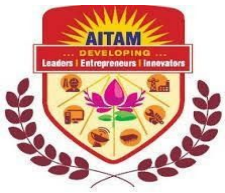
Food Review Analysis (using NLP)

A Mini Project report submitted in partial fulfillment of the requirements or the Award of Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
By

T.Aravind	**18A51A0502**
B.Rohith	**18A51A0504**
B.Pavan sai	**18A51A0505**
D.Sampreeth	**18A51A0511**
L.Vamsi	**18A51A0528**

Under the esteemed guidance of
Dr.R. Srinivas
Professor
Department of CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(Approved by AICTE, Permanently Affiliated to JNTU Kakinada, Accredited by NBA & NAAC With A+ Recognized by UGC under Section 2(f) & 12(B))
TEKKALI, SRIKAKULAM,ANDHRA PRADESH

2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT
TEKKALI

CERTIFICATE

This is to certify that the mini project work entitled "**Food Review Analysis**", is a bonafide work done by **L.Vamsi (18A51A0528)** and submitted in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING.**

Dr. G. S. N. Murty,
(M.Tech., Ph. D)
Head of the Dept.
Department of CSE

Dr. R. Srinivas,
(M. Tech, M.C.A)
Associate Professor.
Department of CSE.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT
TEKKALI

CERTIFICATE

This is to certify that the mini project work entitled "**Food Review Analysis**", is a bonafide work done by **T. Aravind (18A51A0502), B.Rohith (18A51A0504), B.Pavan sai (18A51A0505), D.Sampreeth (18A51A0511), L.Vamsi (18A51A0528)** and submitted in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING.**

Dr. G. S. N. Murty,
(M.Tech., Ph. D)
Head of the Dept.
Department of CSE

Dr. R. Srinivas,
(M. Tech, M.C.A)
Associate Professor.
Department of CSE.

ACKNOWLEDGEMENT

It is indeed with a great sense of pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to **Director Prof.V.V. Nageswara Rao** and **Principal A.S. Srinivasa Rao**, for the facilities provided to accomplish this mini project.

We would like to thank our head of the department **Dr. G. S. N. Murty**, for his constructive criticism throughout the project.

We feel elated in manifesting our sense of gratitude to our guide **Dr. R. Srinivas**, for his valuable guidance .He has been a constant source of inspiration for us and we are very deeply thankful to him for his support and invaluable advice.

We are extremely grateful to my department staff members, lab technicians and non-teaching staff members for their extreme help throughout our project.

Finally, we express our heartfelt thanks to all my friends who helped in successful completion of this project.

T.Aravind
B.Rohith
B.Pavan sai
D.Sampreeth
L.Vamsi

DEPARTMENT OF CSE

Vision of the Department:

To become a pioneer in providing high quality education and research in the area of computer science and engineering.

Mission of the Department:

M1: Enrich society and advance computer science and engineering by preparing graduates with the knowledge, ability, and skill to become innovators and leaders who are able to contribute for the aspirations of the country and society.

M2: Benefit humanity through research, creativity, problem solving, and application development.

M3: Share knowledge and expertise to benefit the country, the region, and beyond while inspiring people to engage in computing fields.

The **Programme Educational Objectives (PEOs)** for our Computer Science and Engineering program are to produce graduates who will:

PEO1: Be employed as a practicing engineer in fields such as design, development, testing and research or undertake higher studies.

PEO2: Engage in lifelong self-directed learning, a capacity that is vital for success in today's global and rapidly changing engineering environment.

PEO3: Create new methods / processes to meet the society's needs with their knowledge.

PEO4: Conduct themselves as ethical and responsible professionals with good communication skills and demonstrate leadership skills.

PROGRAM OUTCOMES (POs):

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/Development Of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct Investigations Of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and ap12.5

12. Life-Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs):

By the completion of Computer Science program the student will be able to:
PSO1: Apply mathematical foundations, algorithmic principles, and theoretical computer science in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

PSO2: Demonstrate understanding of the principles and working of the hardware and software aspects of computer systems.

PSO3: Use knowledge in various domains to identify research gaps and hence to provide solutions to new ideas and innovations.

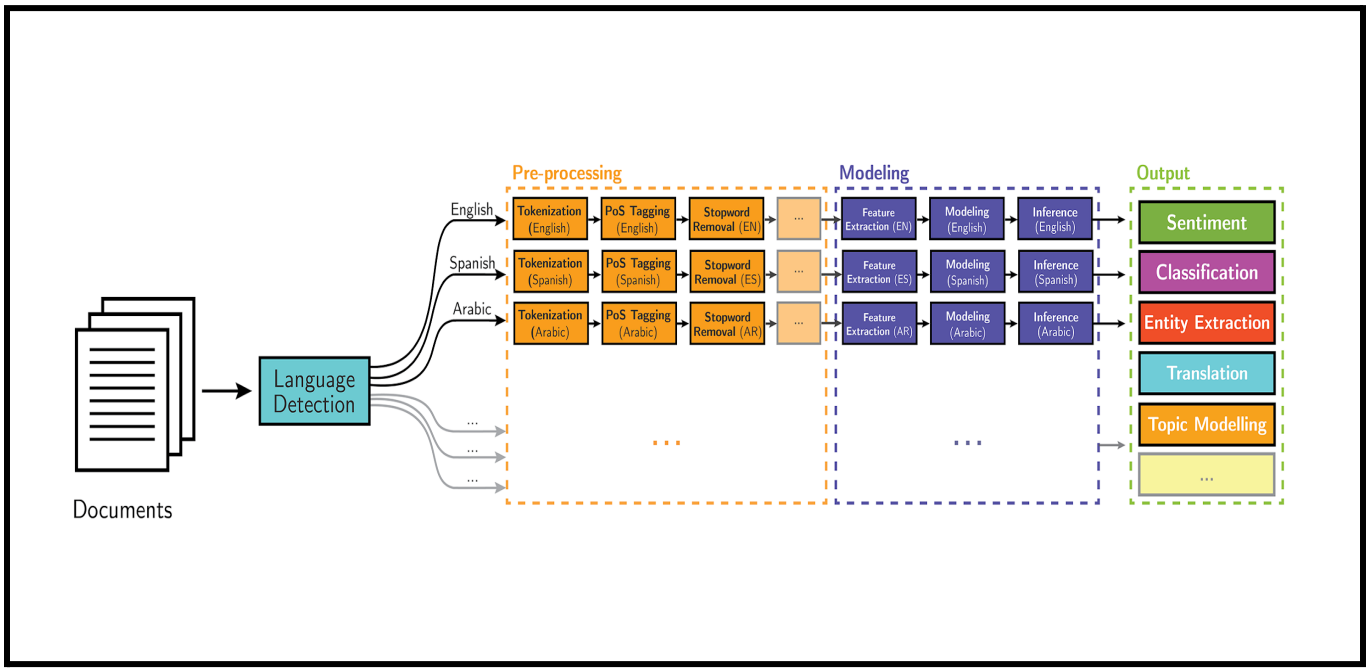
Abstract

There has been a rise in accessibility of online applications and a surge across social platforms for opinion sharing, online review websites, and personal blogs, which have captured the attention of stakeholders such as customers, organizations, and governments to analyze and explore these opinions. There exists some research to predict the consumer reviews then automatically rating individual food items of a restaurant based on online customer reviews and categorize it as positive, negative, or neutral. Opinion Mining or Sentiment Analysis can be defined as the task of detecting, extracting and classifying opinions on something. It is a type of processing of the natural language (NLP). It involves a way of development for the collection and examination of comments and opinions on food reviews. The process of information extraction is very important because it is a very useful technique but also a challenging task. The existing techniques for sentiment analysis include machine learning (supervised), and lexical-based approaches. Hence, the main aim of this project is to predict the rating based on review of the food.

