Yelp Restaurant Popularity Score Calculator

Sneh Bindesh Chitalia
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the Artificial Intelligence and Robotics Commons, and the OS and Networks Commons

Recommended Citation
DOI: https://doi.org/10.31979/etd.d644-pn6j
https://scholarworks.sjsu.edu/etd_projects/1261

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.
Yelp Restaurant Popularity Score Calculator

A Project
Presented to
The Faculty of the Department of Computer Science
San Jose State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Sneh Bindesh Chitalia
May 2023
The Designated Project Committee Approves the Project Titled

Yelp Restaurant Popularity Score Calculator

by

Sneh Bindesh Chitalia

APPROVED FOR THE DEPARTMENTS OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

May 2023

Dr. Katerina Potika       Department of Computer Science
Dr. Robert Chun          Department of Computer Science
Dr. William Andreopoulos Department of Computer Science
ABSTRACT

Yelp Restaurant Popularity Score Calculator

by Sneh Bindesh Chitalia

Yelp is a popular social media platform that has gained much traction over the last few years. The critical feature of Yelp is it has information about any small or large-scale business, as well as reviews received from customers. The reviews have both a 1 to 5 star rating, as well as text. For a particular business, any user can view the reviews, but the stars are what most users check because it is an easy and fast way to decide. Therefore, the star rating is a good metric to measure a particular business’s value. However, there are other attributes available on the platform that can be used to enhance recommendations.

In this project, we hypothesize that by considering six different attributes of reviews, users, and businesses we can enhance recommendations. Based on these attributes we generate an overall popularity score for each business. Furthermore, this popularity score is the possible identifier of the business’s value. We perform experiments on a Yelp available dataset, by using Natural Language Processing techniques, and neural network approaches.
ACKNOWLEDGMENTS

I want to express my gratitude to Professor Katerina Potika, who serves as my adviser, for her unwavering guidance and support, without which this project would not have been feasible. I value her time and dedication to helping me with this endeavor enormously. Her insightful counsel and in-depth understanding of the subject were invaluable.

I also want to express my gratitude to Professors Robert Chun and William Andreopoulos for kindly accepting to serve on my committee and offering their time to help me with my thesis.

Last but not least, I want to express my gratitude to my family, who has helped me both now and in the past.
# TABLE OF CONTENTS

## CHAPTER

1. **Introduction** ...................................................... 1
   1.1 Problem Definition ............................................. 2
2. **Terminology** .......................................................... 5
3. **Existing Methods** .................................................. 6
   3.1 Sentence Preprocessing ......................................... 6
   3.2 Feature Extraction ............................................... 6
   3.3 Sentiment Analysis ............................................... 7
      3.3.1 Naïve Bayes ................................................. 7
      3.3.2 Logistic Regression ....................................... 9
      3.3.3 Linear Support Vector Machine (SVC) .................. 9
      3.3.4 LSTM ...................................................... 10
      3.3.5 GloVe based models ..................................... 10
      3.3.6 BERT .................................................... 10
   3.4 Recommendation System ....................................... 11
      3.4.1 Neural Network-based Recommendations ............... 11
4. **Proposed Methods** ............................................... 18
   4.1 Architecture ................................................... 18
   4.2 Dataset .......................................................... 19
      4.2.1 Dataset Preprocessing .................................. 22
      4.2.2 Dataset Sampling ....................................... 22
4.3 Sentiment Analysis .................................................. 23
4.4 Parameter Identification ............................................. 24
  4.4.1 Normalization ................................................... 26
4.5 Multi-Parameter Model ............................................... 26

5 Results .................................................................. 31

6 Demo .................................................................. 36

7 Conclusion ................................................................. 42
  7.1 Future Work ......................................................... 42

LIST OF REFERENCES .................................................. 44

APPENDIX
<table>
<thead>
<tr>
<th>No.</th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Distribution of dataset</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>Distribution of dataset after sampling</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>Summary of each method</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>Models with their Accuracy for Sentiment Analysis</td>
<td>31</td>
</tr>
<tr>
<td>5</td>
<td>Score of each Neural Network model for Method 1</td>
<td>33</td>
</tr>
<tr>
<td>6</td>
<td>Score of each Neural Network model for Method 2</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>Score of each Neural Network model for Method 3</td>
<td>34</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Recommendation System from [1]</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Figure 2 from [2] shows how various parameters can be used to give product recommendations</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Common predicted variables in Sentiment Analysis from [3]</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>BERT model architecture explained in [4]</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Layers Of Neural Network explained in [5]</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>Graphical Representation of Tanh Activation Function from [6]</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Graphical Representation of ReLU Activation Function from [6]</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>Figure 2 from [7] shows Youtube Recommendation System using Neural Networks</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>Proposed Architecture</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>Comparative study of all the neural network models using Method 1</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>Comparative study of all the neural network models using Method 2</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>Comparative study of all the neural network models using Method 3</td>
<td>34</td>
</tr>
<tr>
<td>14</td>
<td>Login Page</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>Landing Page for all users</td>
<td>36</td>
</tr>
<tr>
<td>16</td>
<td>Landing Page with selection as Highest Rated</td>
<td>37</td>
</tr>
<tr>
<td>17</td>
<td>Landing Page with selection as Most Reviewed</td>
<td>37</td>
</tr>
<tr>
<td>18</td>
<td>Page for Each Restaurant</td>
<td>38</td>
</tr>
<tr>
<td>19</td>
<td>Highest Rated Reviews Each Restaurant</td>
<td>38</td>
</tr>
<tr>
<td>20</td>
<td>Newest First Reviews for Each Restaurant</td>
<td>39</td>
</tr>
</tbody>
</table>
21 Oldest First Reviews for Each Restaurant . . . . . . . . . . . . . 39
22 Lowest Rated Reviews for Each Restaurant . . . . . . . . . . . . 40
23 User Profile . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
24 User Profile for Other Users . . . . . . . . . . . . . . . . . . . . 41
CHAPTER 1

