

# **Sentiment Analysis of Customer Feedback on Restaurants Using IBM Cloud**

## **1.INTRODUCTION**

### **1.1 Overview**

A good restaurant review advocates for the needs of potential customers and provides feedback the businesses may use to improve or evaluate their service. By providing balanced criticism and vivid descriptions, you can invite readers to experience the restaurant's atmosphere and cuisine for themselves.

Reviews not only have the power to influence consumer decisions but can strengthen a restaurant's credibility. Reviews have the power to gain customer trust, and they encourage people to interact with the restaurant. Customer interaction ultimately leads to improved profits for businesses.

Restaurants that are not accredited by experts sell out 27 percent more often if they earn an extra star on Yelp. According to the study, the closer the link between restaurants and customers through reviews, the greater the positive effects on business as a whole.

Sentiment mining plays a very important role in business to understand the opinion of customers to improve the food and quality of a restaurant. Customer also depends on the opinion of others who have tried the food already in the restaurant. Reviews or feedback becomes the deciding factor to increase the business. A review gives a speedy clarification to pact with the food in the restaurant. We will be using Natural language processing to analyse the sentiment (positive or a negative) of the given review.

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 model or system. The approach to this problem is based on review text content analysis and uses the principles of natural language process (the NLP method) and Machine learning. After applying the above method we can classify whether it is a positive review or negative review and can also visualize the total no of positive reviews and negative

reviews. We are working on developing an algorithm that can help in classifying the reviews on the basis of positive and negative reviews with the help of a predefined dictionary of words.

A web Application is created where user can enter their feedback, the entered text is analyzed by the model built and prediction is showcased on UI.

## **1.2 Purpose**

A restaurant review helps other users get a clear idea of the restaurant and food before going to it. They can read the reviews and make their mind clear, and decide whether the restaurant is worth go or not.

Sentiment mining plays a very important role in business to understand the opinion of customers to improve the restaurant. Customer also depends on the opinion of others who have went to restaurant already. Reviews or feedback becomes the deciding factor for developing a restaurant.

## **2.LITERATURE SURVEY**

### **2.1 Existing system**

In the existing system it is not easy to identify the review is correct or not. And also we want to ensure that the review is a valid statement.

It is very time consuming task that to validate the review is true.

### **2.2 Proposed system**

Our Proposed system can identify the review correct or not and validate the review is true without time consuming.

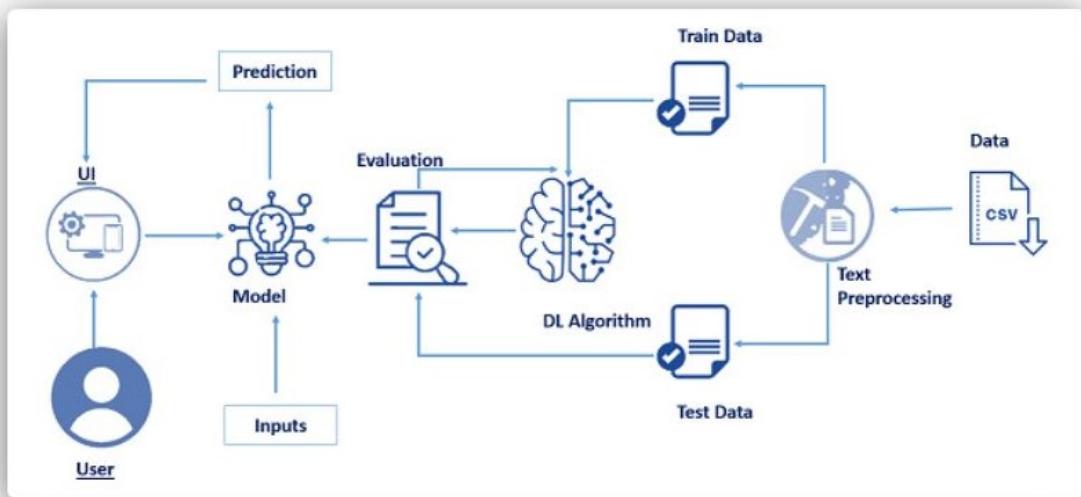
Using deep learning algorithms we have to check whether the review is positive or negative. And also it is easy to identify the review about the restuarant.

