**Sentiment Analysis of Customer Feedback on Restaurants Using NLP:**

**Course:** Data Analytics

**Course code:** CSE4027

**Team:** 17

**Done by,**

Aadil Mohammed, 19BCI7052 (Team Lead)

Jammula Sri Ranga Jayanth, 19BCE7170

Gottipulla Sravan, 19BCE7634

Maladi Sri Raghavendra Uday kiran, 19BCE7520

Contents:

|  |  |
| --- | --- |
| **S.NO** | **Title** |
|  | Abstract   * Git repo link |
|  | Introduction: |
|  | Literature Review:   * Literature table * Challenges in existing system |
|  | Preliminary Concepts |
|  | Proposed Methodology |
|  | Experimental Framework   * Implementation details * Dataset details * Performance metrics * System specs |
|  | Results |
|  | Conclusion |
|  | Bibliography |

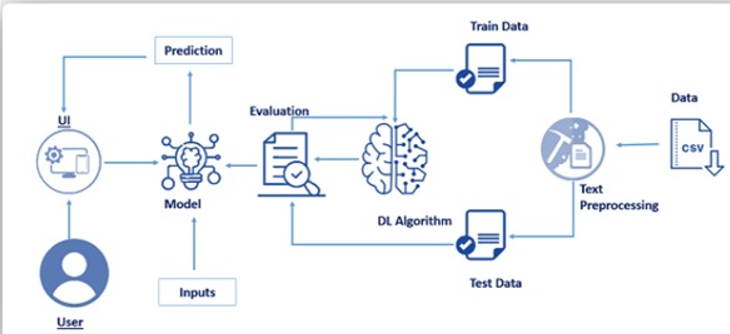
**Sentiment Analysis of Customer Feedback on Restaurants Using NLP:**

# **Abstract**

Most of the customers will follow and choose the best restaurants on the basis of reviews and ratings. So, reviews Play a Crucial Role in any Business model or system. The approach to this problem is based on review upon text content analysis and uses the principles of natural language process (the NLP method) and Machine learning. After applying the above method, we will classify whether it's a positive review or negative review and may also visualize the entire no of positive reviews and negative reviews. We are performing on developing an algorithm which will help in classifying the reviews on the idea of positive and negative reviews with the assistance of a predefined dictionary of words. The programming language of choice would be python, due our previous working experience and skills associated with the language.

Optionally (if time permits) we are interested in creating a Web Application where user can enter their feedback, the entered text is analysed by the model built and prediction is showcased on UI. This can be done using the Flask libraries in the python.

**GITHUB link: -** <https://github.com/Aadil-1041/DA_CSE4027_Team17>

**Architecture Diagram:**

# **Introduction:**

The most impactful marketing strategies of the recent times is by propagating general user consensus and feedback to new users to lure them into the product. This feedback is often being in terms of a product review which sums up the experience of the user. To give a review we usually use text to explain something that we experience with an item, place, or event that we normally experience.

Sentiment analysis, is that field of study people's opinions, sentiments, evaluations, judgments, attitudes, and emotions towards entities such as products, services, tourism, movies, organizations, political issues, individuals, problems, events, topics, etc. It is a type of data mining that measures the inclination of people's opinions through natural language processing (NLP), computational linguistics and text analysis, which are used to extract and analyse subjective information from the Web, mostly social media and similar sources.

# **Objective:**

Performing such analysis helps the organisations to maximise the customer satisfaction, which successively increase the mass appeal of the merchandise in e-commerce/ e-sites. Customer satisfaction is that the opinion between expectation and reality obtained by consumers. Giving a review is additionally a useful activity in order that other customer on the web can determine something else and see opinions about things and its satisfaction.