Introduction

Through their various users, social networks have a wealth of useful information stored as mentioned in [8]. These users according to [9] share their opinions, provide comments, and make mention of their feelings towards various subjects and people. All of this data can provide us with a wealth of knowledge and insight when it is analyzed for marketing and recommendation purposes. The reliability of the persons supplying these inputs is quite crucial in addition to the text inputs.

Online shopping and social networking platforms, for example, incorporate digital technology into business operations to address issues with outdated operational procedures, seize new opportunities, and provide customers with useful services. Additionally, integrating big data analysis with business processes will result in decision support systems that produce findings that are more accurate. Online reviews, on the other hand, have a huge volume of reviews, making it challenging for interested parties to read them all and assess the viewpoints and quality of the products.

Recommendation systems are an algorithm that presents customers with suitable product recommendations based on a number of variables. There is a mechanism in place when you search for a specific product on a website and start receiving several suggestions for related products. By proposing new products, saving users’ time, and limiting the options, it is typically used to target potential users and enhance the user experience. Recommendation systems are an important aspect of any social media platform. In most cases, recommendation systems fish out users with similar interests and then recommend similar items/products to those users. Figure 1 shows
an example of how the recommendation system work in general

![Diagram of recommendation system](image)

Figure 1: Recommendation System from [1]

Chapter 2 consists of terms to be known for the project. Chapter 3 describes methods that are already existing for some of the parts of the project. Chapter 4 describes the proposed method used to implement the project. It describes the advantages and disadvantages of the methods used. Chapter 5 describes the results attained in the project. Chapter 7 describes a conclusive end and also what scope there is in the future for the project.

1.1 Problem Definition

Even though most use cases of recommendation systems identify similar users and recommendations based on them, better approaches than that can be thought of. Thus, we propose a new method for recommending restaurants to users by generalizing the recommendations that will be given to the users based on location by default.

We propose a newer recommendation system, that takes into account various parameters available on the Yelp platform. Although this increases the processing we
have to do to generate recommendations, since it is generalized for all users, it is only a one-time process. The architecture diagram in Figure 2 is a good example of how various factors can be considered to provide recommendations.

Figure 2: Figure 2 from [2] shows how various parameters can be used to give product recommendations.

As you can see in Figure 2, first Sentiment Analysis is carried out on Dataset, then various features are identified which includes feature mapping and reviewer credibility. Thus an overall product ranking is formed which helps provide recommendations.
This new method will calculate one score per business (restaurant) called the Popularity Score. This is a generalized recommendation for all users that live in the same city. Users can apply filters including and not limited to the type of restaurant, etc.
CHAPTER 2

Terminology

In this chapter, we have the basic definitions and notations.

**Sentence Preprocessing:** Before Sentiment analysis can be done on any sentence, it requires a bunch of preprocessing steps to be carried out that make it easier for any machine to deduce the actual meaning of the sentence.

**Stopwords:** words that are often used but that a search engine has been trained to disregard when indexing entries for searching and when returning them as the result of a search query. Some examples include "the", "a", "an", "in".

**Lemmatization:** In order to find similarities between words, lemmatization, a text pre-processing technique, is employed in natural language processing (NLP) models. For example, lemmatization on the word "caring" will give "care".

**Stemming:** the procedure of eliminating affixes from a word so that the word’s stem remains. For example, stemming on the word "caring" will be reduced to "car".

**Feature Extraction:** One of the simple stages to be taken for a better knowledge of the situation we are in is feature extraction.

**Bag of Words (BOW):** a natural language processing (NLP) technique for transforming text into numerical data that a computer program may use.

**N-gram:** An N-gram means a sequence of N words.

**TF-IDF:** approach that determines the meaning of sentences made up of words in order to get over the constraints of the Bag of Words technique, which is useful for classifying texts or helping a machine interpret words in numbers.
CHAPTER 3
Existing Methods

3.1 Sentence Preprocessing

Various preprocessing steps have been carried out during the early days. These are still being used today. The preprocessing steps according to [10] and [11] are:

1. Removing Punctuation and Digits
2. Removing Stopwords
3. Lemmatization
4. Stemming

Sometimes, stemming and lemmatization both give the same reduced form of the word. However, stemming can sometimes hide the true meaning of a word. Hence, in most cases, lemmatization is used for sentence preprocessing.

