PROJECT REPORT – SENTIMENT ANALYSIS OF CUSTOMER FEEDBACK ON RESTAURANTS

Shyamkrishnan Sudhir
19BAI10032

- **INTRODUCTION**

.1 OVERVIEW

"What others think? " is always important information in a decision-making process. Every day people discuss various products on social media sites. Web and its associated distribution services provides information services such as online services where data objects are linked together to facilitate interactive access.

Companies want a piece of that pie to determine how their audience communicates to find the important information that drives business. Sentiment analysis is the robotic mining of opinions and feelings from content through Natural Language Processing (NLP). Sentiment analysis is nothing but categorizing opinions in the given content or documents into "positive" or "negative|| or "neutral".

.2 PURPOSE

To get the detail opinion or sentiments, we have to process the document to aspect level. Aspect level sentiment analysis is to determine the features of the sentiment conveyed towards each aspect and the given target entities. This proposed model handles sentiment polarity classification which is a fundamental problem of sentiment classification. The online

data have several drawbacks to hinder the sentiment analysis task.

The first defect is anybody can post their own contents and impact of their comment is not assured. Online spammers may post their fake opinions. The next fault is the polarity of reviews cannot be ascertained or unavailable. The dataset used in this paper is around 40000 food reviews collected from zomato. Each post in zomato is inspected and verified by the company before it gets posted. Each review has a rating scale from 1 to 5 stars which can be used to identify the sentiment polarity.

- **LITERATURE SURVEY**

.1 EXISTING PROBLEM

Traditional approaches on sentiment analysis use word count or frequencies in the text which are assigned sentiment value by expert. These approaches disregard the order of words. Scientists are actively researching sentiment analysis that has become the biggest area of research in the last few years. Sultana, Kumar [1] described that sentimental analysis has three important aspects, positive, negative, and neutral.

From last few years, the world wide web becomes a key factor of customers' reviews, by the social media and e-commerce websites, such as Facebook, tweeters user can share their reviews and these reviews can be good or bad, and these reviews help in making choices about applying new plan and decisions about products.

Chen, Xue [2] introduced a new technique to remove the traits of sentiment analysis for the reviews of products. The most common TF-IDF vectors can obtain by using the same form of synonyms by viewing the products' reviews, we can categorize the sequences of feature vectors along with clustering

algorithms. By applying this technique we can refine span algorithms for pseudo consecutive phrases with FPCD having word order details. By using the last steps, the text feature is gathered. As a result of applying the different mechanisms of performance can be enhanced.

In Abbas, Memon [3], the authors introduced a new heuristic method along with naïve bias for specified issues. An MNB is an NB classifier used for text categorization and implemented for sentimental analysis.

