasibility Analysis

Recommender systems differ in the way they analyze these data sources to develop notions of affinity between users and items which can be used to identify well-matched pairs (Melville, 2010).

There are various approaches used in recommender systems. The most common procedures used for recommender system are content based filtering and collaborative filtering.

3.1 Collaborative filtering

Collaborative filtering is the type of recommendation algorithm that bases its predictions and recommendations on the rating or behavior of other users in the system. The fundamental idea of collaborative filtering is to find other users in the community that share opinions.

3.2 Content-based filtering

Content-based filtering refers to such methods that provide recommendations by comparing representations of content describing an item to representations of content that interest the user pairs (Melville, 2010).

Content-based filtering, also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analysing the content of items which have been seen by the user.

Several issues have to be considered when implementing a content-based filtering system. First, terms can either be assigned automatically or manually. When terms are assigned automatically a method has to be chosen that can extract these terms from items. Second, the terms have to be represented such that both the user profile and the items can be compared in a meaningful way. Third, a learning algorithm has to be chosen that is able to learn the user profile based on seen items and can make recommendations based on this user profile.

Music Recommendation systems in use web content-based filtering. The increase in multimedia data creates difficulty in searching information within user's desired time frame and according to the interest of the user.

Although the data processing time can be decreased by displaying results of songs which has been searched the most in past and present, this does not ensure that the results displayed matches the preference of the user.

3.2.1 Limitation of Content-Based Filtering

➤ Cannot handle fresh items

The prediction of the model for a given (user, item) pair is the dot product of the corresponding embeddings. So, if an item is not seen during training, the system can't create an embedding for it and can't query the model with this item. This issue is often called the **cold-start problem**.

➤ Hard to include side features for query/item

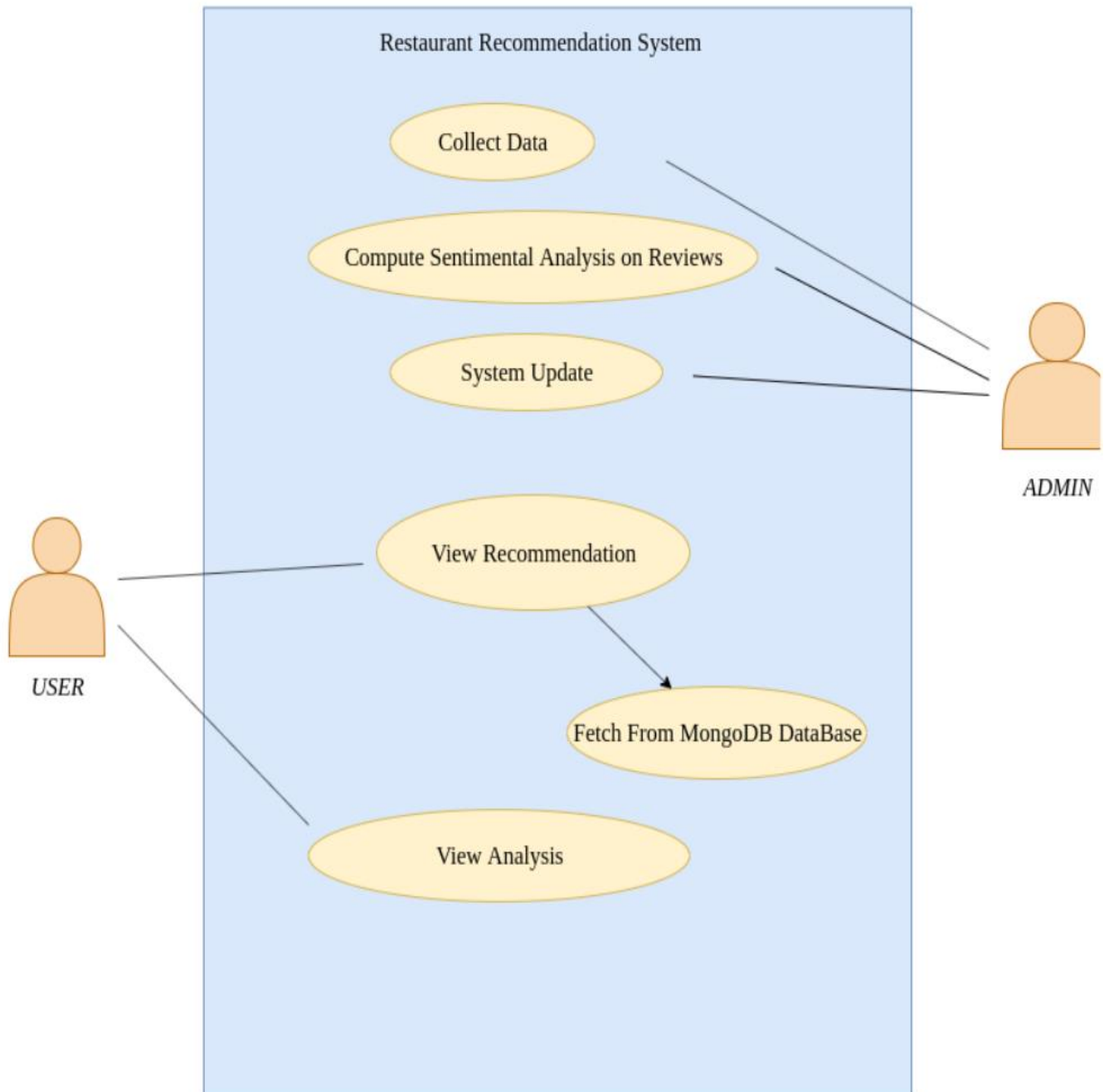
Side features are any features beyond the query or item ID. For movie recommendations, the side features might include country or age. Including available side features improves the quality of the model. Although it may not be easy to include side features in WALS, a generalization of WALS makes this possible.