3.2 Feature Extraction

Once Sentence preprocessing is carried out, various features are extracted from the sentence to get a better sense of the sentence. Feature Extraction ways proposed by [12] and [13] are:

1. Bag Of Words (BOW)
2. N-gram: Unigram, Bigram, Trigram
3. TF-IDF or ( Term Frequency(TF) — Inverse Dense Frequency(IDF) )
3.3 Sentiment Analysis

Sentiment analysis is a technique used in natural language processing (NLP) to determine how the connotation of the input. It determines whether the input is in support (positive), deny (negative), or neutral. Sentiment analysis on textual data is a popular practice among businesses to monitor how their brands and products are perceived by consumers through online reviews and to better understand their target market.

Sentiment Analysis can be done using classical Machine Learning Algorithms and Deep Learning Algorithms. A comparative study between classical machine learning algorithms has been carried out in [14]. Figure 3 explains the most commonly predicted variables when performing Sentiment Analysis.

![Figure 3: Common predicted variables in Sentiment Analysis from [3]](image)

3.3.1 Naïve Bayes

A Naïve Bayes classifier is an algorithm that uses the Bayes theorem to categorize items. Naïve Bayes classifiers make the assumption that the attributes of the data
points are strongly or naively independent. Naïve Bayes algorithm gives an accuracy of 79.12% on Yelp Dataset according to [14]

### 3.3.1.1 Multinomial Naïve Bayes

Each feature has a multinomial distribution that creates a feature vector that depicts how frequently that character appears in a given instance. Advantages of Multinomial Naïve Bayes:

- It is specially used for text processing, Accounts for repeating words and the number of repetitions
- Faster than plain Naïve Bayes

Multinomial Naïve Bayes gives an accuracy of 78.92% on Yelp Dataset according to [14]

### 3.3.1.2 Bernoulli Naïve Bayes

BernoulliNB implements the Naïve Bayes training and classification methods for data that is distributed according to multivariate Bernoulli distributions; i.e., in case of multiple factors, each of them is a binary variable. Advantages of Bernoulli Naïve Bayes:

- When compared to other classification methods, they are incredibly fast.
- It is quick and capable of making forecasts in real-time. Bernoulli Naïve Bayes gives an accuracy of 73.22% on Yelp Dataset according to [14]
3.3.2 Logistic Regression

Logistic regression is a technique for estimating the probability of a discrete result given an input variable.

Advantages of Logistic Regression:

- Simple to implement, Able to handle sparse data

Logistic Regression gives an accuracy of 78.88% on Yelp Dataset according to [14]

3.3.3 Linear Support Vector Machine (SVC)

Information that can be split into two categories with just a single straight line. Such data are classified using Linear SVM, and the classifier used is known as the Linear SVM classifier.

- When there is a large gap between classes, SVM performs comparatively well.

- In large dimensional spaces, SVM performs better. And SVM uses relatively little memory.

Linear Support Vector Machine gives an accuracy of 75.32% on Yelp Dataset according to [14]

The proposed method in [14] is to predict the output which is the mode of all the algorithms mentioned above In [14] and [9] both state that deep learning models can significantly improve the performance of sentiment analysis.

In this section, deep learning-based sentiment analysis methods will be explored. Long Short-Term Memory (LSTM) and Glove-based models give pretty good accuracy for sentiment analysis models.
3.3.4 LSTM

Deep learning, a sequential neural network referred to as Long Short-Term Memory Networks, allows for the persistence of information. Only a particular kind of recurrent neural network can solve the vanishing gradient problem that RNNs experience. RNNs have the disadvantage of being unable to recall long-term dependence because of the declining gradient. When designing LSTMs, long-term dependency problems are specifically avoided.

3.3.5 GloVe based models

GloVe is an unsupervised learning system that teaches word embeddings for various word vector representations. Global Vectors for Word Representations is known as GloVe. GloVe is essentially a weighted least-squares objective log-bilinear model. The key intuition behind the model is the fundamental discovery that ratios of word-to-word co-occurrence probabilities may be used to encode some sort of meaning. According to [14] and [9], these models give really good accuracy being in the range of around 90% - 92%. Another type of model that is heavily discussed is BERT.

3.3.6 BERT

Bidirectional Encoder Representations from Transformers is a language model that was made public in 2018 and achieved state-of-the-art performance on a range of tasks, including question-answering and language comprehension. It not only beat past cutting-edge computing models, but also the capacity of humans to provide answers. BERT converts text to numeric form. This phase is crucial because machine learning models can only be trained on text-based data using an algorithm that converts language to numbers. Machine learning models need numerical inputs—not
Figure 4: BERT model architecture explained in [4]

3.4 Recommendation System

A recommendation system is an artificial intelligence (AI) or machine learning algorithm that uses large data to advise or promote additional products to clients. These can be found out by looking at a variety of factors, including past purchases, search history, demographic information, and other things. Recommender systems are quite useful for guiding individuals toward goods and services they might not have discovered on their own. Various types of recommendation systems have been described in [8]. We use Neural Network based recommendation in our project which will be explained below:

3.4.1 Neural Network-based Recommendations

Neural Network based recommendations are basically generating a collaborative score based on various parameters. This score can now act as a recommendation system. Before diving into neural network-based recommendations, the foundation
of neural networks is explained below:-

A large number of interconnected processing components, sometimes referred to as Nodes, make up an artificial neural network. These nodes share a connection link with other nodes to connect them. Weights are present in the connection link and these weights contain information about the incoming signal. These weights are updated with each iteration and input. The final weights of the neural network and its architecture are known as the "trained neural network" after all of the training data examples have been entered. The training of neural networks is the name of this procedure. Layers are the vertically stacked parts that make up a neural network. Figure 5 gives an example of how each layer in the neural network is viewed.

