**AI (ML DL) April15**

**Project**

**On**

**Zomato review analysis**

**Using**

**Natural Language Process**

**By Team 50**

**Team Members:**

1. B. R .L Prasanna
2. Shazmeen
3. P. Bala Bhavana
4. K. Sai Pujitha
5. A. Lakshmi Varshitha

1

**Table of Contents**

|  |  |
| --- | --- |
| 1. **INTRODUCTION** | |
| 1.1 Overview | |
| 1.2 Purpose | |
| 1. **LITERATURE SURVEY** | |
| 2.1 Existing problem | | |
| 2.2 Proposed solution | | |
| 1. **THEORITICAL ANALYSIS** | |
| * 1. Block diagram | |
| 3.2 Hardware / Software designing | | |
|  | | |
|  | |
| 1. **RESULT** |  |
| 1. **ADVANTAGES & DISADVANTAGES** | | |
| 1. **APPLICATIONS** | |
| 1. **CONCLUSION** | |
| 1. **FUTURE SCOPE** | |
| 1. **BIBILOGRAPHY** | |
| 1. **APPENDIX** | |
| A. Source code | |

1. Introduction
   1. **Overview:**

The basic idea of analysing the Zomato dataset is to get a fair idea about the factors affecting the establishment of different types of restaurant at different places in Bengaluru, aggregate rating of each restaurant, Bengaluru being one such city has more than 12,000 restaurants with restaurants serving dishes from all over the world. With each day new restaurants opening the industry hasn’t been saturated yet and the demand is increasing day by day. Inspite of increasing demand it however has become difficult for new restaurants to compete with established restaurants. Most of them serving the same food. Bengaluru being an IT capital of India . Most of the people here are dependent mainly on the restaurant food as they don’t have time to cook for themselves. With such an overwhelming demand of restaurants it has therefore become important to study the demography of a location.

* 1. **Purpose:**

The basic idea of analyzing the Zomato dataset is to get a fair idea about the factors affecting the aggregate rating of each restaurant, establishment of different types of restaurant at different places, Bengaluru being one such city has more than 12,000 restaurants with restaurants serving dishes from all over the world. With each day new restaurants opening the industry has'nt been saturated yet and the demand is increasing day by day. Inspite of increasing demand it however has become difficult for new restaurants to compete with established restaurants. Most of them serving the same food. Bengaluru being an IT capital of India. Most of the people here are dependent mainly on the restaurant food as they don't have time to cook for themselves. With such an overwhelming demand of restaurants it has therefore become important to study the demography of a location. What kind of a food is more popular in a locality. Do the entire locality loves vegetarian food. If yes then is that locality populated by a particular sect of people for eg. Jain, Marwaris, Gujaratis who are mostly vegetarian. These kind of analysis can be done using the data, by studying different factors.

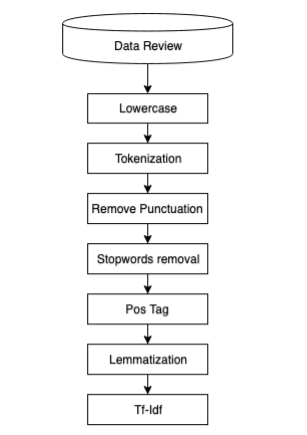
1. Literature survey
   1. **Existing Problem**

It happened to be so difficult to determine the reviews and analyse the customer reaction and there would be many fake reviews. That became a major problem of going down rating of zomato and for this we have used NLP to determine all the reviews and analyse correctly which helps in growth of business.

* 1. **Proposed Solution**

Extraction only the URL, name and address of the restaurant were extracted which were visible on the front page. The URl's for each of the restaurants on the zomato were recorded in the csv file so that later the data can be extracted individually for each restaurant. This made the extraction process easier and reduced the extra load on my machine. The recorded data for each restaurant and each category was read and data for each restaurant was scraped individually. 15 variables were scraped in this phase. For each of the neighbourhood and for each category their online order, book table, rate, votes, phone, location, rest type, dish liked, cuisines, approx. cost(for two people), reviews list, menu item was extracted. See section 5 for more details about the variables.

1. Theoretical Analysis
   1. **Block Diagram**



3.2 **Hardware / Software designing**

Python, Python Web Frame Works, Python for Data Analysis, Python For Data Visualization, Data Pre-processing Techniques, Artificial Intelligence, Natural process language.

1. Result

We have analysed the sentimental analysis on Zomato review data used Artificial intelligence to predict the reviews of Zomato. We have used Natural language process and its libraries to make predictions and compared their performance. NLP randomly cleans the data sampled from the training dataset, this will make our model more robust.

1. Advantages and Disadvantages

Advantages:

* Relieves burden of learning syntax
* No training

Disadvantages:

* Requires clarification on dialogue
* May require more keystrokes
* May not show context
* Is unpredictable

1. Applications
   * Can analyse the sentimental analysis of the Zomato review dataset.
   * Implementable on the website.
2. Conclusion

We purpose a method to analyse user’s sentiment of Zomato Restaurants and focusing review in Bangalore for study case. We are using Natural process language to classify the sentiments of users based on their review. We also find words that affect the classifier model.

1. Future Scope

This model can predict the outcome with many different inputs within seconds. The model will save a lot of time of the owners. Experiment cost is also reduced with creates a bigger opportunity for all vendors.