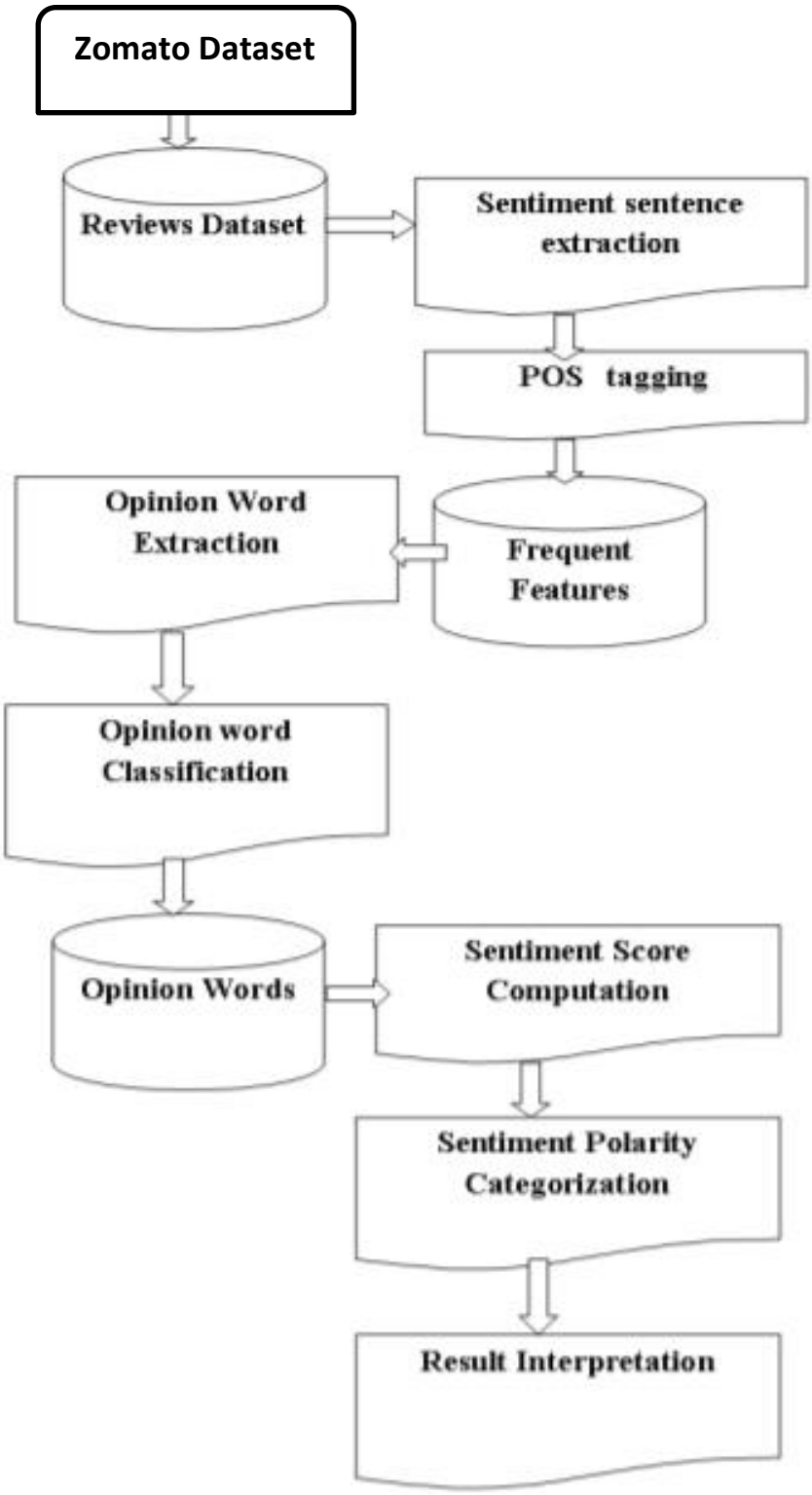
.2 PROPOSED SOLUTION

POS technique is implemented to each and every sentence level and the results are shown in charts. We used supervised and rule based techniques to mine the opinions from online product reviews. The approach which is used in this paper has five stages.

These stages are acquiring dataset through various visualization, data preprocessing, extraction of features implementing the machine learning classifiers through Keras and Anaconda, and lastly, evaluated models through train test split by using different metrics of binary classification.

- **THEORITICAL ANALYSIS**

.1 BLOCK DIAGRAM



.2 HARDWARE/SOFTWARE DESIGNING

Hardware Requirements:

- Core i3/i5 processor
- At least 4/8 GB RAM
- At least 60 GB of Usable Hard Disk Space

Software Requirements:

- Python 3.x
- Anaconda Distribution
- NLTK Toolkit
- UNIX/LINUX/Windows Operating System.
- Flask/Jinja Template/VS Code(or any IDE)

• EXPERIMENTAL INVESTIGATIONS

- **Data Collection:**

  Data which means product reviews collected from zomato.com. Each review includes the following information: 1) reviewer ID; 2) product ID; 3) rating; 4) time of the review; 5) helpfulness; 6) review text. Every rating is based on a 5-star scale, resulting all the ratings to be ranged from 1-star to 5-star with no existence of a half-star or a quarter-star.

- **Sentiment Sentence Extraction & POS Tagging:**

  Tokenization of reviews after removal of STOP words which mean nothing related to sentiment is the basic requirement for POS tagging. After proper removal of STOP words like “am, is, are, the, but” and so on the remaining sentences are converted in tokens.