Input Layer: the input layer, which is the first layer. This layer will accept the data and send it on to the rest of the network.

Hidden Layer: The Hidden layer is the second classification of the layer. There may be one hidden layer or several hidden layers in a neural network.

Output layer: The final category of the layer is the output layer. The output layer contains the answer to the issue.

Figure 5: Layers Of Neural Network explained in [5]
There are several types of neural networks:

1. ANN:-
   It is also known as an artificial neural network. Because the inputs are sent forward, it is a feed-forward neural network. Additionally, it might have undiscovered layers that would add to the model’s density. The programmer has set a preset length for them. It is utilized for tabular or textual data. Facial Recognition is a commonly utilized application in daily life. Compared to CNN and RNN, it is less effective.

2. CNN:-
   It is also known as Convolutional Neural Networks. It is mainly used for Image data. There are various real-life use cases for CNNs, being computer vision, object detection, autonomous vehicles, etc. It is more effective than RNN and ANN combined.

3. RNN:-
   It is also known as Recurrent Neural Networks. Data from time series are processed and interpreted using it. The output from a processing node in this kind of model is sent back into nodes in the same or earlier layers. RNNs of the LSTM (Long Short Term Memory) variety is the most popular.

   Neural Networks also have various Activation functions. It is merely a thing function that you use to obtain a node’s output. It also goes by the name Transfer Function. It is utilized to determine the neural network’s output, such as yes or no. The obtained values are mapped between 0 and 1 or -1 and 1, etc.
1. Sigmoid:

The sigmoid function has values between 0 and 1. The output of this activation function is generally a probability and as we know, probabilities can only lie between 0 and 1, in those use cases, the sigmoid activation function is used. Figure 6 explains a graphical representation of the Sigmoid Activation Function

Figure 6: Graphical Representation of Sigmoid Activation Function from [6]

2. Tanh:

The tanh graph will map the zero inputs close to zero and the negative inputs strongly negative, which is a benefit of this. The function could take many different shapes. Even though the function’s derivative is not monotonic, the function itself is. Figure 7 explains a graphical representation of the Tanh Activation Function

Figure 7: Graphical Representation of Tanh Activation Function from [6]

3. ReLU (Rectified Linear Unit):

The ReLU is currently the activation function that is used most frequently worldwide. Almost all deep learning and convolutional neural network systems
use it. Both the function and the derivative are monotonic. Figure 8 explains a graphical representation of the ReLU Activation Function

![ReLU Activation Function](image)

Figure 8: Graphical Representation of ReLU Activation Function from [6]

Neural Networks can be used for recommendations too. One of the most important things for any neural network model is the initialization of weights. According to [15] and [16], there are various ways to initialize weights in a neural network:

1. **Zero Initialization:**
   Weights are set to zero at the beginning. Each weight has the same value in subsequent iterations if all of the weights are initially set to 0, as each weight’s derivative with respect to the loss function is equal.

2. **Random Initialization:**
   Instead of only zeros, it is desirable to assign weights random values. What would occur if weights were initialized with incredibly high or low values, as well as what would be a realistic initialization of weight values, are two things to keep in mind.

3. **He Initialization:**
   We simply multiply random initialization with the square root of $2/size$ ** $(l-1)$ where $l$ represents layer number.
4. Xavier Initialization:

we just simply multiply random initialization with a square root of $1/\text{size}^{[l-1]}$

where $l$ represents layer number. It is mainly used for tanh() activation function

In [7] it shows that Youtube also uses a recommendation system that is built using Deep Neural Networks. Two neural networks make up the system: one for candidate generation and one for ranking. The candidate generation network uses events from the user’s YouTube activity history as input to choose a small subset (hundreds of videos) from a big corpus. These candidates should be extremely accurate and generally applicable to the user. The candidate generating network only provides substantial customisation through collaborative filtering. The level of user similarity is expressed in terms of broad data including demographics, search query tokens, and IDs of watched movies.

The presentation of a few of the "best" recommendations in a list requires a fine-level representation in order to distinguish between candidates with high recall relative relevance. By assigning a score to each video in line with a specified target function and making use of a plethora of features that characterize both the user and the video, the ranking network accomplishes this mission. The user is shown the highest-scoring videos in order of their score.
Figure 9: Figure 2 from [7] shows Youtube Recommendation System using Neural Networks.
CHAPTER 4
Proposed Methods

4.1 Architecture

In this architecture, there are different parts that lead to the Popularity Score. Figure 10 shows the proposed architecture. The main components are the following:

1. Dataset Preprocessing and Sampling: This component includes taking a small section of the dataset. Furthermore, it also involves cleaning the dataset.

2. Sentiment Analysis: This component includes resolving the connotation of the review text.

3. Parameter Identification: This component identifies different parameters that help in the generation of the popularity score.