Commonly, most of the people express their opinion through social media like Facebook and Twitter or review platform like Zomato, Swiggy, Uber, Yelp, etc. Customer reviews on online media like Zomato become important because it might increase the recognition of something. Zomato is a site where someone can provides a review of a restaurant, how the restaurant is and someone's opinion about the restaurant. Review on Zomato remains within the sort of text and may be classified with positive, negative, or neutral with their ratings. Zomato doesn’t have an analysis of how users interact with the reviews and what words will indicate they like or not it. we'd like to extract the words in review and analysis it so we will skills users interact in Zomato and obtain customers satisfaction by their review.

# **Literature Review:**

One of the fundamental problem in performing Sentiment analysis is categorization of sentiment polarity. Given a piece of written text, the problem is to categorize the text into one specific sentiment polarity, positive or negative. In this application the reviews are categorized as Positive, Average, Negative.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.No | Author | Year of Publication | paper title | Methodology | Algorithm | Advantages | Disadvantage | Remarks |
| 1 | Jaspreet Singh,  Gurvinder Singh & Rajinder Singh | 2015 | Optimization of sentiment analysis using machine learning classifiers | Each review in the first dataset is parsed with NLTK’s parser and title of the review is considered as a feature. | Naive Bayes is used in NLP algorithms. | This paper exploits four machine learning classifiers for sentiment analysis using three manually annotated datasets. | The preprocessing of proposed methodology is limited to extract foreign words. | The future work in the task of sentiment analysis has scope to improve reprocessing with word embeddings using deep neural networks. |
| 2 | Dipak R. Kawade, Dr.Kavita S. Oza | 2017 | Sentiment Analysis: Machine Learning Approach | Six different sentiments can be analyzed using sentiment package namely anger, disgust, fear, joy. | Naive Bayes is used in NLP algorithms. | According to human psychology, any attack or terrorist activity generates panic. Present  work accurately classifies people’s emotions about Uri attack | The preprocessing of proposed methodology is limited to extract foreign word. | The future work in the task of sentiment analysis has scope to improve preprocessing with word embeddings  using deep neural networks |
| 3 | K. Sentamilselvan, D. Aneri,  A. C. Athithiya, P. Kani Kumar | 2014 | Twitter Sentiment Analysis using Machine Learning Techniques | The proposed methodology consists of four modules: (1) Data pre-processing: to pre-process the data (2) Feature Extraction (3) Machine learning classifiers. | Naive Bayes is used in NLP algorithms. | The best performance on the test set comes from the Logistic Regression | Dataset cannot be too large | This can be further implemented using the deep learning techniques |
| 4 | M S Kalaivani,S Jayalakshmi | 2019 | Sentiment analysis on micro-blog data using machine learning techniques | To initialize the wordembedding neural language is used.Embedded words and parameters are applied  to initialize the network,and training phase done by using supervised corpus | Naive Bayes is used in NLP algorithms. Support Vector Machine | This analysis provides the way for development in all the fields. | There are so many models established by using Machine learning and Deep learning Methods. Vector Machines | Dataset used for training and testing can be in different ratio with large subsets,which is randomly selected. |
| 5 | HARISH PARUCHURI,  SIDDHARTHA  VADLAMUDI,  ALIM AL AYUB  AHMED,  WESAM EID,  PRAVEEN KUMAR  DONEPUDI | 2019 | PRODUCT REVIEWS SENTIMENT ANALYSIS USING MACHINE LEARNING: | We followed the systematic literature review methodology by usingthis methodology we answered some research questions that help in understanding the literature related to thesentiment analysis. | Naive Bayes is used in NLP algorithms. Support Vector Machine | The best way to increasethe sales is the creation of the products that satisfies the customers. | There are so many models established by using Deep learning Methods. | The training data can be enlarged for better results in sentence dentification process. |

# **Preliminary concepts:**

Deep Learning can be considered as a small set of machine learning. It is a field based on learning and development on its own by testing computer algorithms. While machine learning uses simple concepts, in-depth learning works with sensory networks, which are designed to mimic the way people think and learn.