3.3 Software Requirement

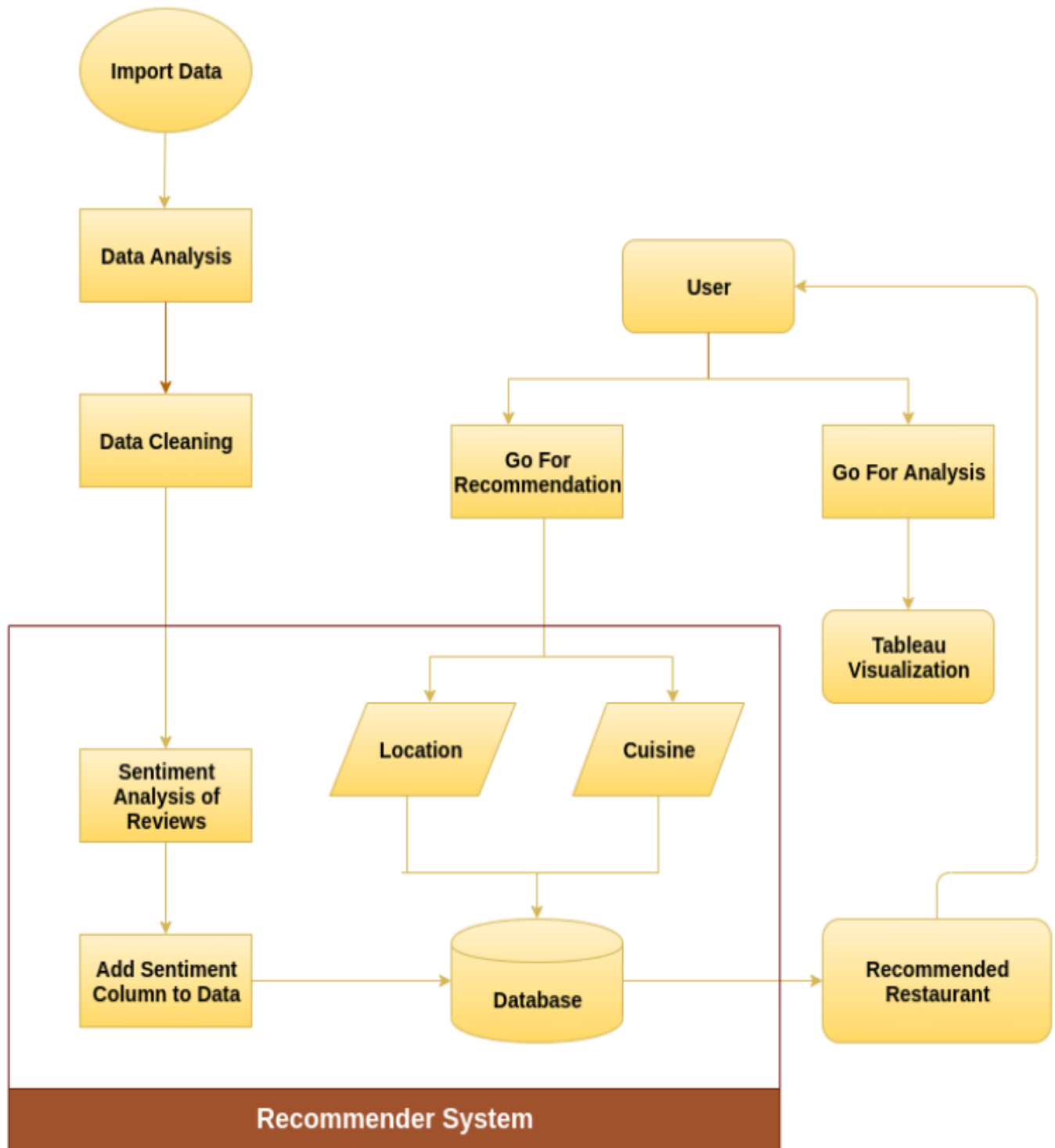
Software	Version
Python	3.6.8
Pandas	0.25.1
Numpy	1.17.2
Flask	1.1.1
MongoDB	4.0.11
Tableau	10.5

4. System Design and Architecture

4.1 Use Case Diagram



4.2. System Flow diagram



4.3 Proposed Algorithm

Natural Language Processing

Natural Language Processing is manipulation or understanding text or speech by any software or machine. An analogy is that humans interact, understand each other views, and respond with the appropriate answer. In NLP, this interaction, understanding, the response is made by a computer instead of a human.

Natural Language Processing (NLP) is an area of growing attention due to increasing number of applications like chat bots, machine translation etc. In some ways, the entire revolution of intelligent machines is based on the ability to understand and interact with humans.

1) Natural Language ToolKit (NLTK)

NLTK stands for Natural Language Toolkit. This toolkit is one of the most powerful NLP libraries which contains packages to make machines understand human language and reply to it with an appropriate response.

Steps of NLTK:

➤ Tokenization

Tokenization is the process by which big quantity of text is divided into smaller parts called tokens. We use the method `word_tokenize()` to split a sentence into words. The output of word tokenization can be converted to Data Frame for better text understanding in machine learning applications. It can also be provided as input for further text cleaning steps such as punctuation removal, numeric character removal or stemming. Machine learning models need numeric data to be trained and make a prediction. Word tokenization becomes a crucial part of the text (string) to numeric data conversion.

➤ Lemmatization/Stemming

Stemming is a kind of normalization for words. Normalization is a technique where a set of words in a sentence are converted into a sequence to shorten its lookup. The words which have the same meaning but have some variation according to the context or sentence are normalized.

In another word, there is one root word, but there are many variations of the same words. For example, the root word is "eat" and its variations are

"eats, eating, eaten and like so". In the same way, with the help of Stemming, we can find the root word of any variations.