4. Multi-Parameter Model: This component trains different Neural Network models with the parameters identified in the previous step and helps in generating the popularity score for each business.
4.2 Dataset

The distribution of the dataset has been explained in the table 1 below. As indicated in Table 1, the dataset from Kaggle has around 4.7 million reviews, 300k users and 100k businesses.

The dataset is taken from Kaggle [17]. There are 3 collections (JSON files) important for the implementation.

Business collection Json Object:
{ "business_id": string => unique identifier of the business
  "name": string
  "address": string
  "city": string
  "state": string
  "postal_code": string
  "latitude": float
  "longitude": float
  "stars": int => stars allotted to the business by the users, increasing from 1 to 5
  "review_count": int => total number of reviews on the business
  "is_open": int => whether the business is open or not, 0 or 1
  "attributes": { "ByAppointmentOnly": string => Appointment required or not }
  "categories": string => Type of services provided by the business
  "hours": NULL}
"business_id": string => unique identifier for each business
"stars": int => stars given to a particular review
"useful": int => count of users that find the review useful
"funny": int => count of users that find the review funny
"cool": int => count of users that find the review cool
"text": string => review text
"date": string => date the review was posted

User Collection Json Object:

{"user_id": string => unique identifier for each user
 "name": string
 "review_count": int => total number of reviews by user
 "yelping_since": string => day and time at which the user started using Yelp
 "useful": int => how many other users found this particular user useful
 "funny": int => how many other users found this particular user funny
 "cool": int => how many other users found this particular user as cool
 "elite": string => number of years the user has an elite status
}
"friends": => user_id of the other users that are friends with this user

4.2.1 Dataset Preprocessing

The whole dataset has around 4.7 million reviews. The review might be clean or might not be clean. Different Exploratory Data Analysis (EDA) were carried out on the dataset to see if the data is clean. All the columns are thoroughly examined to see that the data in fact has no issues with it. Since it is such a large dataset, there is no use in eyeballing the data to see if it is clean, rather Sentence Preprocessing has been carried out as a part of dataset processing. The most important aspect of the entire application is to get clean reviews. Thus, the text field from the Review JSON object which contains the review text is preprocessed using the processes explained in 3.

4.2.2 Dataset Sampling

A dataset sample has been considered since the whole dataset cannot be used due to a lack of computing reasons. A sample of the dataset has been taken for this project. Most existing sampling techniques take a random sample of the dataset which will not be very useful to us. Thus, the sample of the dataset it taken in such a way that it will help in the final output and at every step of the project. A total of 3000 business have been sampled. The conditions of Dataset Sampling are:-

- Reviews on business after the year 2020 are considered.
- Businesses are sorted in descending order by the number of reviews received (after 2020). Out of these businesses, the first 3000 most reviewed businesses
are sampled.

The distribution of the dataset after sampling is explained in table 2

<table>
<thead>
<tr>
<th>Reviews</th>
<th>Users</th>
<th>Business (Restaurants)</th>
</tr>
</thead>
<tbody>
<tr>
<td>350,000</td>
<td>190,000</td>
<td>3000</td>
</tr>
</tbody>
</table>

Table 2: Distribution of dataset after sampling

4.3 Sentiment Analysis

In Chapter 3, various Machine Learning and Deep Learning algorithms are discussed for Sentiment Analysis. Out of which, BERT outperforms the other methods, therefore, in the proposed method we use BERT for Sentiment Analysis.

Main Advantages of BERT:-

- BERT has been pre-trained on a lot of data and then made open-source:
  BERT has two models that have been pre-trained on approximately 800 million words and 2500 million words respectively.

- The model accounts for the context of words

In contrast to previous word-embedding methods, which always returned the same vector for a word, BERT returns various vectors for the same word based on the words around it.

BERT gives an accuracy of around 90% - 95%.

Since BERT uses Transfer Learning and has already been pre-trained on a vast number of parameters, it performs better than LSTM or GloVe-based models. This hypothesis is confirmed by [9]

Steps on using BERT for Sentiment Analysis:
1. Install the Hugging Face library

2. Load the input modules, the BERT Classifier, and the Tokenizer.

3. To produce a processed dataset, download the IMDB Reviews Data.

4. Configure and practice with the Loaded BERT model

5. Utilize the refined model to make predictions

Table 4 gives the accuracy of various Machine Learning models explained in Chapter 3 and Chapter 4.

### 4.4 Parameter Identification

Various attributes were identified as parameters from the Yelp dataset. The parameters identified were:

1. Stars of the review - Sum of the total number of stars given to the review by each user for a particular business.

2. Usefulness of review - Users can mark the review as useful if it helps them, this is a numerical parameter that is considered.