After we are going to save the model after training then we are going to develop a flask app where our app is going to do predict the review is good or bad by feedback given by the customer about the restaurant.

### 3.THEORETICAL ANALYSIS

#### 3.1 Block diagram

Technical Architecture:



#### 3.2 Software/ Hardware designing

##### Hardware Requirements:

Processor : Intel Core I3

RAM : 8.00 GB

Storage 250 GB OS :

Windows/Linux/MAC

##### Software Requirements:

Operating System: Windows 10 Home

Anaconda:Anaconda must be installed as it provides jupyter notebook and Spyder

IBM Academic initiative account is required to access IBM Services

**IBM Watson Studio** IBM Watson Studio helps data scientists and analysts prepare data and build models at scale across any cloud. **IBM Watson Machine Learning - IBM Watson Machine Learning** helps data scientists and developers accelerate AI and machine-learning deployment.

**IBM Cloud Object Storage** IBM Cloud Object Storage makes it possible

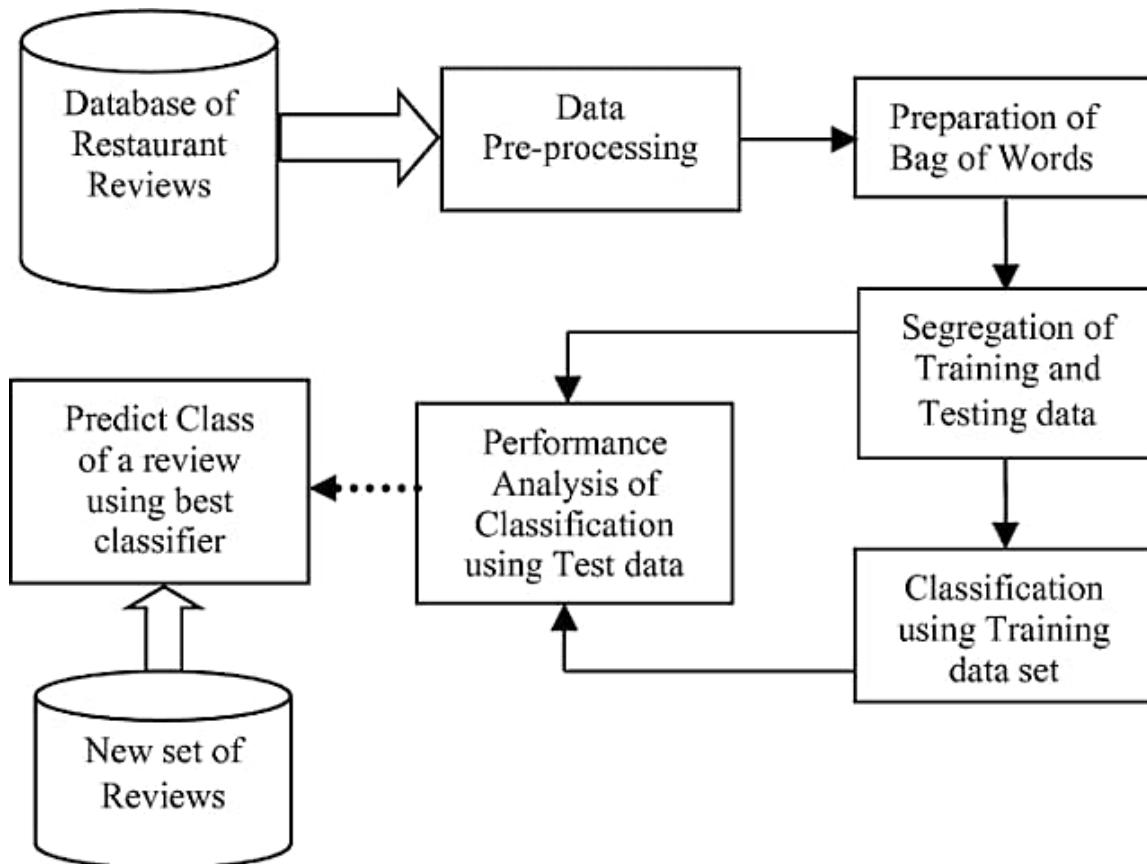