Recurrent Neural Networks (RNN) is a type of Neural Network where output from the previous step is provided as input to the current step. The most widely used RNNs, Sequential Editing - Emotional Sharing & Video Editing. Sequential Labelling - Part of speech tagging & recognition of a named business.

Another distinguishing feature of duplicate networks is that they share parameters in each network layer. Although feedforward networks have a different weight in each area, normal neural networks share the same weight parameter within each network layer. Having said that, these weights are still adjusted through backpropagation distribution processes and gradient downtime to facilitate reinforced learning.

General neural networks enhance backpropagation through time (BPTT) gradient-determining algorithm, which differs slightly from normal back-to-back distribution as defined in data tracking. The principles of BPTT are similar to normal backpropagation, in which the model trains itself by calculating errors from its output layer to the input layer. These calculations allow us to adjust and fit the model parameters correctly. BPTT differs from the general method in that BPTT incorporates errors in each step and feedforward networks do not need to combine errors as they do not share the parameters throughout each layer.

# **Proposed Methodology:**

Sentiment categorization is essentially a classification problem, where features that contain opinions or sentiment information should be identified before the classification. Every word of a sentence has its syntactic role that defines how the word is used. The syntactic roles are also known as the parts of speech.

There are 8 parts of speech in English: the verb, the noun, the pronoun, the adjective, the adverb, the preposition, the conjunction, and the interjection. In natural language processing, part-of-speech (POS) taggers have been developed to classify words based on their parts of speech. For sentiment analysis, a POS tagger is very useful because of the following two reasons:

1. Words like nouns and pronouns usually do not contain any sentiment. It is able to filter out such words with the help of a POS tagger.
2. A POS tagger can also be used to distinguish words that can be used in different parts of speech.

The Pre-processing techniques to be used in for the reviews are mentioned below along with their specific use case for the project.

## **Pre-Processing-1:**

We will begin by removing punctuation, number, converting each word into lower case, taking root of the words, and removing the stop words.

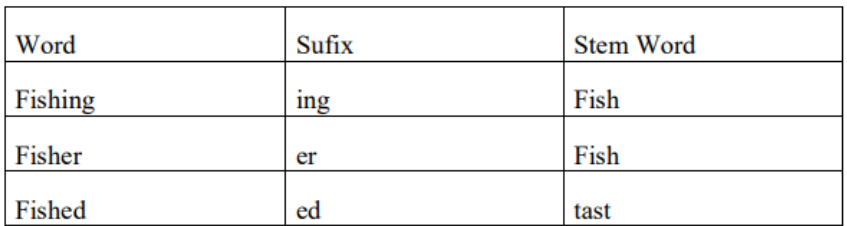
**Remove Punctuations**: Punctuations, Numbers don’t help much in processing the given text, so we will be using re library to replace all the punctuations, numbers with space while excluding the alphabets. All the special characters, numbers will be replaced by a space

**Convert each word into its lower case**: Words having the same meaning awesome, Awesome, AWESOME will be considered as non-identical words in the vector space model. So, it’s better to convert all the words into lowercase.

## **Pre-Processing-2:**

**Stemming or Lemmatization**

Words like ‘tasteful’, ‘tastefully’, etc are all variations of the word ‘tasty’. Hence, if we will have all these words in our text data, we will end up creating vectors for each of them when all these imply (more or less) the same thing. To avoid this, we can extract the root word for all these words and create a single vector for the root word. The process of extracting out the root word is called stemming.



**Removal of stop Words**

Stopwords are the English word which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For example, words like **the, he, have, that, what, etc**are considered as stop words

**Bag of Words with count Vectorizer**

Bag of Words (BOW) is a method to extract features from text documents. These features can be used for training machine learning algorithms. It creates a vocabulary of all the unique words occurring in all the documents in the training set.

In simple terms, it’s a collection of words to represent a sentence with word count and mostly disregarding the order in which they appear.