INDEX

1.INTRODUCTION

Natural language processing (NLP) is an area of computer science and artificial intelligence. It is the intersection of computer science, linguistics and machine learning. The field focuses on communication between computers and humans in natural language and NLP is all about making computers understand and generate human language. Applications of NLP techniques include voice assistants like Amazon's Alexa and Apple's Siri, but also things like machine translation and text-filtering. Our goal is to predict whether they are positive or negative to achieve this, we use binary classification .



2. TECHNICAL REQUIREMENTS

2.1 TECHNICAL DESCRIPTION:

Tools:

Software Components:

- Jupyter notebook
- Visual studios
- Web browser
- Mongodb

Hardware Components:

- Computer

Technologies:

- Python, mongodb, javascript, html, css.
- Python packages: sklearn, numpy, pandas, flask, pickle, jinja.

Numpy: Numpy is a general-purpose array-processing package. It provides a high multidimensional array object, and tools for working with these arrays.

Pickle: Pickle Module is used for serializing and de-serializing a Python object structure by pickling being the way to convert it into a character stream for it to be saved on disk. Any object in Python can be pickled so that it can be saved on disk.

Sklearn:Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

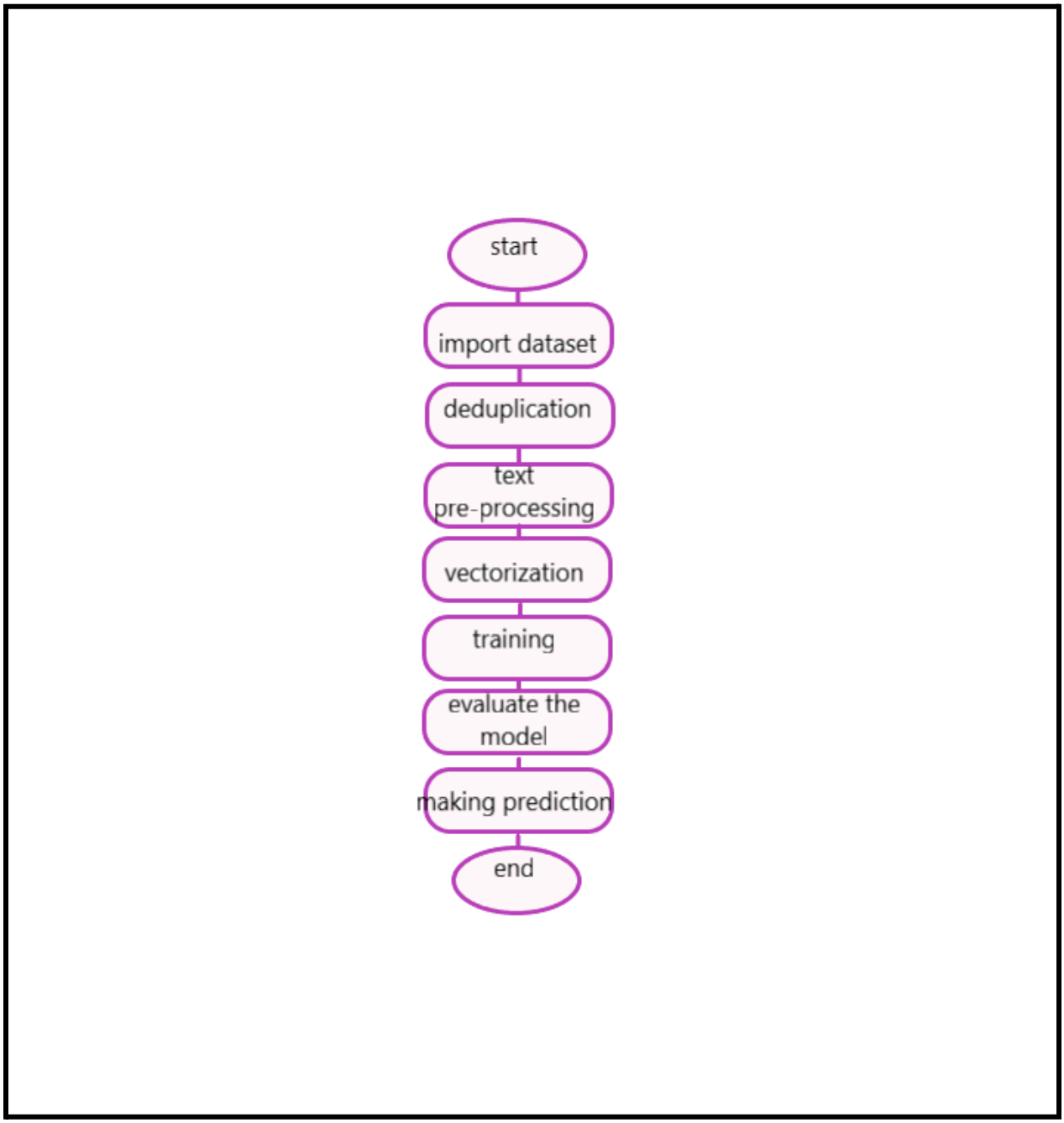
Flask:Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja

Pandas:pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

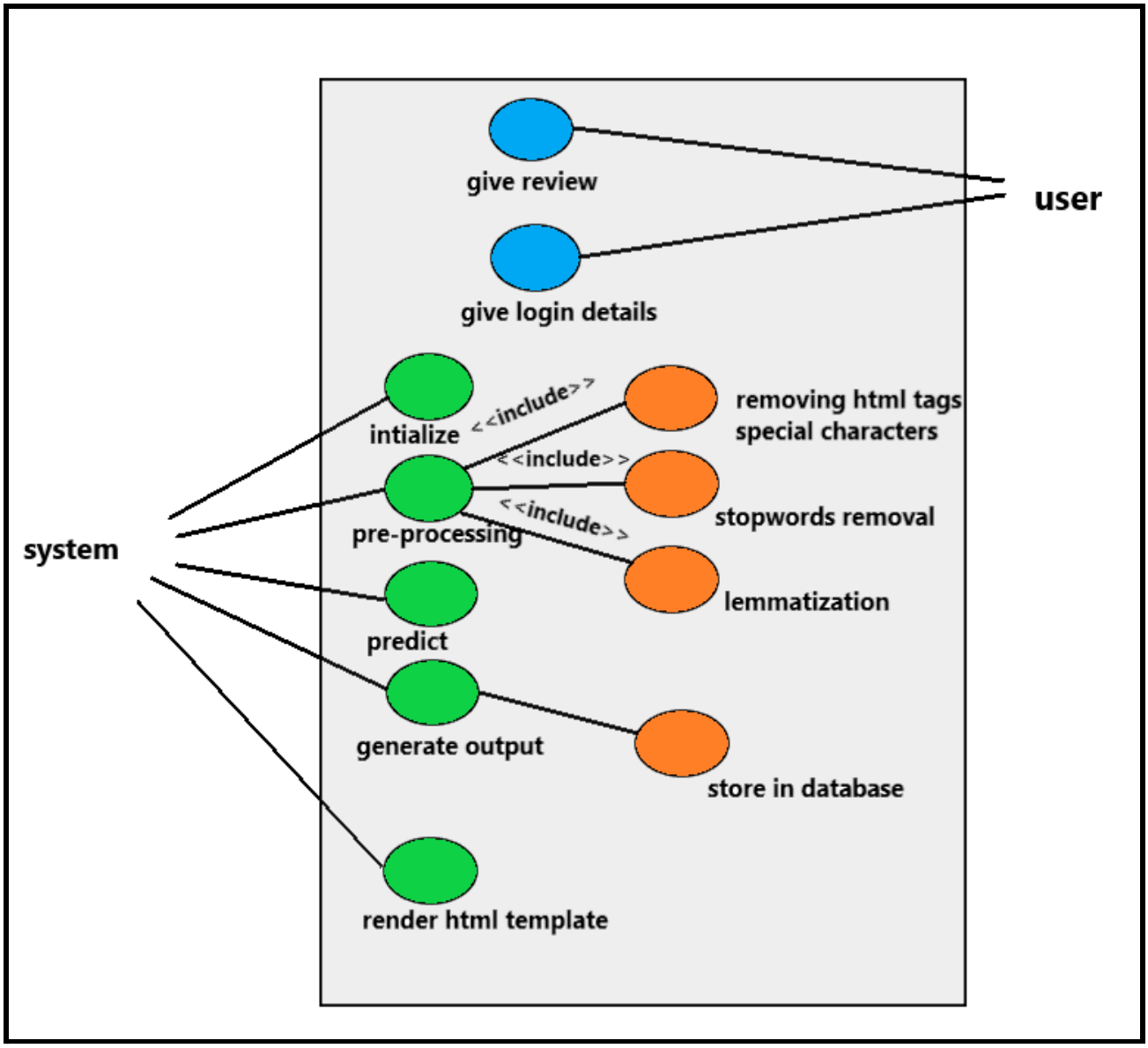
Jinja: jinja is a fast, expressive, extensible templating engine. Special placeholders in the template allow writing code similar to Python syntax

3.DESIGN

3.1 FLOW CHART:



3.2 USE CASE DIAGRAM:



4.IMPLEMENTATION

4.1 AMAZON FINE FOOD REVIEW DATASET:

This dataset is taken from **kaggle.com** and it consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all ~500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.

Data includes:

- Reviews from Oct 1999 - Oct 2012
- 568,454 reviews
- 256,059 users
- 74,258 products
- 260 users with > 50 reviews

4.1.1 EXPLORATORY DATA ANALYSIS:

This dataset has 10 column values

Attribute Information:

1. Id
2. ProductId - unique identifier for the product
3. UserId - unique identifier for the user
4. ProfileName
5. HelpfulnessNumerator - number of users who found the review helpful
6. HelpfulnessDenominator - number of users who indicated whether they found the review helpful or not
7. Score - rating between 1 and 5
8. Time - timestamp for the review
9. Summary - brief summary of the review
10. Text - text of the review

For analysis, we take score and text columns as data frames for training purposes. This dataset has a score of one to five, taking 1 and 2 as zero. 4 and 5 as one, neglecting score value 3 because the score value three is considered a neutral statement.

This **Binary classification** is the task of classifying the elements of a set into two groups on the basis of a classification rule. To classify we take 1 and 0 as two state classifiers.

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

4.1.2 HANDLING IMBALANCED DATASET:

Data imbalance usually reflects an unequal distribution of classes within a dataset. In Amazon fine food dataset the score column has one to five values after converting the values as 0 and 1 for the value of 1 there are 4,43,777 reviews and for score value one there are 82,037 reviews which is highly imbalanced. When we perform training on this data the output will be always one which is always positive for value 0 also. To handle these types of datasets we have two techniques: oversampling and undersampling which balances the dataset.

```
food_data.Score.value_counts()
```

```
5    363122
4     80655
1     52268
3     42640
2     29769
Name: Score, dtype: int64
```

After converting as 1 and 0

```
food_data.Sentiment.value_counts()
```

```
1    443777
0     82037
Name: Sentiment, dtype: int64
```

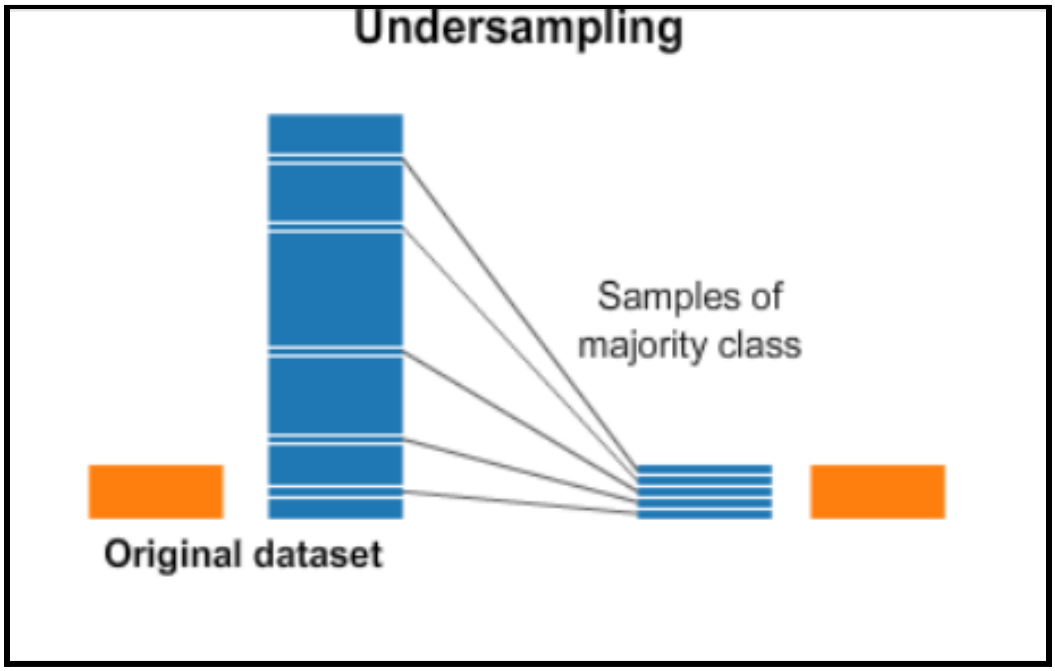
4.1.2.1 UNDER SAMPLING OF DATASET:

Undersampling refers to a group of techniques designed to balance the class distribution for a classification dataset that has a skewed class distribution.

An imbalanced class distribution will have one or more classes. where class 1 is the majority class and class 0 is the minority class in above dataset

sklearn.utils.resample(*arrays,replace=True,n_samples=None,random_state=None, stratify=None)

This function reduces the two classes based on number of samples that user defined



```
food_data_major_downsampled = resample(food_data_major,replace=False,n_samples=82037,random_state=123)
food_data_major_downsampled.Sentiment.value_counts()
food_data_balanced = pd.concat([food_data_minor,food_data_major_downsampled])
food_data_balanced.Sentiment.value_counts()
```

```
1    82037
0    82037
Name: Sentiment, dtype: int64
```

4.2 TEXT PRE-PROCESSING:

Text preprocessing is traditionally an important step for natural language processing (NLP) tasks. It transforms text into a form that is **predictable** and **analyzable** for our task.