1. Bibliography

**Data repositories**

Kaggle.com

**Algorithms**

Thesmartbridgeteachable.com

1. Appendix

**Source code**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

mydataset = pd.read\_csv('zomato.csv')

mydataset

mydataset.drop(["url","address","online\_order","book\_table","votes","phone","location","rest\_type","dish\_liked","cuisines","approx\_cost(for two people)","menu\_item","listed\_in(type)","rate","name","listed\_in(city)\

"], axis=1, inplace=True)

mydataset

for i in range(0,51717):

if(mydataset['reviews\_list'][i]=='[]'):

mydataset.drop(i, inplace=True)

mydataset.reset\_index(inplace=True)

mydataset.drop("index", inplace=True, axis=1)

review=[]

for i in range(0, 44122):

r=[]

r1=mydataset['reviews\_list'][i]

r=r1.split(")")

del r[-1]

review.extend(r)

import re

for i in range(len(review)):

s=review[i]

m=re.search("RATED",s)

if(m):

s=s[(m.end()+4):]

s=re.sub("(n\'t)|(\w+nt)"," not",s)

s=re.sub("(\'ve)"," have",s)

s=re.sub("(\'s)"," is",s)

s=re.sub("(\'m)"," am",s)

s=re.sub("\\\\n"," ",s)

s=re.sub(".\\\\."," ",s)

s=re.sub("\\\\."," ",s)

s=re.sub("\\\\x"," ",s)

s=re.sub("x8"," ",s)

s=re.sub('[^a-zA-Z]',' ',s)

review[i]=s

while '' in review:

review.remove('')

from pandas import DataFrame

df = DataFrame (review,columns=['reviews'])

import nltk

nltk.download('vader\_lexicon')

from nltk.sentiment.vader import SentimentIntensityAnalyzer

sia = SentimentIntensityAnalyzer()

df['scores']=df['reviews'].apply(lambda reviews : sia.polarity\_scores(reviews))

df['compound']=df['scores'].apply(lambda score\_dict : score\_dict['compound'])

lbl=[]

for i in range(1533313):

l=df.loc[i]['scores']

del l['compound']

lbl.extend(l)

l2=df.loc[1533312]['scores']

l2

m=max(l2,key=l2.get)

print(m)

df['label']=df['scores'].apply(lambda c : max(c,key=c.get))

mydataset2=df

df2=mydataset2.drop(["scores","compound"],axis=1)

df2.to\_csv("zomato1533313\_dataset")

sampleData=df2[0:50000]

sampleData

sampleData['label'].value\_counts()

sampleData.to\_csv("zomato50000\_dataset.csv")

import re

import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer

c=[]

for i in range(0,50000):

review = re.sub('[^a-zA-Z]', ' ', sampleData['reviews'][i])

review=review.lower()

review=review.split()

review = [word for word in review if not word in set(stopwords.words('english'))]

#applystemming concept

ps = PorterStemmer()

review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]

# we use same line as there may be word left which are stopword

review=' '.join(review)

c.append(review)

labels=[]

for i in range(0,50000):

review = re.sub('[^a-zA-Z]', ' ', sampleData['label'][i])

labels.append(review)

import csv

with open('zomato50000\_reviews.csv', 'w') as myfile:

wr = csv.writer(myfile, quoting=csv.QUOTE\_ALL)

wr.writerow(c)

import csv

with open('zomato50000\_labels.csv', 'w') as myfile:

wr = csv.writer(myfile, quoting=csv.QUOTE\_ALL)

wr.writerow(labels)

import csv

with open('zomato50000\_dataset.csv', 'w') as myfile:

wr = csv.writer(myfile, quoting=csv.QUOTE\_ALL)

wr.writerow(sampleData)

from sklearn.feature\_extraction.text import CountVectorizer

cv = CountVectorizer(max\_features=1500)

x=cv.fit\_transform(c).toarray()

y=sampleData.iloc[:,-1].values

from sklearn.preprocessing import LabelEncoder

lb=LabelEncoder()

y=lb.fit\_transform(y)

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.2, random\_state=0)

import keras

from keras.models import Sequential

from keras.layers import Dense

model = Sequential()

model.add(Dense(input\_dim=1500, init="random\_uniform", activation='sigmoid',output\_dim=3000))

model.add(Dense(output\_dim=3000, init="random\_uniform", activation='sigmoid'))

model.add(Dense(output\_dim=3, init="random\_uniform", activation='softmax'))

model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

model.fit(x\_train,y\_train, epochs=10, batch\_size=10)

y\_pred=model.predict(x\_test)

y\_pred=y\_pred.all()

model.save("NLPmodel.h5")

import pickle

filename = 'nlp\_zomato\_model.pkl'

pickle.dump(model, open(filename, 'wb'))

pickle.dump(cv, open('cv\_tranform.pkl', 'wb'))

s= " It is amazing. never had this type of food"

s=re.sub("(n\'t)|(\w+nt)"," not",s)

s=re.sub("(\'ve)"," have",s)

s=re.sub("(\'s)"," is",s)

s=re.sub("(\'m)"," am",s)

s=re.sub("\\\\n"," ",s)

s=re.sub(".\\\\."," ",s)

s=re.sub("\\\\."," ",s)

s=re.sub("\\\\x"," ",s)

s=re.sub("x8"," ",s)

s=re.sub('[^a-zA-Z]',' ',s)

s=s.lower()

s=s.split()

s = [word for word in s if not word in set(stopwords.words('english'))]

ps = PorterStemmer()

s = [ps.stem(word) for word in s if not word in set(stopwords.words('english'))]

s=' '.join(s)

t=[]

t.append(s)

x1=cv.fit\_transform(t).toarray()

x1

from keras.preprocessing import sequence

x1 = sequence.pad\_sequences(x1, maxlen=1500)

c1=model.predict\_classes(x1)

0\*\* ==> \*\*Negative

1\*\* ==> \*\*Positive

2\*\* ==> \*\*Neutral

lb.inverse\_transform(c1)

y = model.predict(cv.transform([c1]))