These tokens take part in POS tagging 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:

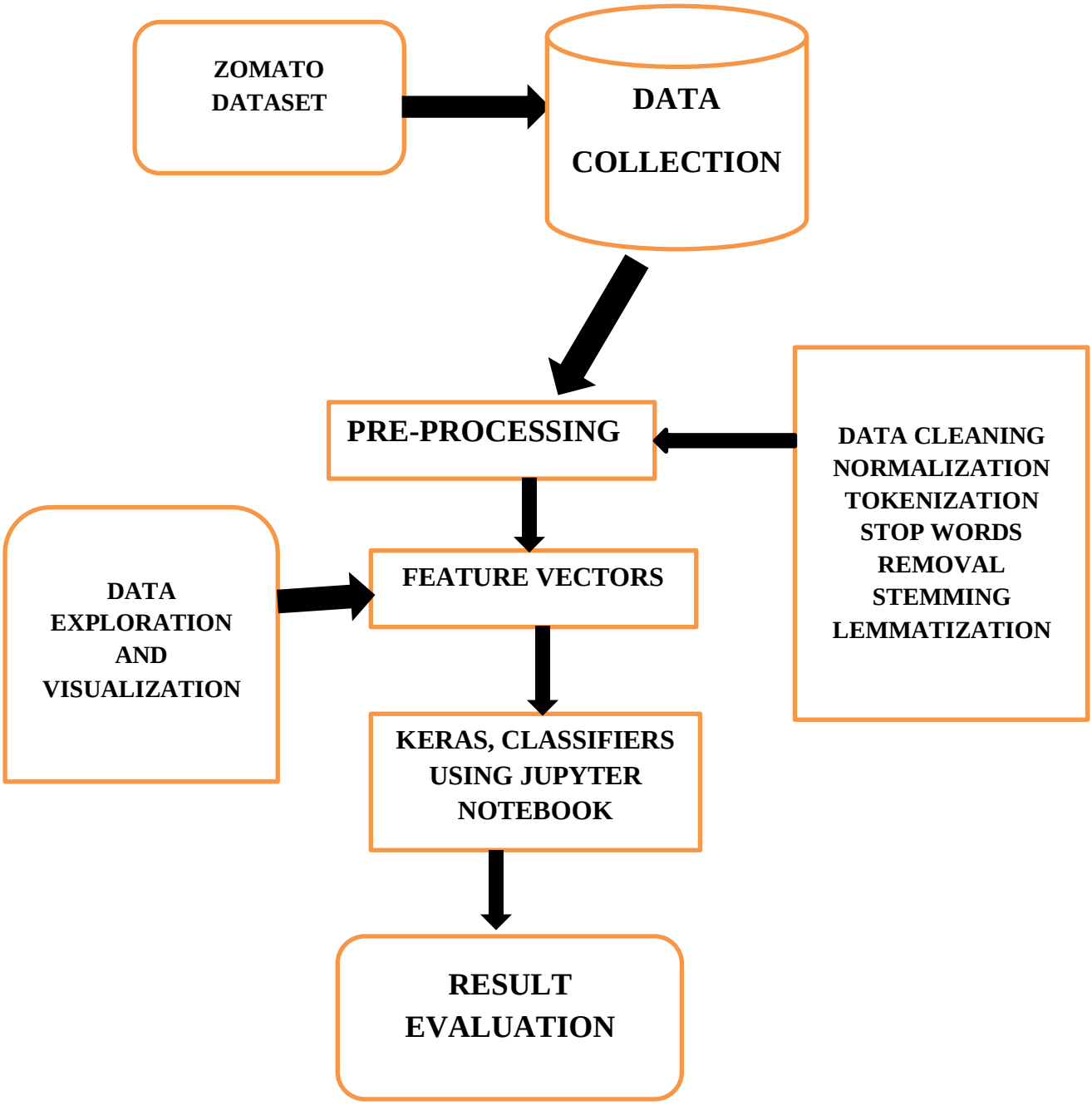
1) Words like nouns and pronouns usually do not contain any sentiment. It is able to filter out such words with the help of a POS tagger;
2) A POS tagger can also be used to distinguish words that can be used in different parts of speech.

- **Negative Phrase Identification:**

  Words such as adjectives and verbs are able to convey opposite sentiment with the help of negative prefixes. For instance, consider the following sentence that was found in an electronic device's review: "The built in speaker also has its uses but so far nothing revolutionary." The word, "revolutionary" is a positive word according to the list in. However, the phrase "nothing revolutionary" gives more or less negative feelings.