3. Sentiment of Review - Sentiment of the particular review (Positive, Negative).

4. Time Spent as an elite user - Elite is a concept introduced by Yelp that states that users who are regularly active on Yelp and post useful comments after a certain threshold are considered to be Elite users. The number of years spent by a user as an Elite is important because Elite carries more weight to a positive or negative review.
5. Usefulness of user - How many other users find reviews written by the particular user as useful

6. Stars of a business - Stars given to a business by users in the form of reviews.

The code for taking all these parameters into account is as follows:

```python
base_model_df = neuralNetworkData.groupby('business_id').agg(
    {'elite_in_positive': 'sum', 'elite_in_negative': 'sum',
     'reviews_useful': 'mean', 'reviews_stars': 'sum',
     'business_stars': 'mean', 'users_useful': 'mean'}).round(2)
```

These are the base parameters that are taken, using these parameters, composite parameters are also formed which help in generating the Popularity Score. The composite parameters generated are:

- **Elite in Positive** - How many Elite users have given Positive reviews to a business
- **Elite in Negative** - How many Elite users have given Negative reviews to a business
- **Influence of a User** - Influence is a composite parameter of Usefulness of User, Usefulness of reviews posted by that user, number of friends, number of fans and number of years spent as an Elite user.

The code for calculating influence of a user is as follows:

```python
influential_users['influence'] = (influential_users['useful'] + influential_users['fans'] + influential_users['number_of_elite_years'] + influential_users['number_of_friends'] + influential_users['reviews_useful'])/5
```
4.4.1 Normalization

Normalization of numerical data is very important since it brings the data in a specific range. For example, if a numerical column has values 100 and 2, a bias is created in the Machine Learning model which will overtrain it on the 100. Thus, normalization is very important since it helps bring these values in [0,1]. This will eliminate the bias created. After generating all the parameters in the Parameter Identification step and before passing the data onto the Multi-parameter model, all numerical parameters are normalized. Here, we are using Min-Max Normalization.

4.4.1.1 Min-Max Normalization

One of the most used techniques for normalizing data is min-max normalization. Every feature’s minimum and maximum values are each converted to a 0 and a 1, respectively, while all other values are converted to a decimal between 0 and 1. For instance, 30 would be translated to about 0.5 if the minimum and maximum values for a feature were 20 and 40, respectively. This is because 30 is halfway between 20 and 40. The formula is as follows:

\[
\frac{(value - \text{minimum value})}{(\text{maximum value} - \text{minimum value})}
\]

4.5 Multi-Parameter Model

The output after this stage is the Popularity Score of each business. This is an unsupervised Machine Learning challenge since there is no ground truth available that tells us the target popularity score of a business.

To solve this task, we are using Neural Networks that can identify patterns in the data and give a composite (popularity) score. We are using 3 different neural
networks:-

- Neural network with just one hidden layer
- Neural network with just two hidden layers
- Neural network with just three hidden layers

Here, we are also using 3 different methods each of them having a different formula to generate some parameters identified in the section above. So all in all, we are experimenting using 9 different neural network models and figuring out which works best for our use case.

- Method 1

Here, when calculating the number of elite users giving a positive review, the number of years spent as an elite user has been considered a binary variable. This means that the condition used here is that if the number of elite years > 0 and the review is positive, then elite_in_positive for that particular review and user is 1. Similarly, the elite_in_negative parameter has been generated. Below is the code representing how Elite in Positive has been calculated:-

```python
neuralNetworkData[ ’elite_in_positive ’ ] = np.where(
    (neuralNetworkData[ ’number_of_elite_years’ ] > 0) &
    ((neuralNetworkData[ ’sentiments’ ] == ’Positive’ ) ), 1, 0)
```

Factors identified for Method 1:-

- Elite in Positive
- Elite in Negative
- Reviews Useful - Mean number of the usefulness of reviews
- Users Useful - mean number of the usefulness of users
- Stars of Business - mean number of stars of business
- Stars of Review - total number of stars given to each review of business

**Method 2**

Here, when calculating the number of elite users giving a positive review, the number of years spent as an elite user has been considered a numerical variable. This means that the condition used here is that if the number of elite years > 0 and the review is positive, then elite_in_positive for that particular review and the user is (the number of years as elite * 1 if positive or 0 if negative).

Similarly, the elite_in_negative parameter has been generated.

Below is the code representing how Elite in Positive has been calculated:

```python
neuralNetworkData['elite_in_positive'] = np.where(
    (neuralNetworkData['number_of_elite_years'] > 0) &
    ((neuralNetworkData['sentiments'] == 'Positive') ),
    neuralNetworkData['number_of_elite_years'],
    0)
```

Factors identified for Method 2:-

- Elite in Positive
- Elite in Negative
- Reviews Useful - Mean number of the usefulness of reviews
- Users Useful - mean number of the usefulness of users
- Stars of Business - mean number of stars of business
- Stars of Review - total number of stars given to each review of business

- Method 3
In this method, a composite parameter is calculated called Influence. In [18], a formula is given which helps in the calculation of this influence parameter. The influence parameter is a composite parameter that is formed using the parameters: usefulness of each user, fans of that user, number of years spent as an elite user, number of friends the user has, and number of useful reviews.