Lemmatization is the algorithmic process of finding the lemma of a word depending on their meaning. Lemmatization usually refers to the morphological analysis of words, which aims to remove inflectional endings. It helps in returning the base or dictionary form of a word, which is known as the lemma. The NLTK Lemmatization method is based on WordNet's built-in morph function. Text preprocessing includes both stemming as well as lemmatization. Many people find the two terms confusing. Some treat these as same, but there is a difference between these both. Lemmatization is preferred over the former because of the below reason.

➤ **Remove Stop Words**

A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. We would not want these words taking up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to be stop words. NLTK(Natural Language Toolkit) in python has a list of stopwords stored in 16 different languages.

```
from nltk.corpus import stopwords
example_sent = "This is a sample sentence, showing off
the stop words filtration."

stop_words = set(stopwords.words('english'))
Filtered_sentence = [w for w in word_tokens if not w in
stop_words]
```

2) TextBlob

TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. A good thing about TextBlob is that they are just like python strings. So, you can transform and play with it same like we did in python. Below, I have shown you below some basic tasks. Don't worry about the syntax, it is just to give you an intuition about how much-related TextBlob is to Python strings.

Features of TextBlob

- Noun phrase extraction
- Part-of-speech tagging
- Sentiment analysis
- Classification (Naive Bayes, Decision Tree)
- Language translation and detection powered by Google Translate
- Tokenization (splitting text into words and sentences)
- Word and phrase frequencies
- Word inflection (pluralization and singularization) and lemmatization
- Spelling correction
- Add new models or languages through extensions

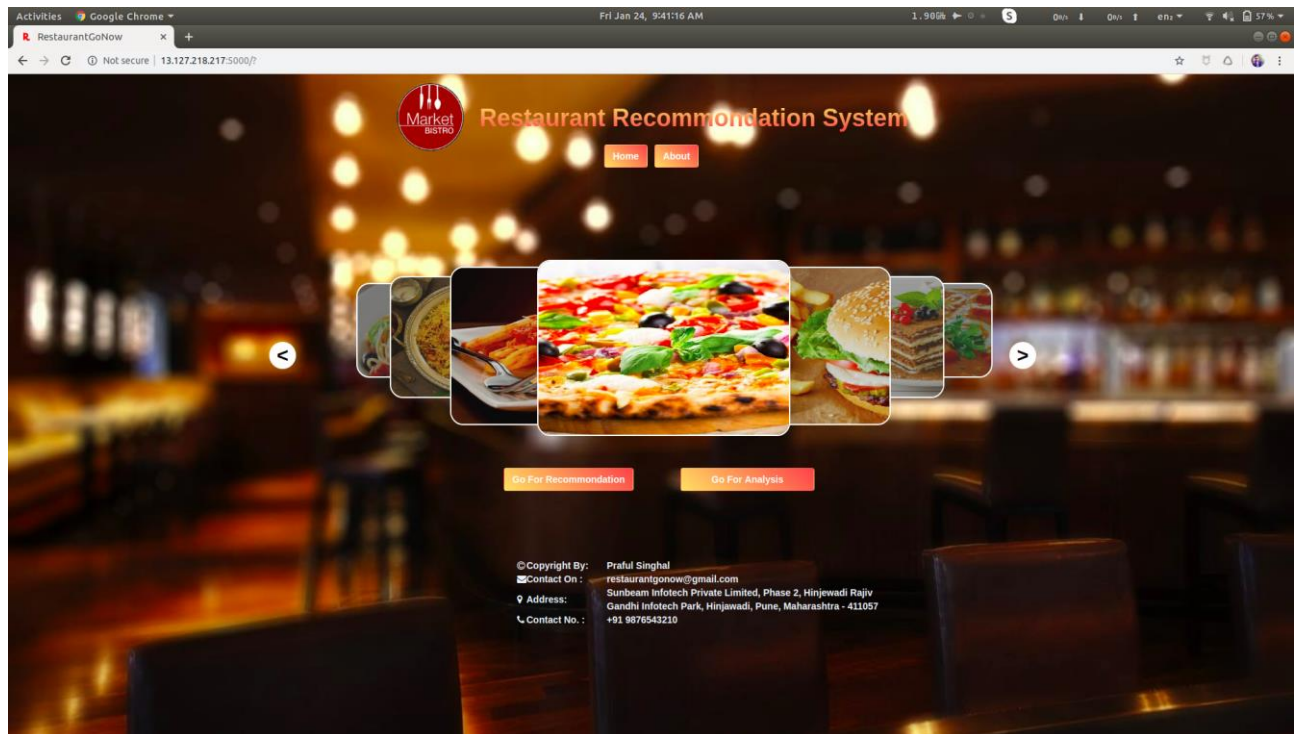
Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral. The *sentiment* function of textblob returns two properties, polarity, and subjectivity.

Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. Subjectivity is also a float which lies in the range of [0,1].

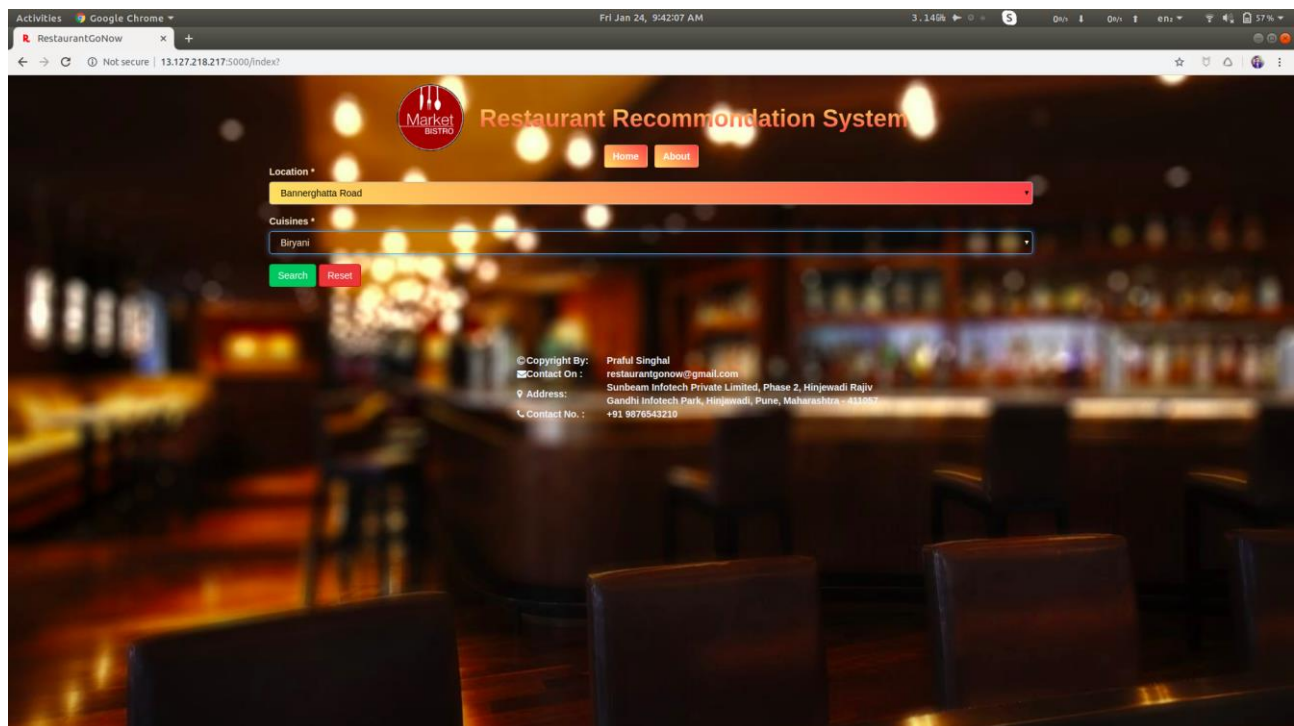
```
from textblob import TextBlob  
text = "This Restaurant is a nice place."  
polarity = TextBlob(text).polarity
```