There are different types of pre-processing techniques such as

1. Remove HTML tags
2. Remove extra whitespaces
3. Expand contractions
4. Remove special characters
5. Lowercase all texts
6. Convert number words to numeric form
7. Remove numbers
8. Remove stopwords
9. Lemmatization

For this dataset we use 3 techniques like

4.2.1 REMOVING OF SPECIAL CHARACTERS AND HTML TAGS:

The reviews or texts are web scraped, chances are they will contain some HTML tags. Since these tags are not useful for our NLP tasks, it is better to remove them. So we remove tags like:
,#.
We remove special characters which don't have any meaning to analyze, special characters like: #,@,&,%,$,|,\,/,!.

4.2.2 STOPWORDS:

stopwords are very common words. Words like “we” and “are” probably do not help at all in NLP tasks such as sentiment analysis or text classifications. Hence, we can remove stopwords to save computing time and efforts in processing large volumes

of text.

4.2.3 LEMMATIZATION:

Lemmatization is the process of converting a word to its root form, e.g., “caring” to “care”.

4.3 VECTORIZATION:

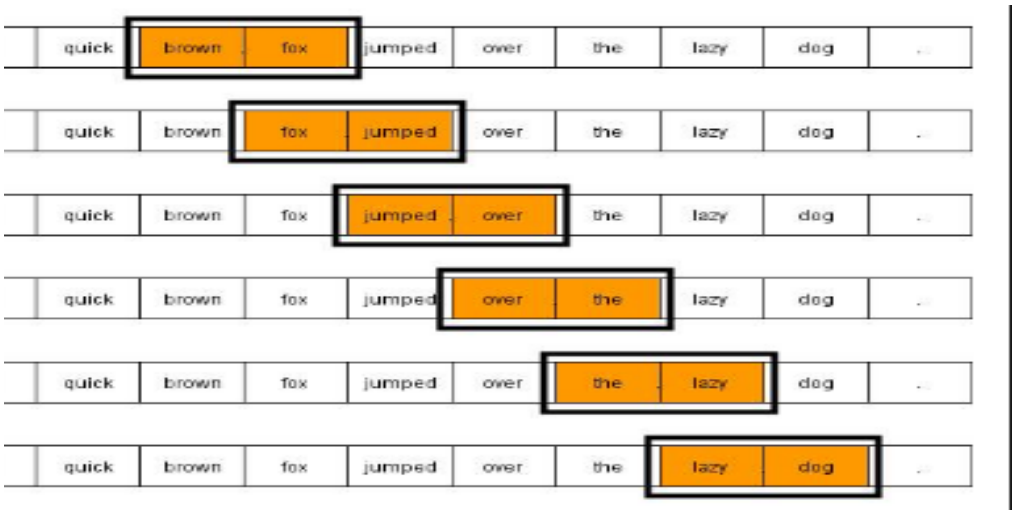
Word Embeddings or Word vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers which used to find word predictions, word similarities/semantics.

Types of vectorizations:

1. Bag of words
2. Word2vec
3. N-grams
4. Tf-idf

For this project we used ngrams with countvectorizer to build vocabulary

N-grams: is a contiguous sequence of *n* items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The *n*-grams typically are collected from a text or



speech corpus.

vect = CountVectorizer(ngram_range = (1,2)).fit(X_train)

X_train_vectorized = vect.transform(X_train)

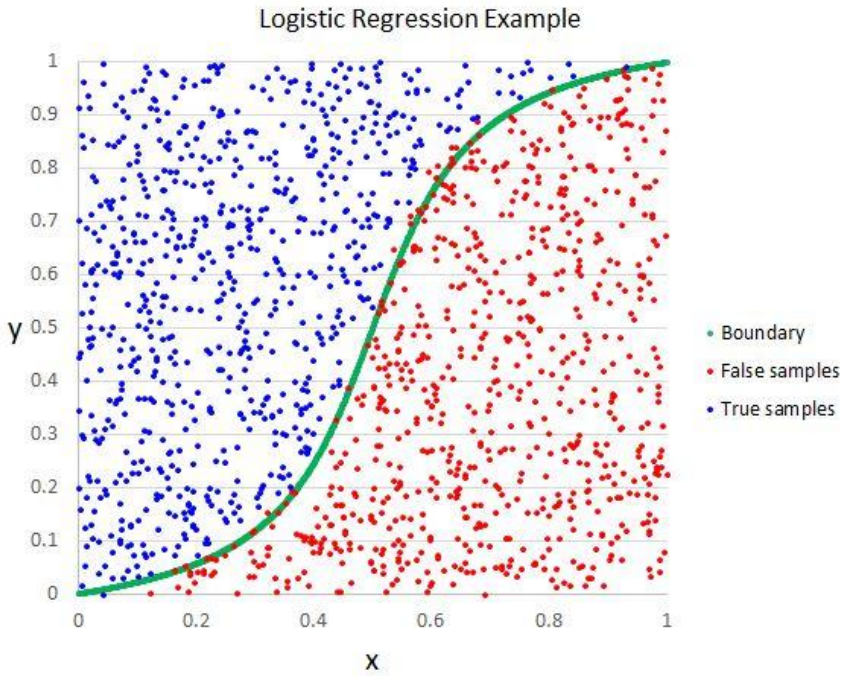
Countvectorizer has default arguments like stopwords, tokenizer, encoding.

4.4 LOGISTIC REGRESSION AND TRAINING THE MODEL:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

For this project the output be 0 and 1 which can be explained 0 as negative and 1 as positive



4.4.1 TRAINING THE MODEL:

To train the model I split the total data in a ratio of 70:30. Where 70% data is used for training the model and 30% of data for testing the trained model.

This entire data is sorted on the basis of the time and year of the review is posted to improve the accuracy of the model for the further reviews.

```
model1 = LogisticRegression(max_iter = 100)
model1.fit(X_train_vectorized, y_train)
pickle.dump(model1,open("logetest.pkl",'wb'))
```

```
print(model1.predict(vect.transform(["I ordered in swiggy I got my order like this without cover and you people tell that you
```

```
[1]
```

4.4.2 PERFORMANCE ANALYSIS:

AUC-ROC CURVE:

The **Receiver Operator Characteristic (ROC)** curve is an evaluation metric for binary classification problems. It is a probability curve that plots the **TPR** against **FPR** at various threshold values and essentially **separates the 'signal' from the 'noise'**. The **Area Under the Curve (AUC)** is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

After training the model we test the model with test data set to find the accuracy of the model so, we can understand how perfect the model is predicting the positive and negative class

For our model we get **94% accuracy**

4.5 FLASK IMPLEMENTATION:

Flask:
Flask is an API of Python that allows us to build up web-applications by using the flask library tools. which include functions such as Flask (for app creation), render_template(to establish a connection between pages)

```
@app.route("/sign")
def imfint2():
    return  render_template("signup.html")
```

App routing:
App routing is used to map the specific URL with the associated function that is intended to perform some task.This acts as location marks for html pages that guides them to the precise function. On performing further tasks it will lead to the next page.