#### **4. EXPERIMENTAL INVESTIGATION**

We will be creating and testing our model for predicting if a review is negative or positive. since there are multiple algorithms, we can use to build our model, we will compare the accuracy scores after testing and pick the most accurate algorithm.

From the list, we are using Natural language processing we are going to use ANN model which gives the highest accuracy .

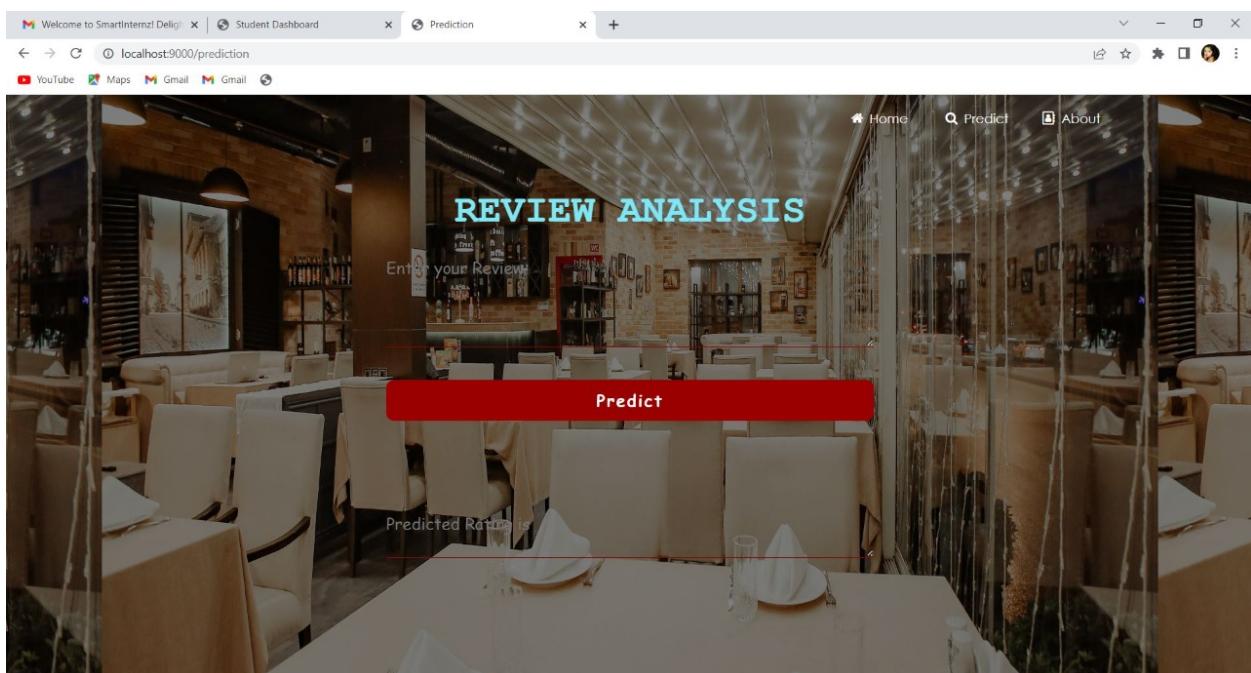
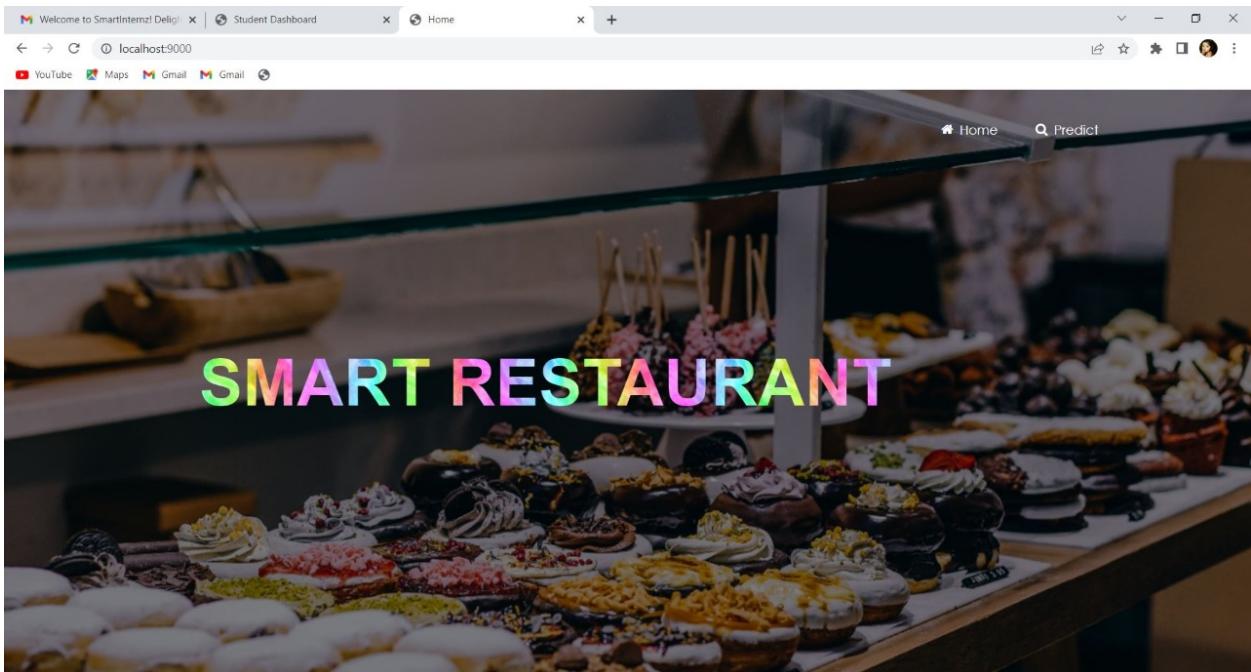
On the basis of results we have done the conclusion that our model works good.