4.4 User Interface Snapshots

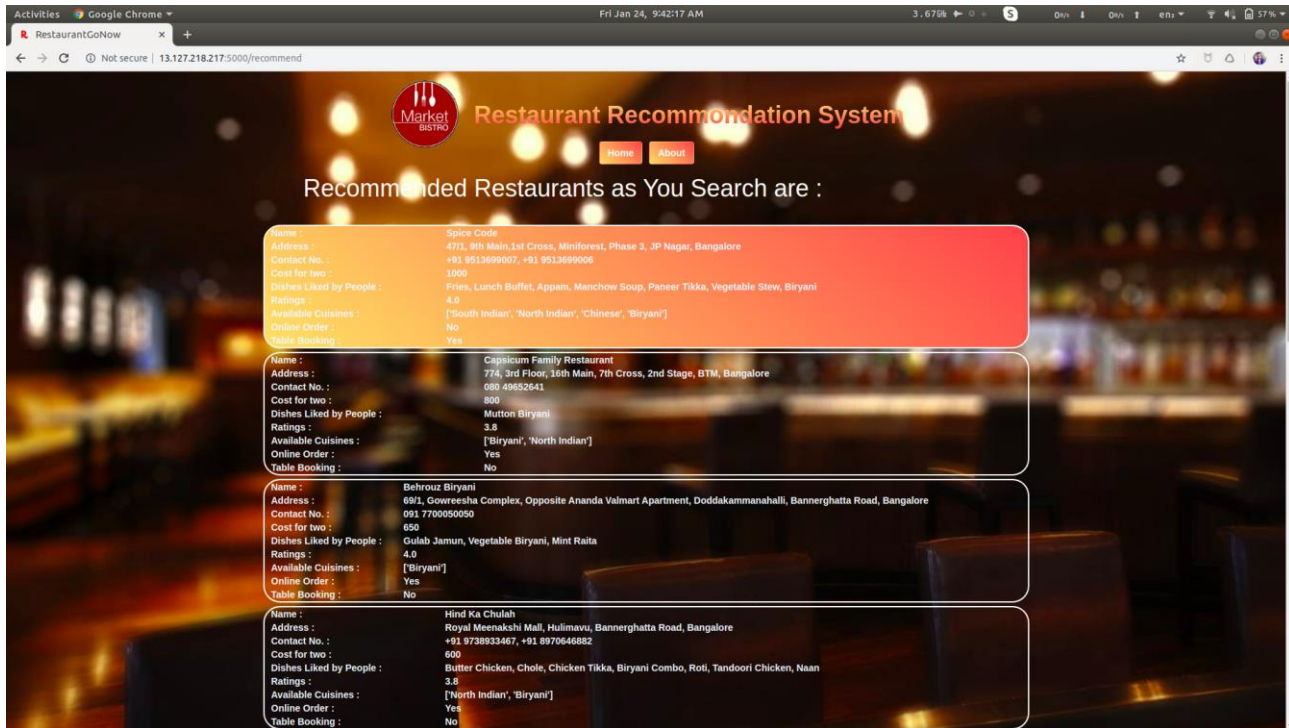
4.4.1 Home Page



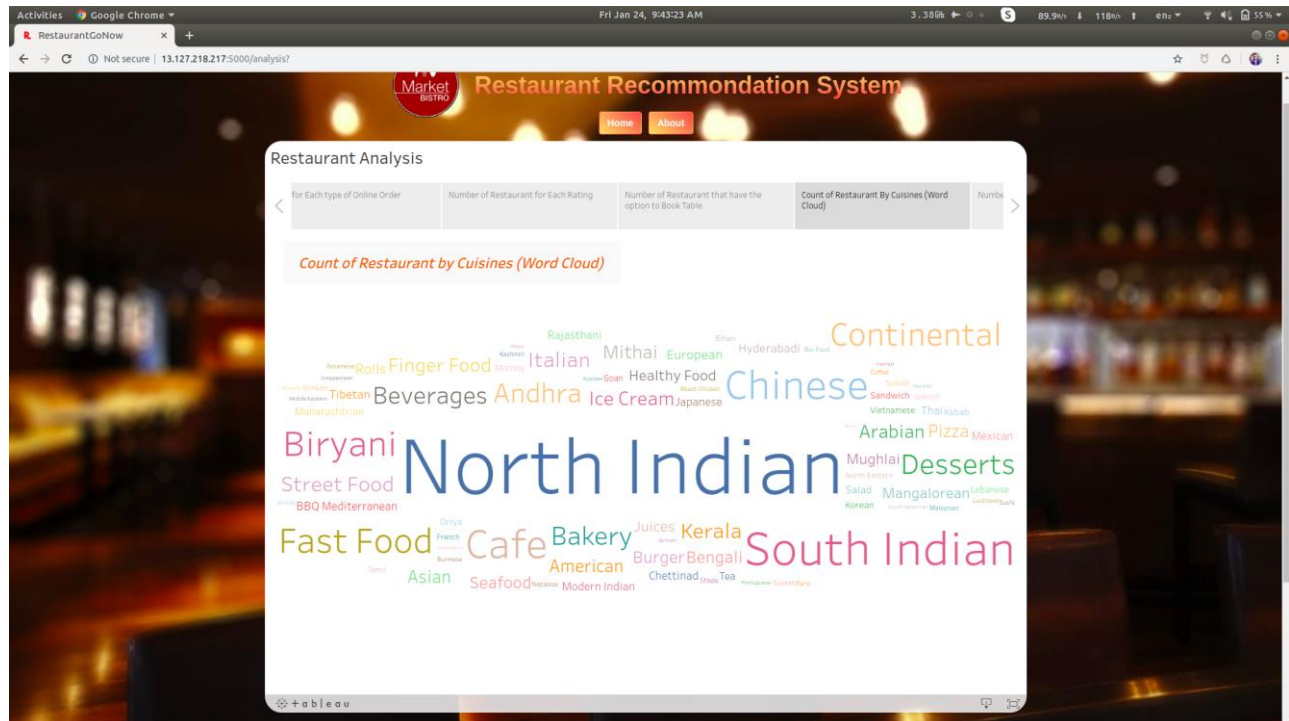
4.4.2 Recommendation Input Page



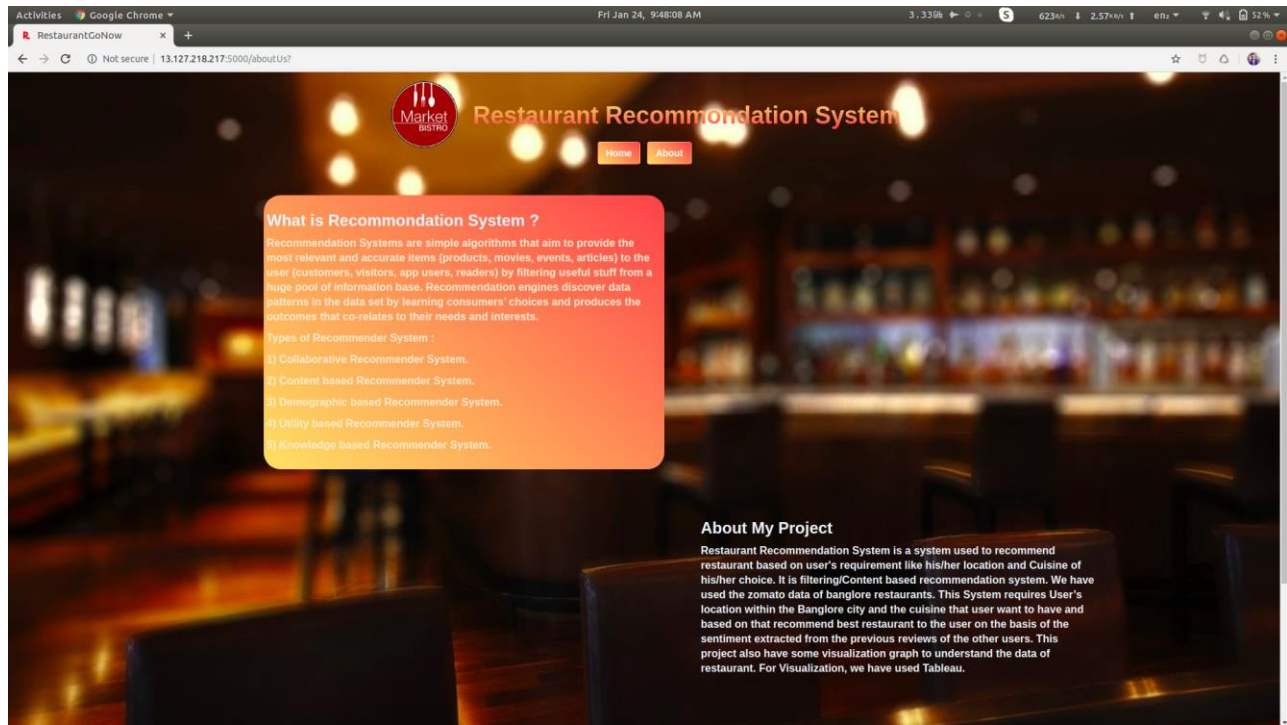
4.4.3 Restaurant Recommended Page



4.4.4 Tableau Visualization Graph



4.4.5 About Us



5. CONCLUSION

- The project Restaurant Recommendation System was successfully completed by using Content-based filtering.
- The data set were collected from zomato. The review from data were then used to Analyze the sentiment of previous users for every restaurant.
- In order to commercially use the product, it is important to collect data of all the restaurants in the Bangalore City.
- Furthermore, since the use of mobile phones is huge, the application will be more effective if built in mobile platform.