ML integration:
We will further load the previously trained models that are stored in pickles to process our data. We will pass the review as a list to transform into a vector, later we will pass the obtained vector to ml model to predict. These predictions help us to obtain a conclusion.The possible predictions are either 1 or 0.

```
vectorizer = pickle.load(open("vect.pkl", "rb"))
ve= pickle.load(open("logetest.pkl", "rb"))
gg = vectorizer.transform([review])
fg  =  ve.predict(gg)
```

Template rendering:
Jinjall is a template engine for Python. It is used to create HTML, XML or other markup formats that are returned to the user via an HTTP request.

4.6 MONGODB:

4.6.1 CONNECTION ESTABLISHMENT:

```
client=MongoClient('mongodb://localhost:27017')
db=client.get_database('aravind')
login=db.login
reviews=db.review
```

The URI connection string specifies connecting to a MongoDB instance that is running on localhost using **port 27017**.

4.6.2 DATABASE TABLES:

we have divided our data into two tables namely,

1.login table
2.review table

Login table:
To collect the information from the user about their personal details ,like mail ,phone number, name.

Review table:
This table will contain all the necessary data that is needed to make an analysis for our project.

Status:
The status value is concluded based on the sentiment analysis of the review which will be simultaneously calculated before insertion into the database.This row is further used to get the count of no of visitors,no of positive ones and negative ones and the average rating which will be further used to provide data for the graphs.

5.WEB APP

5.1 SIGN-UP PAGE:

This is the page page of the web app ,which is built to collect the data from the user such as their name,mail id ,phone number .when every detail is provided correctly by the user, the data will be passed for evaluation .When a user provides his/her details the web app will direct the user to the next page .when an admin provides his details the web app will lead him to admin page.

5.2 ITEMS LIST:

This page contains all the items that are provided ,along with their description and a button which will guide to the feedback form on click. We implemented flip cards for a better view of a restaurant page.

5.3 FEEDBACK FORM:

This is the most important page in the web app, the purpose of this page is to collect the review provided by the user which is main objective of the web app .When provided with a review that the user think is suitable .The review will be sent to the flask app for further processing.After this the user will be guided to the thank you page.

5.4 ADMIN PAGE:

The admin page is the place where all the processed data is displayed. The admin page includes :

1.graphs.
2.tables.
3.comments section.
4.data count.

These are used to uniquely describe the data we have.Admins manage and maintain websites, taking into account functionality, appearance, content, and performance. Their main focus is on the technical aspects of web maintenance, such as building servers and troubleshooting, but they also update the more visible parts of the website, tweaking the design or adding new sections.

6. SOURCE CODE

6.1 ML APPLICATION CODE:

```
#Libraries used
import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import numpy as np
from sklearn.utils import resample
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import pickle

#Loading the dataset
food_data = pd.read_csv(r'F:/Dataset/archive (2)/Reviews.csv')

#Taking only Score and Text columns
food_data = food_data[['Score','Text']]

#Removing the scores which have rating 3
food_data = food_data[food_data.Score != 3]
def f(r):
    if r > 3:
        return 1
    else:
        return 0

food_data['Sentiment'] = food_data.Score.map(f)
food_data = food_data.groupby('Sentiment')

#Taking the scores of 1 and 2 as food_data_minor
#Taking the scores of 4 and 5 as food_data_major
for i,j in food_data:
    if i==0:
        food_data_minor = j
    if i==1:
        food_data_major = j
```

#The training dataset is highly imbalanced to balance the dataset we are downsampling of food_data_major which will be equal to food_data_minor

#Taking 82,037 rows of both minor and major dataset
food_data_major_downsampled=resample(food_data_major,replace=False,n_samples=82037,random_state=123)

#Concatenating both the datasets
food_data_balanced = pd.concat([food_data_minor,food_data_major_downsampled])
food_data_balanced.drop(['Score'],axis = 1, inplace = True)

#Taking X as text column and Y as sentiment column
X= food_data_balanced["Text"]
y = food_data_balanced["Sentiment"]

#Splitting the dataset to 70% as training and 30% for testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)

#Converting the datasets into vector model
#CountVectorizer has default parameters as removing stop words,lemmatization,special characters and numbers
vect = CountVectorizer(ngram_range = (1,2)).fit(X_train)
X_train_vectorized = vect.transform(X_train)

#Pickling the model for deployment into flask web app
pickle.dump(vect,open("vect.pkl",'wb'))

#Training the dataset with logistic regression model
model1 = LogisticRegression(max_iter = 100)
model1.fit(X_train_vectorized, y_train)

#Pickling the trained model for deployment into flask web app
pickle.dump(model1,open("logetest.pkl",'wb'))

6.2 WEB APP CODE:

#Libraries used
from flask import Flask, render_template,request,redirect,url_for ,session
from pymongo import MongoClient
from datetime import datetime
import pickle
import numpy as np
import sqlite3
import pandas as pd
import nltk
import string
import matplotlib.pyplot as plt

```
from sklearn.utils import resample
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
#Establishing a connection to the data-base
client=MongoClient('mongodb://localhost:27017')
db=client.get_database('aravind')
login=db.login
reviews=db.review
#Creates the flask app instance
app=Flask(__name__)
app.secret_key = 'any randon key'
#Routing to specific URL
#Directing to welcome page
@app.route("/")
def imfint():
   session['seq']='one'

   return  render_template("welcome.html")
#Directing to sign-up  page
@app.route("/sign")
def imfint2():
   return  render_template("signup.html")
#Receiving data from the user
@app.route("/admin",methods=["POST","GET"])
def logins():

   if request.method=='POST'  :
      if request.form.get('uname')=='admin' or request.form.get('uname')=='admin2' or
request.form.get('perks')=='admin':
#Fetching the data from the database using find
         ver=reviews.find({},{ "_id":0, "Name":1, "Item":1, "Review":1,"Entry_time":1
,"Entry_date":1,"Status":1}).sort("_id",-1)

         ver2=reviews.find({},{ "_id":0, "Name":1, "Item":1, "Review":1,"Entry_time":1
,"Entry_date":1,"Status":1,"Mail":1,"Phone":1}).sort("_id",-1)
         ver3=reviews.find({},{ "_id":0, "Name":1, "Item":1, "Review":1,"Entry_time":1
,"Entry_date":1,"Status":1}).sort("_id",-1)

          #Items reviews
         ver4=reviews.find({"Item":"Veggie -paradise"},{ "_id":0, "Name":1, "Item":1,
"Review":1,"Entry_time":1 ,"Entry_date":1,"Status":1}).sort("_id",-1)
         ver5=reviews.find({"Item":"Margherita"},{ "_id":0, "Name":1, "Item":1,
"Review":1,"Entry_time":1 ,"Entry_date":1,"Status":1}).sort("_id",-1)
         ver6=reviews.find({"Item":"Capricciosa"},{ "_id":0, "Name":1, "Item":1,
```

```
"Review":1,"Entry_time":1 ,"Entry_date":1,"Status":1}).sort("_id",-1)

        ver7=reviews.find({"Item":"Cheesy-Delight"},{ "_id":0, "Name":1, "Item":1,
"Review":1,"Entry_time":1 ,"Entry_date":1,"Status":1}).sort("_id",-1)
         ver8=reviews.find({"Item":"Hawaiian"},{ "_id":0, "Name":1, "Item":1,
"Review":1,"Entry_time":1 ,"Entry_date":1,"Status":1}).sort("_id",-1)
        ver9=reviews.find({"Item":"Caeasars pizza"},{ "_id":0, "Name":1, "Item":1,
"Review":1,"Entry_time":1 ,"Entry_date":1,"Status":1}).sort("_id",-1)

        bb=db.review.count()
        bc=db.review.distinct('Name')
        bc=len(bc)
        counts=countz(ver)
        counts=round(counts,2)
        #Counting the no of positive and negative reviews for individual items
        it1,it11=countzz(ver4)
        it2,it22=countzz(ver5)
        it3,it33=countzz(ver6)
        it4,it44=countzz(ver7)
        it5,it55=countzz(ver8)
        it6,it66=countzz(ver9)

         if request.form.get('uname')=='admin':
           pv=0
        elif request.form.get('uname')=='admin2':
            pv=1
#Directing the processed data  to Admin  page
         return
render_template('dashboardbase.html',pv=pv,vers=ver2,versi=ver3,visit=bb,rati=counts,unq=bc,it
1=it1,it2=it2,it3=it3,it4=it4,it5=it5,it6=it6,it11=it11,it22=it22,it33=it33,it44=it44,it55=it55,it66=it66)

      else:
        user=request.form.get('uname')
        phone=request.form.get('uphone')
        mail=request.form.get('umail')
         session['sequence1'] = user
         session['sequence3'] = phone
         session['sequence4'] = mail
         login.insert_many([{"Name":user,"Phone":phone,"Email":mail}])
#Directing to flipcards
        return render_template('flipcard.html')
#processing the reviews
def countz(ver):
   count=0
   n=0
   for x in ver:
      b=int(x["Status"])
      count=count+b
```

```
        n=n+1
    try:
        count=count*5/n
        return count
    except:
        count=0
        return count
def countzz(ver2):
    count=0
    count2=0
    for x in ver2:
        if(x["Status"]==0):
            count=count+1
        else:
            count2=count2+1
    return (count,count2)
```