The influence parameter is the average of all these parameters. Influence is calculated for each user. For each business, the influence of each user that has reviewed that business is taken into account. The code for calculating the influence parameter has been explained below:

\[
\text{influential\_users['influence']} = (\text{influential\_users['useful'] + influential\_users['fans'] + influential\_users['number\_of\_elite\_years'] + influential\_users['number\_of\_friends'] + influential\_users['reviews\_useful']})/5
\]

Factors identified for Method 2:-

- Elite in Positive
- Elite in Negative
- Reviews Useful - Mean number of the usefulness of reviews
- Stars of Business - mean number of stars of business
- Stars of Review - total number of stars given to each review of business
- Influence - Mean influence of all users that have reviewed the business

Here is a summary of all the Methods of the Multi-Parameter model being compared. Table 3 gives a Summary of all the methods including their advantages and disad-

<table>
<thead>
<tr>
<th>Methods</th>
<th>Method 1</th>
<th>Method 2</th>
<th>Method 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantages</td>
<td>Base Method. Not much calculation is required.</td>
<td>Considers the number of elite years as a numerical parameter.</td>
<td>Influence is calculated.</td>
</tr>
<tr>
<td>Disadvantages</td>
<td>Number of elite years are binary.</td>
<td>Effect on popularity score directly proportional to the number of elite years.</td>
<td>More processing and heavy calculation. Equal importance for calculating influence.</td>
</tr>
</tbody>
</table>

Table 3: Summary of each method

vantages. Since this is an unsupervised machine learning task, no target variable or ground truth is present to compare which model works better. Thus, as a measure to calculate performance, a Popularity Score is generated using the formula method where each attribute is given equal importance. This Popularity Score is generated by a formula compared with the Popularity Score calculated using neural networks. Whichever model has the least variation to this is considered the best model.
CHAPTER 5

Results

The results were already described in Chapter 3 and Chapter 4, where for sentiment analysis, different models were used and their accuracies were compared.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Bayes</td>
<td>79.12</td>
</tr>
<tr>
<td>Multinomial Naïve Bayes</td>
<td>78.92</td>
</tr>
<tr>
<td>Bernoullie Naïve bayes</td>
<td>73.22</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>78.88</td>
</tr>
<tr>
<td>Linear Support Vector Machine</td>
<td>75.32</td>
</tr>
<tr>
<td>LSTM</td>
<td>92</td>
</tr>
<tr>
<td>BERT</td>
<td>95</td>
</tr>
</tbody>
</table>

Table 4: Models with their Accuracy for Sentiment Analysis

The results of Sentiment Analysis on the IMDB dataset where the training and test set are at 80% - 20%. The target variable for Sentiments of the reviews is Positive and Negative. Thus, from Table 4, it can be clearly interpreted that BERT performs the best in terms of Sentiment Analysis.

We will now see the results of the multi-parameter model. We will compare the results of all 3 methods within themselves. Out of them, the best model will be found.

Since it is an unsupervised machine learning task, we need something to compare the performance of each model. Thus we calculate a formula score for each business and compare how far the popularity score of each business is from the formula score. We take a sum of the difference for all businesses. Whichever model is the least furthest from the formula score is the best. The parameters being considered for each method have been explained in Chapter 4.
Formula Score =

\[ \Sigma(\text{parameters identified for method})/(\text{number of parameters identified}) \]

The metric that we described above to assess the performance of a model will be referenced as a Score.

Score =

\[ \Sigma|\text{formulaScore} - \text{PopularityScore}| \]

The code for Score is:-

\[
\text{Score} = \text{sum}(\text{abs}(\text{target}_\text{df}[\text{`formula_score`] - \text{target}_\text{df}[\text{`popularity_score`}])))
\]

Method 1:

![Table](https://example.com/table1)

Figure 11: Comparative study of all the neural network models using Method 1

In Figure 11, formula_score is the established ground truth proposed in Chapter 4. model_1, model_2, and model_3 are the Popularity Scores predicted by each of the neural network models.

We will now have a look at how far the predicted scores are from the ground truth. Here, the score or the measurement metric is calculated by taking the absolute difference between the formula score and the model score and then summing those up column-wise.
Table 5: Score of each Neural Network model for Method 1

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network Model_1</td>
<td>0.406</td>
</tr>
<tr>
<td>Neural Network Model_2</td>
<td>0.466</td>
</tr>
<tr>
<td>Neural Network Model_3</td>
<td>1.004</td>
</tr>
</tbody>
</table>

Table 5 gives a score using each of the neural network models proposed in Chapter 4. Here, as we can see the score for model_1 is lowest. Thus, for method 1, model_1 is the best model.

Method 2:

Figure 12: Comparative study of all the neural network models using Method 2

In Figure 12, formula_score is the established ground truth proposed in Chapter 4. model_1, model_2, and model_3 are the Popularity Scores predicted by each of the neural network models.

We will now have a look at how far the predicted scores are from the ground truth. Here, the score or the measurement metric is calculated by taking the absolute difference between the formula score and the model score and then summing those up column-wise.

Table 6 gives a score using each of the neural network models proposed in Chapter 4. Here, as we can see the score for model_3 is lowest. Thus, for method 2,
model_3 is the best model.

Method 3:

In Figure 13, formula_score is the established ground truth proposed in Chapter 4. model_1, model_2, and model_3 are the Popularity Scores predicted by each of the neural network models.

We will now have a look at how far the predicted scores are from the ground truth. Here, the score or the measurement metric is calculated by taking the absolute difference between the formula score and the model score and then summing those up column-wise.