Let’s start with an example to understand by taking some sentences and generating vectors for those, Consider the two sentences given below:

* **"John likes to watch movies. Mary likes movies too."**
* **"John also likes to watch football games**."

These two sentences can be also represented with a collection of words.

* **['John', 'likes', 'to', 'watch', 'movies.', 'Mary', 'likes', 'movies', 'too.']**
* **['John', 'also', 'likes', 'to', 'watch', 'football', 'games']**

Further, for each sentence, remove multiple occurrences of the word and use the word count to represent this:

* **{"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1}**
* **{"John":1,"also":1,"likes":1,"to":1,"watch":1,"football":1, "games":1}**

Assuming these sentences are part of a document, below is the combined word frequency for our entire document. Both sentences are taken into account.

* **{"John":2,"likes":3,"to":2,"watch":2,"movies":2,"Mary":1,"too":1, "also":1,"football":1,"games":1}**

The above vocabulary from all the words in a document, with their respective word count, will be used to create the vectors for each of the sentences. The length of the vector will always be equal to vocabulary size. In this case the vector length is 11.

In order to represent our original sentences in a vector, each vector is initialized with all zeros — [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

This is followed by iteration and comparison with each word in our vocabulary and incrementing the vector value if the sentence has that word.

**John likes to watch movies. Mary likes movies too. [1, 2, 1, 1, 2, 1, 1, 0, 0, 0]**

**John also likes to watch football games. [1, 1, 1, 1, 0, 0, 0, 1, 1, 1].**

This concludes the text pre-processing in python using CountVectorizer, the remaining part is to split this processed data into test and train set and move on to the model building.

# **Experimental Framework: -**

## **Implementation details: -**

## **Text Pre-processing: -**

Text Pre-processing includes the following main tasks

* Import the Libraries.
* importing the dataset
* Remove Punctuations
* Convert each word into a lower case.
* Stemming.
* Splitting Data into Train and Test.

**Note:** These are the general steps of pre-processing the data before training. Depending on the condition of your dataset, you may or may not have to go through all these steps or steps may vary

**Import Libraries and Read the Data**

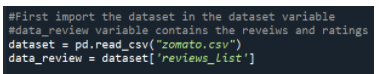
The first step is usually importing the libraries that will be needed in the program.

* Import numpy and pandas to your Python script:

A screenshot of a computer

Description automatically generated with medium confidence

**Read the Dataset :**



If you see the below picture, Review and Rating Towards review are placed in the same column. so we have to split the review and rating apart and store them in different variables

Loop over the review column and split the data properly to grab reviews and ratings and append them to declared empty lists x and y

Text

Description automatically generated

**Preprocessing -1:-**

Now clean the data, import necessary libraries to clean the data.

We will be using different libraries for the steps that are included in this step. “**Re**” is the library that is used to replace the selected special characters with the desired parameter. “**NLTK**” – Natural language Tool Kit is the library used for stemming using a special class in the library.

Text

Description automatically generated

* In the x variable, we have stored ratings which has both words and rating for ex :(Rating:4.0 )
* we would like to loop over all these ratings and grab only the numbers from it ( ex : 4.0)
* Then we assign the sentiment of the review based on some conditions to the Rating\_final list.

Text

Description automatically generated

**Preprocessing -2:-**

Let’s binarize or vectorize the textual sentiment in rating\_final  using label Encoding and One hot encoding

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

**Remove Punctuations:**

Punctuations, Numbers don’t help much in processing the given text, so we will be using re library to replace all the punctuations, numbers with space while excluding the alphabets.

Code line 89

* We declared a variable and assigned the first row in our y (review array).
* All the special characters, numbers will be replaced by a space

**Convert each word into its lower case.**

Words having the same meaning awesome, Awesome, AWESOME will be considered as non-identical words in the vector space model. So it’s better to convert all the words into lowercase.