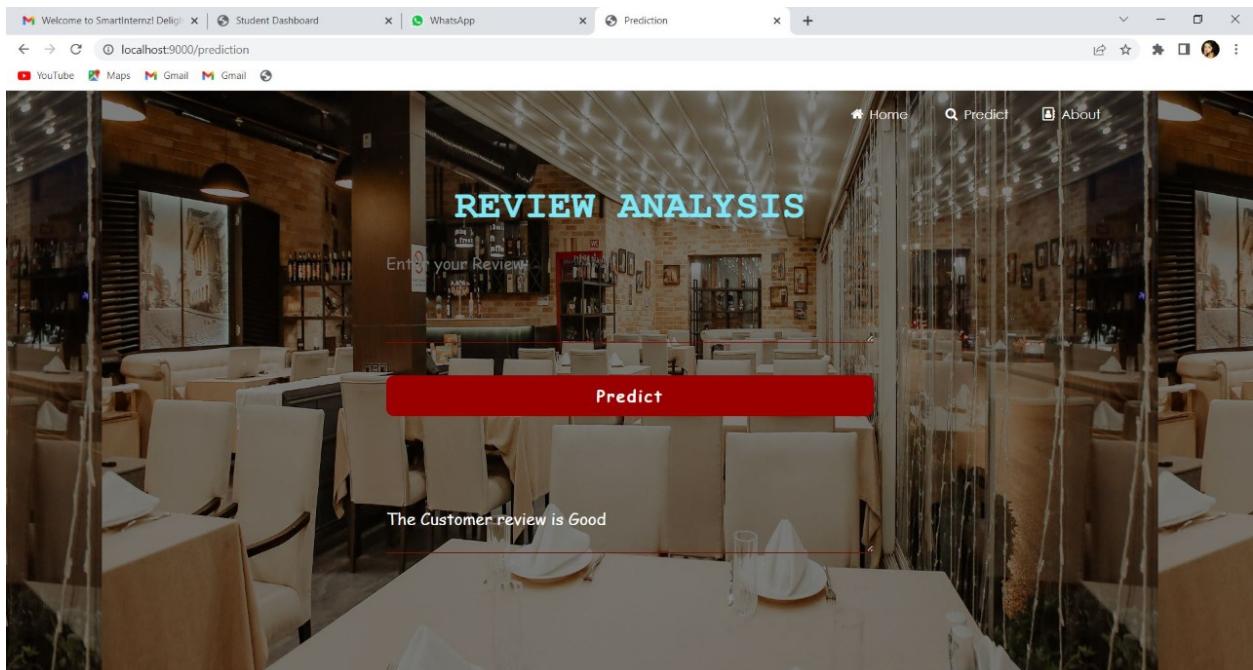
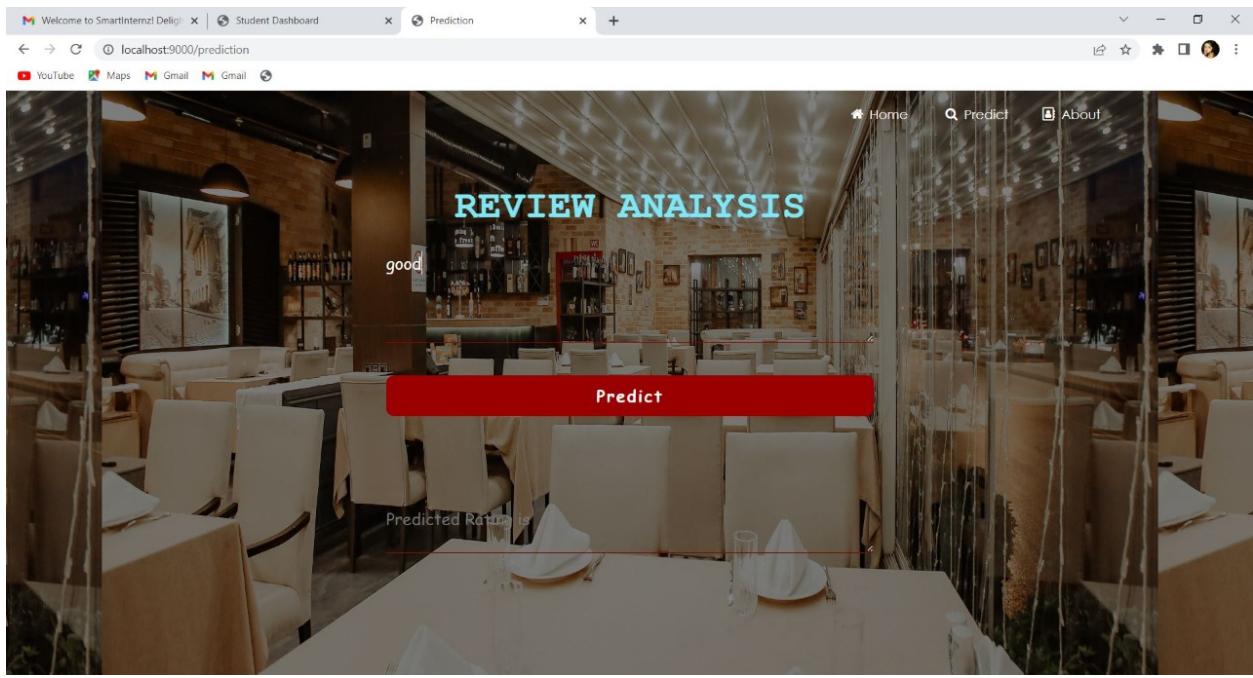
#### **5.FLOWCHART**