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network Model_1</td>
<td>0.394</td>
</tr>
<tr>
<td>Neural Network Model_2</td>
<td>0.389</td>
</tr>
<tr>
<td>Neural Network Model_3</td>
<td>0.258</td>
</tr>
</tbody>
</table>

Table 6: Score of each Neural Network model for Method 2

<table>
<thead>
<tr>
<th>business_id</th>
<th>formula_score</th>
<th>model_1</th>
<th>model_2</th>
<th>model_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.185425</td>
<td>0.202642</td>
<td>0.182042</td>
<td>0.192972</td>
</tr>
<tr>
<td>1</td>
<td>0.313842</td>
<td>0.308937</td>
<td>0.311129</td>
<td>0.314891</td>
</tr>
<tr>
<td>2</td>
<td>0.268249</td>
<td>0.257931</td>
<td>0.276301</td>
<td>0.252197</td>
</tr>
<tr>
<td>3</td>
<td>0.324883</td>
<td>0.324402</td>
<td>0.319686</td>
<td>0.326857</td>
</tr>
<tr>
<td>4</td>
<td>0.193093</td>
<td>0.203153</td>
<td>0.200720</td>
<td>0.189204</td>
</tr>
</tbody>
</table>

Figure 13: Comparative study of all the neural network models using Method 3

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network Model_1</td>
<td>0.205</td>
</tr>
<tr>
<td>Neural Network Model_2</td>
<td>0.221</td>
</tr>
<tr>
<td>Neural Network Model_3</td>
<td>0.180</td>
</tr>
</tbody>
</table>

Table 7: Score of each Neural Network model for Method 3
Table 7 gives a score using each of the neural network models proposed in Chapter 4. Here, as we can see the score for model_3 is lowest. Thus, for method 3, model_3 is the best model.

As it can be seen from each of the result Tables 5, 6 and 7, model 3 of method 3 has the lowest Score and is thus the best model that can help us in predicting the Popularity Score.
CHAPTER 6

Demo

This chapter includes the workflow of the demo.

1. Login Page

Figure 14: Login Page

Figure 14 gives the basic login page for each and every user

2. Landing Page

Figure 15: Landing Page for all users
Figure 15 gives a list of all restaurants in the city of the logged-in User. The restaurants are initially given in the Recommended order which is according to decreasing order of Popularity Score.

Figure 16: Landing Page with selection as Highest Rated

Figure 16 gives a list of all restaurants in the city of the logged-in User. The restaurants are given in the Highest Rated order which is according to decreasing order of Business stars as selected in the drop-down menu.

Figure 17: Landing Page with selection as Most Reviewed

Figure 17 gives a list of all restaurants in the city of the logged-in User. The
restaurants are given in the Most Reviewed order which is according to decreasing order of review count as selected in the drop-down menu.

3. Restaurant Page

Figure 18: Page for Each Restaurant

Figure 18 gives information about each restaurant. It gives the reviews for each restaurant, the location of the restaurant, etc.

Figure 19: Highest Rated Reviews Each Restaurant

Figure 19 gives a list of all reviews for that particular restaurant in the decreasing order of stars of reviews.
Figure 20: Newest First Reviews for Each Restaurant

Figure 20 gives a list of all reviews for that particular restaurant in decreasing order of date.

Figure 21: Oldest First Reviews for Each Restaurant

Figure 21 gives a list of all reviews for that particular restaurant in increasing order of date.
Figure 22: Lowest Rated Reviews for Each Restaurant

Figure 22 gives a list of all reviews for that particular restaurant in the increasing order of stars of reviews

4. User Page

Figure 23: User Profile

Figure 23 represents the user page for the logged-in user. Once the user icon on the Navigation bar is clicked and the profile is selected, the logged-in user profile page is displayed.
Figure 24: User Profile for Other Users

Figure 24 represents the user page for any user. If any names from the reviews in the restaurants page is clicked, then this user page is displayed with the information of the clicked user.
CHAPTER 7

Conclusion

In this project, we propose a way to enhance recommendations provided on the Yelp website. This new method takes inspiration from various types of recommendation systems available. It also takes into account previously unknown factors like the inclusion of sentiments of reviews, stars of the business, stars of a review, etc. Furthermore, we take advantage of a new composite parameter called Influence proposed in [18]. This makes the recommendations more comprehensive by taking more attributes to calculate the Popularity Score.

Chapter 5 shows that we successfully achieved our goal. Our proposed implementation is able to calculate a generalized Popularity Score for each business based on the factors discussed in Chapter 4. Furthermore, this Popularity Score can now be used as a way to enhance recommendations by providing businesses having higher Popularity Scores as recommendations.

7.1 Future Work

While the work we have done may have significant improvements, there is still more we could do to improve the method by which we generate the Popularity Score.

Currently, we have identified 6 different parameters to generate the Popularity Score. We can still identify more parameters that can be used. More composite parameters can also be identified.

Currently, we are using Neural Networks for this task. We can figure out a more extensive approach to do this. We can use Self Organizing Maps (SOM) which use
neural networks to cluster the data. We can then do some processing and generate the Popularity Score according to each cluster. We can also use Adaptive Resonance Theory (ART) instead of SOMs and then generate the Popularity Score. ARTs are a more unified version of SOMs.
LIST OF REFERENCES