#Receiving data from flipcards page

```
@app.route("/item",methods=["POST"])
def items():
    if request.method=='POST' :


        item=request.form.get('item')
        session['sequence2'] = item
        bver12=reviews.find({}).sort("_id",-1).limit(15)
        verr=reviews.find({},{ "_id":0, "Name":1, "Item":1, "Review":1,"Entry_time":1
,"Entry_date":1,"Status":1}).sort("_id",-1)

        count=0
        n=0
        for x in verr:
            b=int(x["Status"])
            count=count+b
            n=n+1
        try:
            count=count*5/n
            mn=int(count)
        except:
            count=0
        mn= int(count)
        mm=5-mn
```

#Directing to input page

```
        return render_template('input.html',speec=bver12,cnts=mn,cnty=mm)
```

#Receiving data from the input page

```
@app.route("/review",methods=["POST"])
def comment():
   if request.method=='POST' and 'review' in request.form and session['seq']=='one' :
      review=request.form.get('review')
      timing=datetime.now()
      timingdate=timing.strftime("%x")
      timingtime=timing.strftime("%X")
      user= session['sequence1']
      item= session['sequence2']
      phone=session['sequence3']
      mail= session['sequence4']
      session['seq']='two'
      vectorizer = pickle.load(open("vect.pkl", "rb"))
      ve= pickle.load(open("logetest.pkl", "rb"))
      gg = vectorizer.transform([review])
      #Performing sentiment analysis for the review

      fg  =  ve.predict(gg)
      fg=str(fg[0])
      fg=int(fg)

#Inserting data into the data-base

reviews.insert_many([{"Name":user,"Item":item,"Review":review,"Status":fg,"Mail":mail,"Phone":pho
ne,"Entry_time":timingtime,"Entry_date":timingdate}])
       #Directing to Thank-you page

       return render_template('thank.html')
   else:
      return '<h1>Please return to home page</h1>'
#Directing to creators page

@app.route("/profile")
def toprofile():
   return render_template('profile.html')
@app.route("/homed")
def tohome():
    return render_template('signup.html')

if __name__ == "__main__":

   app.run(debug=True)
```