  Therefore, it is crucial to identify such phrases. In this work, there are two types of phrases have been identified, namely negation-of-adjective (NOA) and negation-of-verb (NOV).

- **FLOWCHART**


ZOMATO DATASET
DATA COLLECTION
PRE-PROCESSING
DATA CLEANING
NORMALIZATION
TOKENIZATION
STOP WORDS REMOVAL
STEMMING
LEMMATIZATION
DATA EXPLORATION AND VISUALIZATION
FEATURE VECTORS
KERAS, CLASSIFIERS USING JUPYTER NOTEBOOK
RESULT EVALUATION

- **RESULT**

  - The ultimate outcome of this Training of Public reviews dataset is that, the machine is capable of judging whether an entered sentence bears positive response or negative response. Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while Recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Both precision and recall are therefore based on an understanding and measure of relevance.

  - F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

  - In statistics, a receiver operating characteristic curve, i.e. ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The Total Operating Characteristic (TOC) expands on the idea of ROC by showing the total information in the two-by-two contingency table for each threshold. ROC gives only two bits of relative information for each threshold, thus the TOC gives strictly more information than the ROC. 21 True Negative False Positive False Negative True Positive When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative').

- The machine evaluates the accuracy of training the data along with precision Recall and F1 The Confusion matrix of evaluation is calculated. It is thus capable of judging an externally written review as positive or negative. A positive review will be marked as [1], and a negative review will be hence marked as [0]. Results obtained using Hold-out Strategy(Train-Test split) [values rounded upto 2 decimal places].

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- By using sentiment analysis, you gauge how customers feel about different areas of your business without having to read thousands of customer comments at once.

- If you have thousands of feedback per month, it is impossible for one person to read all of these responses. By using sentiment analysis and automating this process, you can easily drill down into different customer segments of your business and get a better understanding of sentiment in these segments.

- Dictionary is not required. Exhibit the high precision of classification.

DISADVANTAGES

- While sentiment analysis is useful, it is not a complete replacement for reading survey responses. Often, there are

useful nuances in the comments themselves. Where sentiment analysis can help you further is by identifying which of these comments you should read.

- Classifier trained on the textual data in a single field much of the time doesn't work with different fields.

- **APPLICATIONS**

Product analysis

- Find out what the public is saying about a new product right after launch, or analyze years of feedback you may have never seen. You can search keywords for a particular product feature (interface, UX, functionality) and use aspect-based sentiment analysis to find only the information you need.

- Discover how a product is perceived by your target audience, which elements of your product need to be improved, and know what will make your most valuable customers happy. All with sentiment analysis.

Market and competitor research

- Use sentiment analysis for market and competitor research. Find out who’s receiving positive mentions among your competitors, and how your marketing efforts compare.
- Analyze the positive language your competitors are using to speak to their customers and weave some of this language into your own brand messaging and tone of voice guide.

- ## CONCLUSION

Sentiment analysis is the process of identifying the feeling expressed in the text or document. We proposed a methodology for mining the food reviews based on score combined with existing text analysing packages. The proposed system has produced a very good result using the score ratings.

The limitation of this system is, it works better only for the open sentiments like rating or scores. The results were not promising for hidden sentiments. The algorithms like Naïve Bayes, LinearSVC, and logistic regression were applied. By performing the analysis it shows that the linear support vector classifier works well than other classifiers.

- ## FUTURE SCOPE

- In Future work, prediction based methods will be implemented with existing approach. More features will be extracted to handle the implicit sentiment analysis.

- Sentiment analysis is an emerging research area in text mining and computational linguistics, and has attracted considerable research attention in the past few years. Future research shall explore sophisticated methods for opinion and product feature extraction, as well as new classification models that can address the ordered labels property in rating inference. Applications that utilize results from sentiment analysis is also expected to emerge in the near future.

- In the future, to improve the performance of the classifier different features set will be considered such as bi-gram, tri-gram, and four-gram.

11. BIBLIOGRAPHY

1.Sultana, N., et al., Sentiment Analysis for product review. 2019. 9(3).