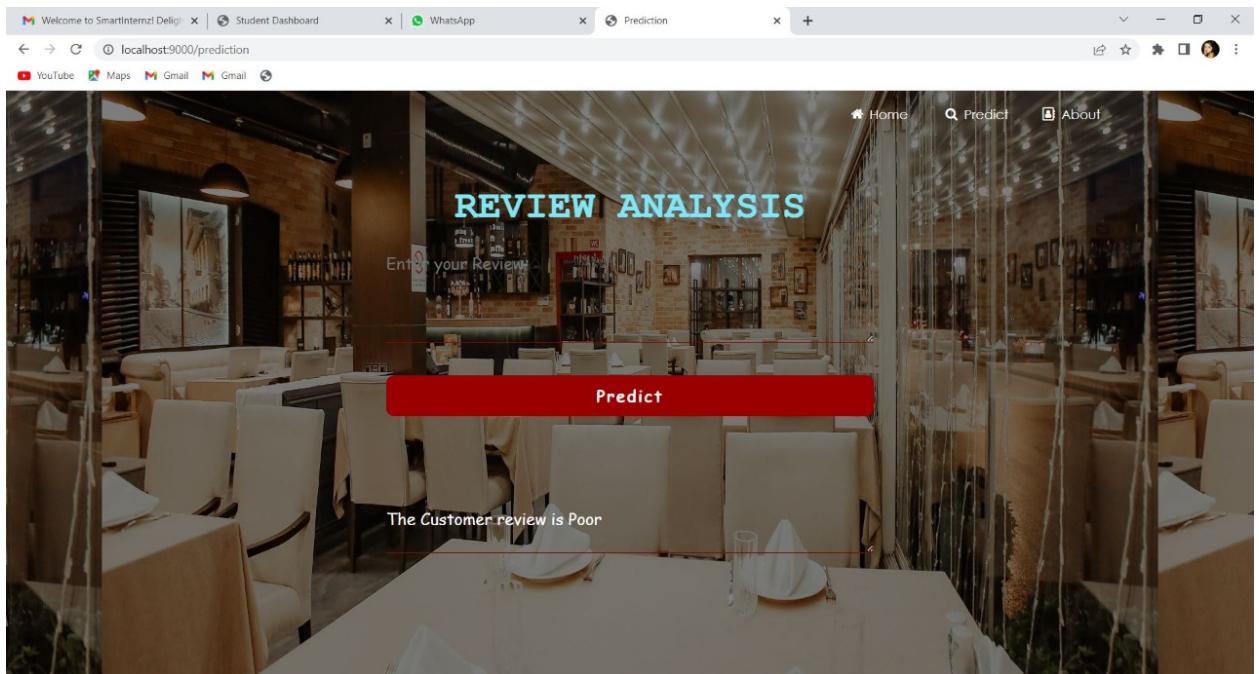
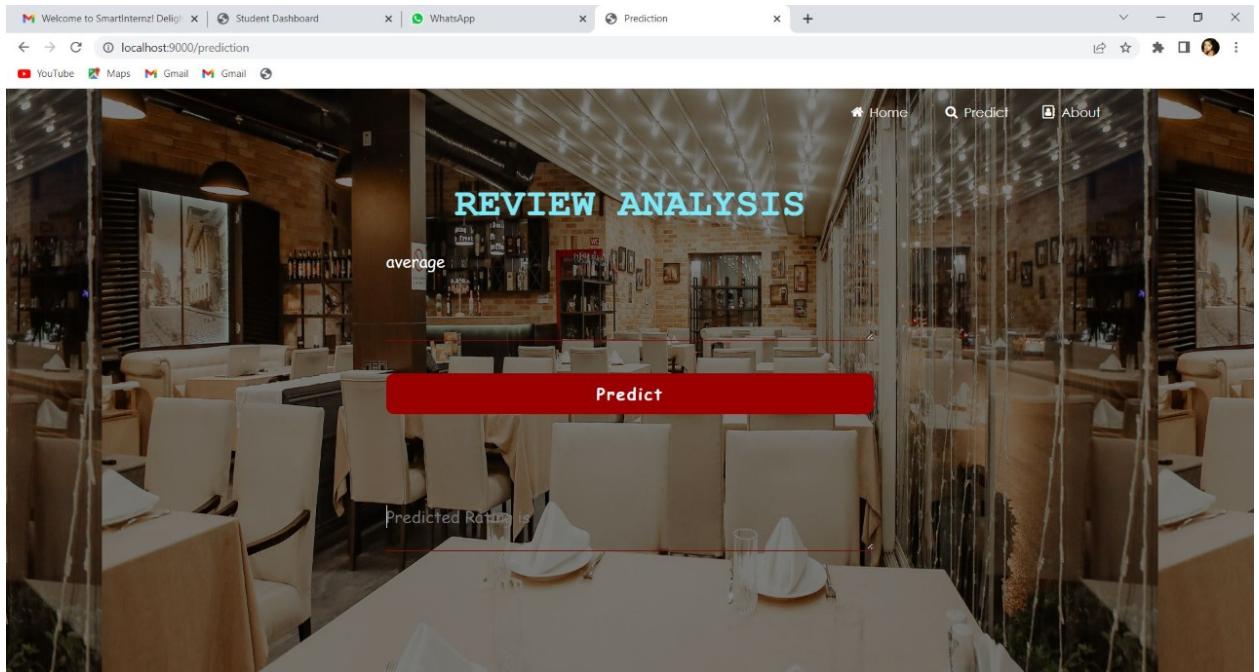