- Recommender systems have great value in recommending relevant resources to users. It can be quite useful in finding novel and serendipitous recommendations.
- The effectiveness of recommender system relies on the algorithm it uses to find interesting resources.
- In the beginning, current recommendation systems and main theoretical issues behind them are generally introduced.
- Afterwards the related work in the related area has been covered by analyzing a variety of recommendation systems from different domains.
- Subsequently, these techniques are examined in both positive and negative directions.
- In the most crucial part, comprehensive amount of study is done about overall system design and the prediction approach.
- Website URL is : <http://13.127.218.217:5000/>

6. BIBLIOGRAPHY

➤ Books

1. Learn Python 3 the Hard Way (Zed A. Shaw)
2. Introduction to Machine Learning with Python (Andreas C. Muller & Sarah Guido)
3. MongoDB in Action – Second Edition (Kyle Banker, Peter Bakkum, Shaun Verch, Douglas Garrett & Tim Hawkins)
4. Learning Tableau (Joshua N. Milligan)

➤ Websites

1. India's Restaurant Evolution
<https://www.hungrito.com/evolution-indian-restaurants/>
2. The Restaurant Industries
<https://marketrealist.com/2019/09/the-restaurant-industry-an-overview/>
3. There is a new French Restaurant Boom in Bangalore
<https://scroll.in/article/822487/theres-a-new-french-restaurant-boom-in-bengaluru>
4. Recommendation System
<https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>
5. Pandas
<https://pandas.pydata.org/pandas-docs/stable/>
6. Numpy
<https://numpy.org/doc/1.17/user/quickstart.html>
7. Flask
<http://flask.palletsprojects.com/en/1.1.x/quickstart/>
8. NLTK (Natural Language ToolKit)
<https://www.nltk.org/>