2.Chen, X., et al., A novel feature extraction methodology for sentiment analysis of product reviews. 2019. 31(10): p. 6625-6642. 13.Abbas, M., et al., Multinomial Naive Bayes classification model for sentiment analysis. 2019. 19(3): p. 62.

3.Neethu, M. and R. Rajasree. Sentiment analysis in twitter using machine learning techniques. in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT). 2013. IEEE.

4. Hamzah, A.A., M.F.J.J.o.U.S.S. Shamsudin, and Technology, Why customer satisfaction is important to business? Journal of Undergraduate Social Science and Technology, 2020. 1(1).

5. Alam, T.M. and M.J. Awan, Domain analysis of information extraction techniques. International Journal of Multidisciplinary Sciences and Engineering, 2018. 9: p. 1-9.

12.APPENDIX

WEB.PY PYTHON FILE

```
import graphlib
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 keras.models import load_model
import pickle
import re
import nltk
nltk.download("stopwords")
```

```
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

with open(r'cv.pkl','rb') as file:
    cv = pickle.load(file)

model = load_model("zomato_2_analysis.h5",compile=False)
app = Flask(__name__,template_folder="templates")

@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()
        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')

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

SENTIMENT ANALYSIS MODEL IPYNB FILE

```
import numpy as np
import pandas as pd
dataset = pd.read_csv("zomato.csv")
data_review = dataset['reviews_list']
```

```
x = []
y = []
for row_num in range(51717):
    lst = data_review[row_num].split("('")
    for i in lst:
        if len(i) > 5:
            if i.find("',") != -1:
                single_rev = i.split("',")
                if len(single_rev[0]) > 2:
                    x.append(single_rev[0])
                if len(single_rev[1]) > 2:
                    y.append(single_rev[1])

import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

rating_final = []
review_final = []

import re
for loop in range(40000):
    data_x = x[loop]
    data_x = re.sub('[a-zA-Z]'," ",data_x)
    data_x = data_x.split()
    data_x = ''.join(data_x)
    data_x = float(data_x)
    if data_x < 2.5:
        rating_final.append("poor")
    elif data_x >= 2.5 and data_x <= 3.5:
        rating_final.append("average")
    elif data_x > 3.5:
        rating_final.append("good")

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
rating_final = le.fit_transform(rating_final)
rating_final = np.array(rating_final)
rating_final = np.expand_dims(rating_final, axis=1)
from sklearn.preprocessing import OneHotEncoder
one = OneHotEncoder()
rates = one.fit_transform(rating_final).toarray()

for loop in range(40000):
    data_y = y[loop]
    data_y = re.sub('[^a-zA-Z]'," ",data_y)
```

```
    data_y = data_y.lower()
    data_y = data_y.split()
    data_y = [ps.stem(word) for word in data_y if not word in
set(stopwords.words('english'))]
    data_y = ' '.join(data_y)
    review_final.append(data_y)

from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 20000)
x_final = cv.fit_transform(review_final).toarray()

import pickle
pickle.dump(cv, open('cv.pkl','wb'))

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_final, rates, test_size
= 0.2, random_state = 0)

from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense
model = Sequential()
model.add(Dense(units = 13264, kernel_initializer = 'random_uniform',
activation = 'relu'))
model.add(Dense(units = 2000, kernel_initializer = 'random_uniform',
activation = 'relu'))
model.add(Dense(units = 2000, kernel_initializer = 'random_uniform',
activation = 'relu'))
model.add(Dense(units = 2000, kernel_initializer = 'random_uniform',
activation = 'relu'))
model.add(Dense(units = 3, kernel_initializer = 'random_uniform', activation =
'softmax'))
model.compile(optimizer = 'adam', loss='categorical_crossentropy',
metrics=["mae", "acc"])
model.fit(x_train, y_train, batch_size = 128, epochs = 5)

y_pred = model.predict(x_test)
text = "the food is okay. average place "
text = re.sub('[^a-zA-Z]',' ',text)
text = text.lower()
text = text.split()
text = [ps.stem(word) for word in text if not word in
set(stopwords.words('english'))]
text = ' '.join(text)

y_p = model.predict(cv.transform([text]))
model.save("zomato_2_analysis.h5")
```