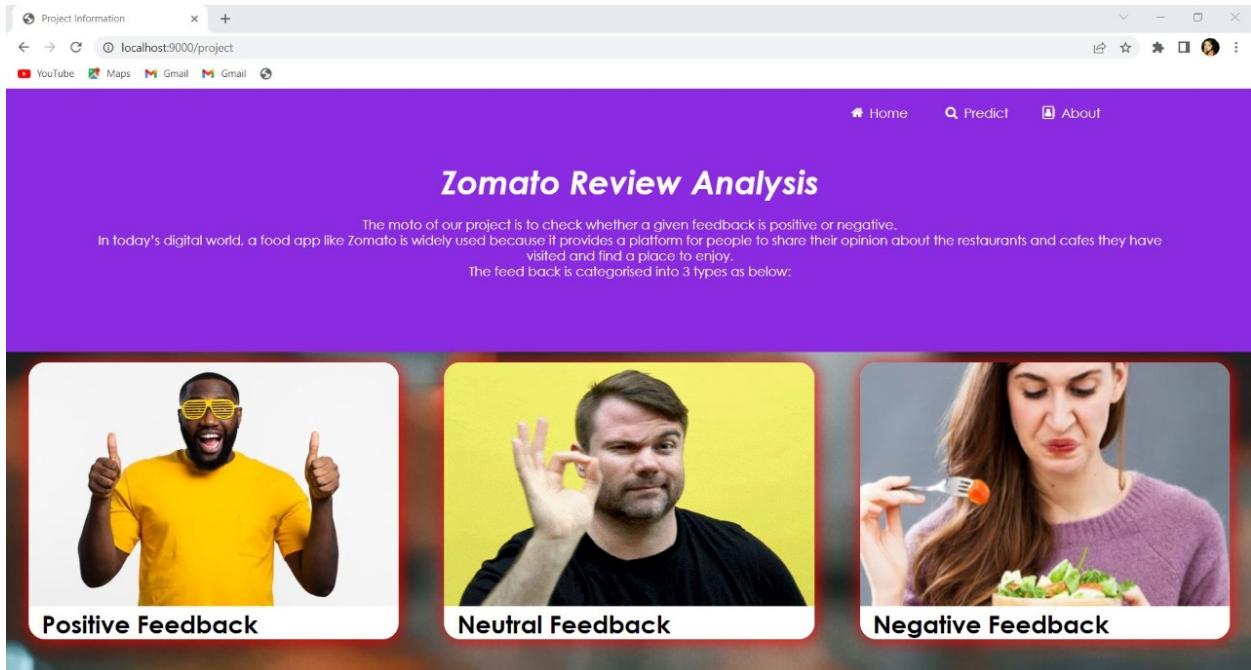
## 6.RESULT

The final result of our Smart Restaurant using NLP and ANN model predicts the review is positive or negative