**Stemming or Lemmatization with Stopword removal:**

Code line 92 does the stemming and stop word removal

To stem each word first we have to split the review into a list and then apply lemmatization functionality. We use PortStemmer() for stemming of the word.

**Preprocessing-3:-**

A screenshot of a computer

Description automatically generated with medium confidence

**Split The Data To Train And Test:-**

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

* Train Dataset: Used to fit the machine learning model.
* Test Dataset: Used to evaluate the fit machine learning model.

Graphical user interface, text

Description automatically generated

**LSTM(Long Short Term Memory):-**

Now that the data is cleaned and ready for model building, let's use the preprocessed data to train the model  Model building includes

* Training The Model - We make use of artificial Neural networks to train the model to
* Saving the model
* Testing the model

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

Diagram

Description automatically generated

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

A screenshot of a video game

Description automatically generated with medium confidence

# **Test And Save Model:-**

In this we perform text processing such that all the alphabets are retained and punctuations are removed.

After this, we convert all the capital letters into small letters.

Next, we search for stop words and remove stop words using NLTK packages.

**Dataset description:**

Customer reviews on online media like Zomato become important because it might increase the recognition of something. Zomato is a site where someone can provides a review of a restaurant, how the restaurant is and someone's opinion about the restaurant. Review on Zomato remains within the sort of text and may be classified with positive, negative, or neutral with their ratings. Zomato doesn’t have an analysis of how users interact with the reviews and what words will indicate they like or not it. we'd like to extract the words in review and analysis it so we will skills users interact in Zomato and obtain customers satisfaction by their review.

***Column fields:*** *url address name online\_order book\_table rate votes phone location rest\_type dish\_liked cuisines approx\_cost(for two people) reviews\_list menu\_item listed\_in(type) listed\_in(city)*

**Performance metrics:**

* Validation accuracy
* Cross entropy loss
* New test instances

**System specifications:**

* Google colab
* G Drive as cloud storage

# **Future Scope:**

* Customer Churn predictions,
* Tone Detectors,
* Similar implementation of the model for different platforms, which record user feedback.
* Build StreamLit Application: - In this section, we will be building a web application that is integrated to the model we built. A web app is provided for the users where we can enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the web app. This section has the following tasks
* Building HTML Pages
* Buildingserverside script

# **Project Work Flow:**

* Data Collection
* Text Pre-processing
  + Import the Libraries.
  + importing the dataset
  + Remove Punctuations
  + Convert each word into a lower case.
  + Stemming.
  + Splitting Data into Train and Test.
* Model Building
  + import the model building Libraries
  + Initializing the model
  + Adding Input Layer
  + Adding LSTM Layer
  + Adding Output Layer
  + Configure the Learning Process
  + Training and testing the model
  + Optimize the Model
  + Save the Model
* Application Building
  + Build StreamLit Web App

# **Flow Chart:**

Diagram

Description automatically generated

**Results:**

# An exhaustive approach of data analysis, of the dataset, has been done. All major aspects have been analysed.

# The model achieved an accuracy of about 0.95 and a loss of 0.12 with the model configs mentioned in the file.

**Conclusion:**

# This Zomato data generated aims at analysing demography of the location. Most importantly it will help new restaurants in deciding their theme, menus, cuisine, cost etc for a particular location. It also aims at finding similarity between neighbourhoods of Bangalore on the basis of food. The dataset also contains reviews for each of the restaurant which will help in finding overall rating for the place.

# **Bibliography:**

* <https://www.analyticsvidhya.com/blog/2020/11/create-a-pipeline-to-perform-sentiment-analysis-using-nlp/>
* <https://www.analyticsvidhya.com/blog/2021/06/nlp-sentiment-analysis/>
* <https://towardsdatascience.com/real-time-sentiment-analysis-on-social-media-with-open-source-tools-f864ca239afe>
* <https://ieeexplore.ieee.org/abstract/document/7219856/>
* <https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/>
* <https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/>