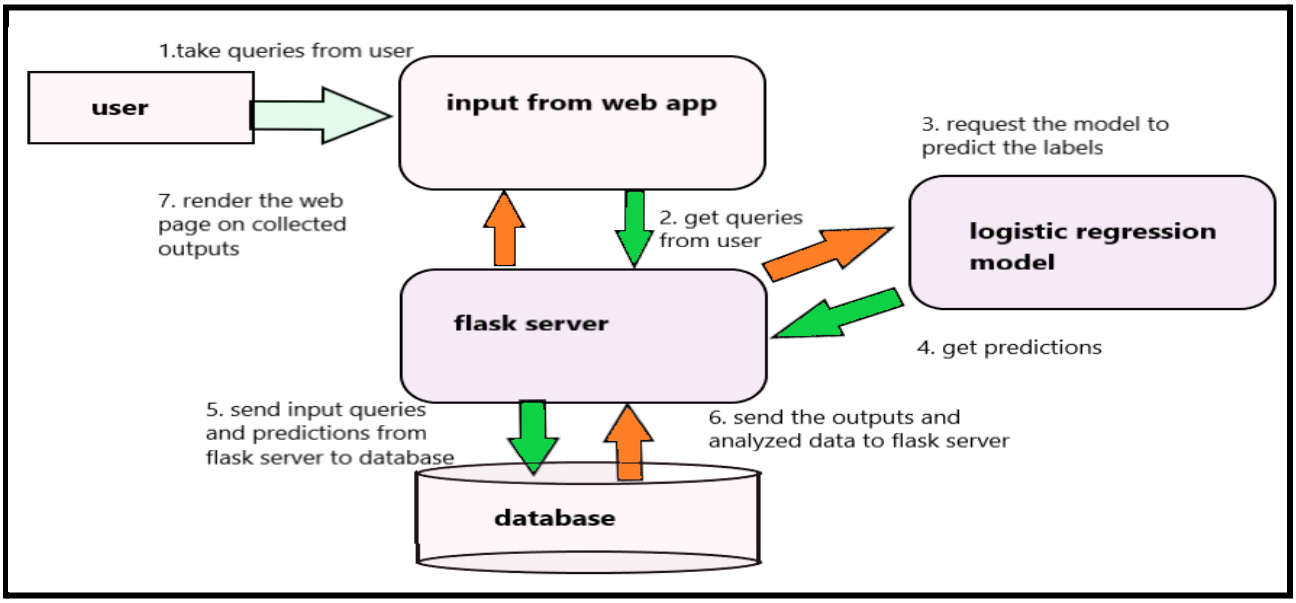
7.Deployment

7.1 PICKLING THE ML MODEL:

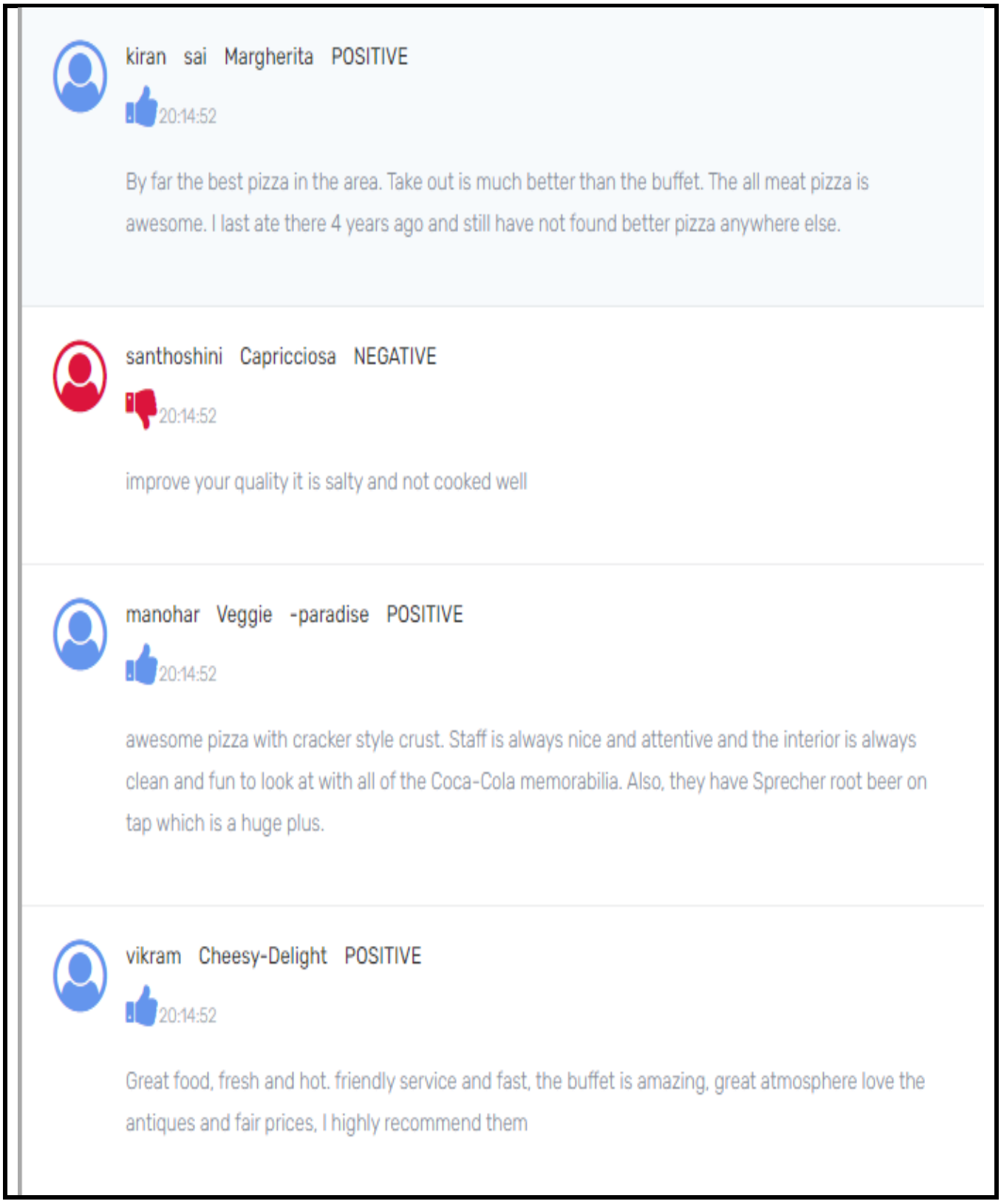
Pickle is the standard way of serializing objects in Python. We use the pickle operation to serialize your machine learning algorithms and save the serialized format to a file and Later we load this file to deserialize your model and use it to make new predictions.

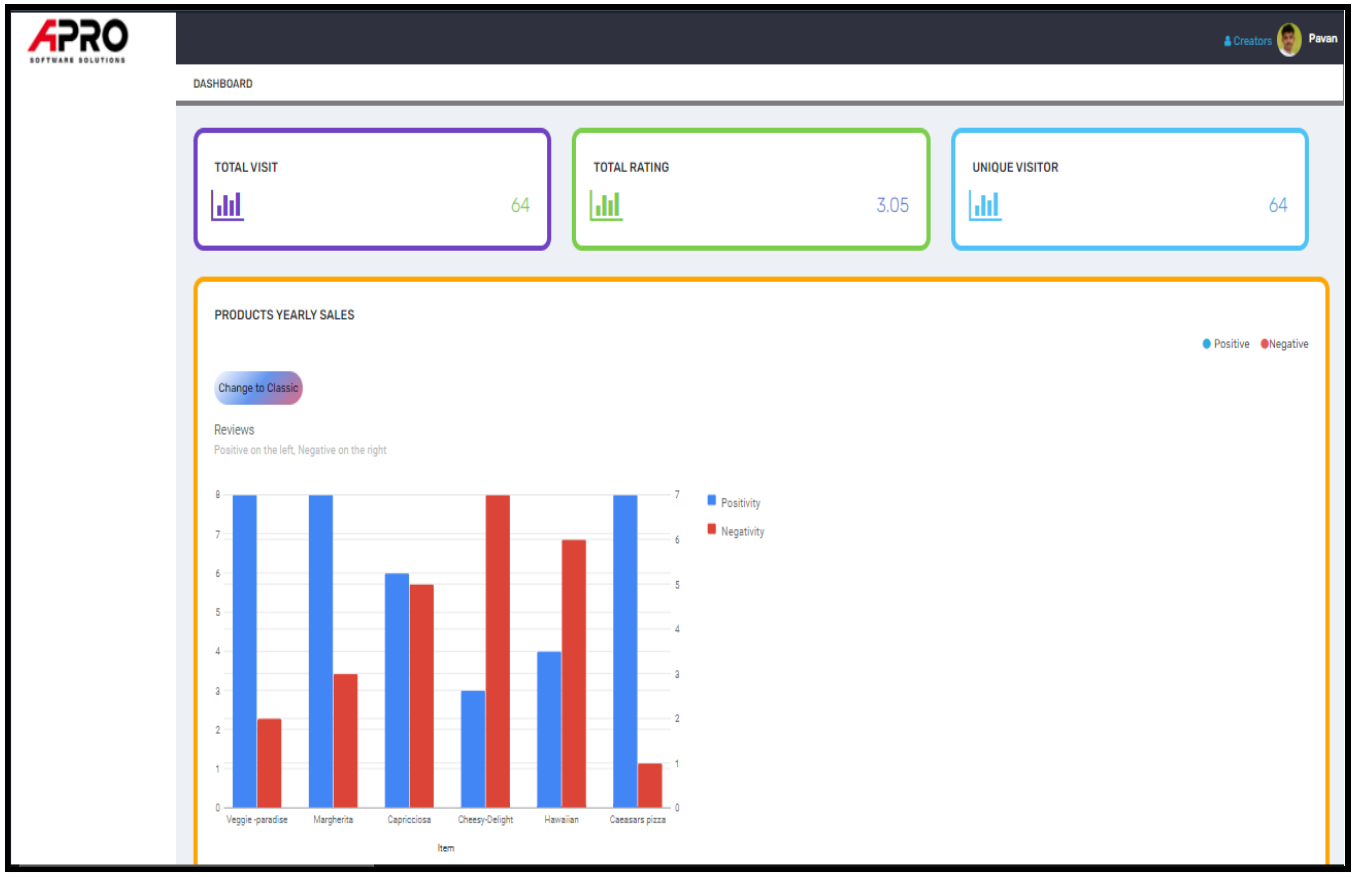
7.2 DEPLOYMENT IN FLASK:

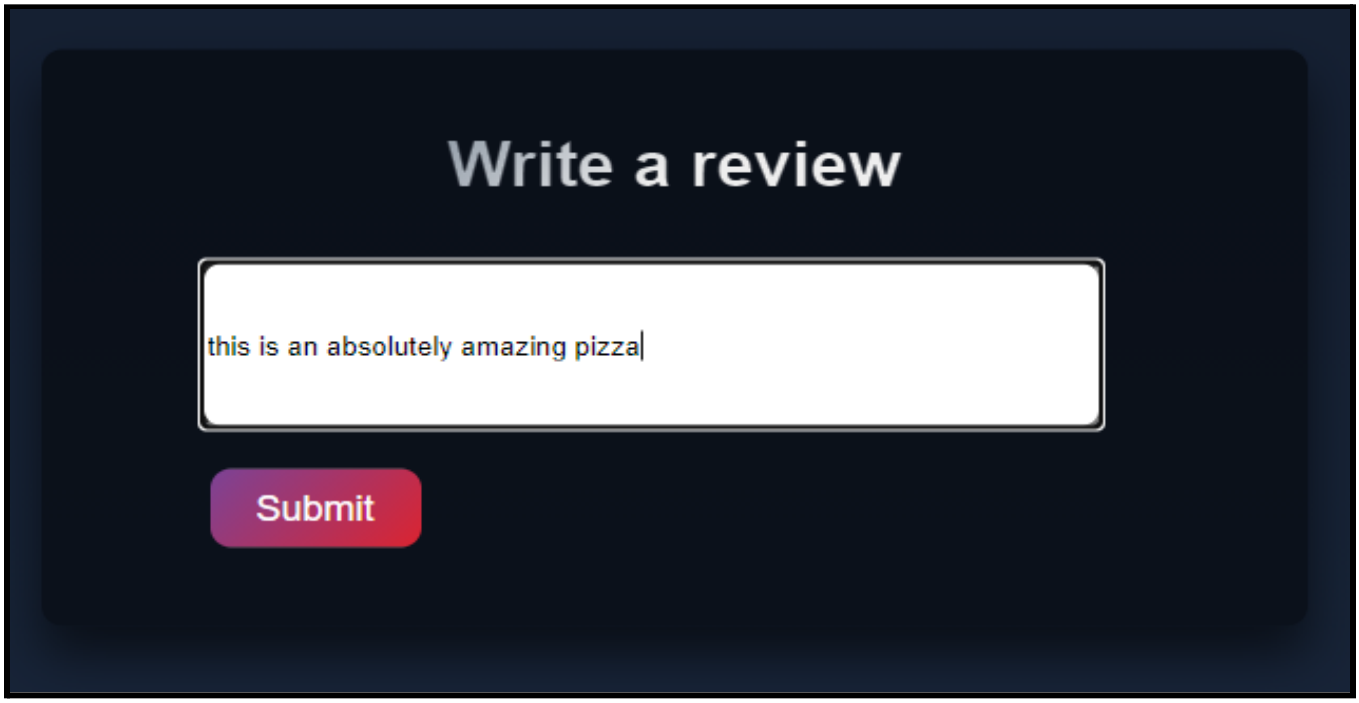
Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Now, we have a trained model that works well to predict the reviews positive or negative. Also, we have a pickle file developed that allows us to serve this model. We need to encapsulate the code and port the application to different platforms. For that, we used flask. Flask allows encapsulating the code in a container and porting the application in an easy way.



7.3 OUTPUT SCREENSHOTS:

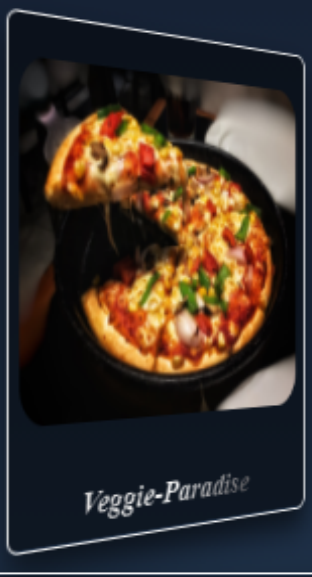



APRO
SOFTWARE SOLUTIONS
Creators
Pavan
DASHBOARD
TOTAL VISIT
64
TOTAL RATING
3.05
UNIQUE VISITOR
64
PRODUCTS YEARLY SALES
Positive
Negative
Change to Classic
Reviews
Positive on the left, Negative on the right
Positivity
Negativity
Veggie -paradise
Margherita
Capricciosa
Cheesy-Delight
Hawaiian
Caeasars pizza
Item


Write a review
this is an absolutely amazing pizza
Submit

SELECT WHAT MADE YOU "HAPPY"

Hover over a card to flip it.



Veggie-Paradise

Margherita

Capricciosa

Cheesy-Delight

Hawaiian

Caesars pizza

8. TESTING

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

8.1 Purpose of Testing:

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of the testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors.

8.2 Testing Objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally it can be said that,

- Testing is a process of executing a program with the intent of finding an error.
- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.
- The software and hardware more or less confirms to the quality and reliable standards. We use 24,500 reviews and from amazon fine food dataset as our testing data to validate our training model.
- We tested the web app for code testing by executing every line of the code and how the flask renders the html to show output , tested the database connections and database query statements, tested graph API's.

8.3 LEVELS OF TESTING:

In order to uncover the errors present in different phases we have the concept of testing. The basic levels of testing are as shown below:

System Testing:

The philosophy behind testing is to find errors. Test cases are devised with

this in mind. A strategy employed for system testing is code testing.

Code Testing:

This strategy examines the logic of the program. To follow this method we developed some test data that resulted in executing every instruction in the program and module i.e every path is tested. Systems are not designed as entire nor are they tested as single systems. To ensure that the coding is perfect two types of testing is performed or for that matter is performed on all systems.

White Box Testing:

This is a unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible error. I tested step wise every piece of code, taking care that every statement in the code is executed at least one. The white box testing is also called Glass Box Testing. I have generated a list of test cases, sample data Which is used to check all possible combinations of execution paths through the code at every module level.

Black Box Testing:

This testing method considers a module as a single unit and checks the unit at interface and communication with the other modules rather getting into details at statement level. Here the module will be treated as a block box that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

8.4 AUC-ROC CURVE:

The **Receiver Operator Characteristic (ROC)** curve is an evaluation metric for binary classification problems. It is a probability curve that plots the **TPR** against **FPR** at various threshold values and essentially **separates the 'signal' from the 'noise'**. The **Area Under the Curve (AUC)** is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and

all Positives as Negatives.
When **0.5<AUC<1,** there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. This is so because the classifier is able to detect more numbers of **True positives** and **True negatives** than **False negatives** and **False positives**.

When **AUC=0.5**, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.

After training the model we test the model with test data set to find the accuracy of the model so, we can understand how perfect the model is predicting the positive and negative class

For our model we get **94% accuracy**

```
from sklearn.metrics import roc_auc_score

predictions = model1.predict(vect.transform(X_test))

print('AUC: ', roc_auc_score(y_test, predictions))
```

```
AUC:  0.9411859755791537
```

```
print(model1.predict(vect.transform(['The sweet does not  tastes good, I would never buy them again',
                                     'The sweet is not bad, I will buy them again'])))
```

```
[0 1]
```

9.CONCLUSION

The purpose of the paper is to investigate the state of the art of textual translation theories, methods, and tools into formal and numerical requirements to support information modelling and project management process. Natural Language Processing helps translate text requirements into numerical terms that are necessary for the application and success of information modelling and management in a data-driven process.