## 1. ADVANTAGES AND DISADVANTAGES

### a. Advantages

The model is fast and accurate and it gives the exact reviews about the best restaurants . Its very time consuming because we use NLP and ANN model in this project. We get the output easily when we analyse so user can't wait much time for the output. It is very useful for a new customer who hasn't any idea about the restaurant.

### b. Disadvantages

A lack of negative review will affect the sustainability of restaurant . So it's a big disadvantage. We use deep learning to implement this system so its need a large amount of data to predict the restaurant reviews. The review is depend only on the interest of a customer so it may vary depend on the persons.

## **8.APPLICATIONS**

1. From Restaurant side, they can know the interest of the customers.
2. From Customers side, they can know the review as negative or positive.
3. Smart Restaurant is going to become more user friendly.

## **9.CONCLUSION**

Smart Restaurant going to be more innovative day by day by improving the customer reviews. Online reviews are a category of restaurant information created by users based on personal handling experience. Online food ordering websites endow with platforms for consumers to review restaurants and carve up opinions. The problem is most of the comments from customer reviews about the food items are contradicted to their ratings. Many customers will post their comments and forgot to rate the restaurant or not engrossed to rate it.

## **10. FUTURE SCOPE**

Deep learning and machine learning are the growing technologies so we can easily update the system in future. It is very useful for the new users so there is always a future scope for the system.

## **11. BIBLIOGRAPHY**

- 1)<https://machinelearningmastery.com/>
- 2)<http://www.kaggle.com/>
- 3)<https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>
- 4)<https://arxiv.org/>
- 5)[https://www.ibm.com/cloud/learn/natural-language-processing#:~:text=Natural%20language%20processing%20\(NLP\)%20refers,same%20way%20human%20beings%20can.](https://www.ibm.com/cloud/learn/natural-language-processing#:~:text=Natural%20language%20processing%20(NLP)%20refers,same%20way%20human%20beings%20can.)
- 6)<https://cloud.ibm.com/docs/create-deploy-machine-learning-mode>

## APPENDEX

### A.Source Code

```
import numpy as np
import pandas as pd
from flask import Flask, render_template, request
import tensorflow as tf
global graph
graph = tf.compat.v1.get_default_graph()
from tensorflow.keras.models import load_model
import pickle

with open(r'cv.pkl','rb') as file:
    cv=pickle.load(file)
import re
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

model = load_model("zomato_2_analysis-002.h5", compile = False)

app = Flask(__name__,template_folder="template")
```

```
@app.route('/')
def welcome():
    return render_template('home.html')

@app.route('/prediction', methods = ['GET','POST'])
def pred():
    if request.method == 'POST':
        review = request.form['message']
        review = re.sub('[^a-zA-Z]', ' ',review)
        review = review.lower()
        review = review.split()
        review = [ps.stem(word) for word in review if not word
                  in set(stopwords.words('english'))]
        review = ' '.join(review)
        review = cv.transform([review]).toarray()
        # with graph.as_default():
        y_p = model.predict(review)
        if y_p.argmax() == 0:
            output = "Average"
        elif y_p.argmax() == 1:
            output = "Good"
        else:
            output = "Poor"
    return render_template('prediction.html',prediction =
```

```
("The Customer review is " + output))

else:
    return render_template('prediction.html')

@app.route('/project')
def project():
    return render_template("project data.html")

if __name__ == '__main__':
    app.run(host = 'localhost',port =9000, debug = True , threaded = False)
```