

CAPSTONE PROJECT

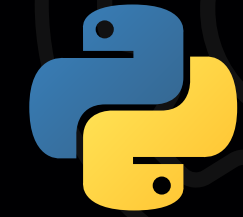
MESS MANAGEMENT SYSTEM

DATA ANALYTICS & ML

- We have used Exploratory Data Analysis to work on dashboards. The dashboard analytics we are also providing a healthy competition between the catering services through data analytics.
- We are visualizing and providing dashboards that can help the catering services understand and tackle their problems accordingly.
- With the help of sentimental analysis , we are analysing the feedback collected by students and help the catering service for more healthy and better services.



Exploratory Data Analysis



01

Dataset

We have conducted various suveys and collected the data from our VITAP students. Here is a link to check out how are data looks :-

https://drive.google.com/drive/folders/1OZ03rs15CfA19jHCB3QAurt0XmQcs20A?usp=share_link

The image shows a 'Campus Catering Feedback' form on the left and an Excel spreadsheet on the right. The form includes fields for 'Email', 'Mention Year' (Freshner, Sophomore, Junior, Senior), 'Current Mess' (A&R, CRCL, IBVTH), and 'Quality of food' (1-4). The Excel spreadsheet has columns for 'Mention Year', 'Current Mess', 'Quality of food', 'Food Variety', 'Adequacy of food and position', 'Hygiene Level', 'Overall Rating', and 'Review'. It contains multiple rows of student feedback data.

02

Data Pre-Processing

To start with, we imported necessary libraries for this elike pandas, numpy,matplotlib and seaborn and loaded the data set. We proceeded by checking the columns and their corresponding data types, along with finding whether they contain null values or not.

```
Importing the Libraries and reading the tsv data file

[1] import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

data=pd.read_csv('/content/_CRCL Mess data (2).xlsx - Sheet1 (2).tsv', delimiter = '\t', quoting = 3)
data.head()

Mention Year Current Mess Quality of food Food Variety Adequacy of food and position of queue at the counter
0 Junior CRCL 4 3
1 Fresher CRCL 3 1
2 Fresher CRCL 2 3
3 Sophomore CRCL 3 2
4 Junior CRCL 1 1

[ ] import seaborn as sns
sns.countplot(x='Liked', data=data)
plt.show()
```


Exploratory Data Analysis

03

Data Visualization

With the help of statistical summary and graphical representations by using the visualization libraries like Matplotlib and Seaborn we have done Exploratory Data Analysis to all the three different datasets of the catering services we have. So once we have visualized using univariate data analysis we have deployed these graphical representations to our dashboards.



We have made graphical representations of different catering services mainly from the data we have collected which focusses more on

- Quality Of Food
- Food Variety
- Adequacy of food
- Hygiene Levels
- Overall Rating.



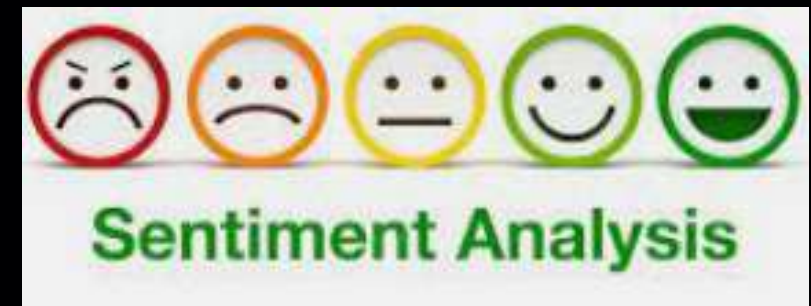
RULE BASED NLP SENTIMENTAL ANALYSIS FOR MESS REVIEWS

- Sentiment Analysis is a branch of natural language processing (NLP) that analyses unstructured text to determine how strongly people feel about opinions (Positive/Negative/Neutral).
- Both rule-based and machine learning approaches can be used to analyze sentiment.

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written -- referred to as natural language.

There are two main phases to natural language processing that we have worked on

- Data Preprocessing
- Algorithm development.



RULE BASED NLP SENTIMENTAL ANALYSIS FOR MESS REVIEWS

- Rule-based NLP has improved accuracy relative to keyword extraction.
- A rule-based NLP system simply follows these rules to categorise the language it's analysing.
- This is a practical approach to analyzing text without training or using machine learning models. The result of this approach is a set of rules based on which the text is labeled as positive/negative/neutral.

Widely used Rule-based approaches are :-

- TextBlob
- VADER (Valence Aware Dictionary and sEntiment Reasoner)
- SentiWordNet.

In our Project we have used TextBlob and VADER lexicon based sentiment analysis algorithms for our dataset.



Data Pre-Processing

01

Cleaning the text

In this step, we basically remove the special characters, numbers from the text. We have used the regular expression operations library of Python. The “clean” function that takes text as input and returns the text without any punctuation marks or numbers in it.

Review	Cleaned Reviews
Fried food quantity must be increased along wi...	Fried food quantity must be increased along wi...
Okay not bad food	Okay not bad food
nice food	nice food
chicken is nice	chicken is nice
nice food	nice food

02

Tokenization

In this step, using nltk tokenize function word_tokenize(), which helps to break the text into smaller pieces called Tokens. It can be performed at sentences(sentence tokenization) or word level(word tokenization).

```
import nltk
nltk.download('averaged_perceptron_tagger')
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
from nltk import pos_tag
nltk.download('stopwords')
from nltk.corpus import stopwords
nltk.download('wordnet')
from nltk.corpus import wordnet
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] /root/nltk_data...
[nltk_data] Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

03

Enrichment – POS tagging

Parts of Speech (POS) tagging is a process of converting each token into a tuple having the form (word, tag). POS tagging essential to preserve the context of the word and is essential for Lemmatization.

Review	Cleaned Reviews	POS tagged
Fried food quantity must be increased along wi...	Fried food quantity must be increased along wi...	[(Fried, v), (food, n), (quantity, n), (must, ...
Okay not bad food	Okay not bad food	[(Okay, n), (bad, a), (food, n)]
nice food	nice food	[(nice, a), (food, n)]
chicken is nice	chicken is nice	[(chicken, n), (nice, a)]
nice food	nice food	[(nice, a), (food, n)]

```
from nltk.stem import WordNetLemmatizer
word_lemmatizer = WordNetLemmatizer()
```


Data Pre-Processing

04

Removal Of Stopwords

.A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore. In this step, using nltk directory NLTK(Natural Language Toolkit) in python which has a list of stopwords stored in 16 different languages ,we need remove them as part of text preprocessing.

```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()

def lemmatize(pos_data):
    lemma_rew = ""
    for word, pos in pos_data:
        if not pos:
            lemma = word
            lemma_rew = lemma_rew + " " + lemma
        else:
            lemma = wordnet_lemmatizer.lemmatize(word, pos=pos)
            lemma_rew = lemma_rew + " " + lemma
    return lemma_rew

data['Lemma'] = data['POS tagged'].apply(lemmatize)
```

05

Obtaining the stem words

A stem is a part of a word responsible for its lexical meaning.

The two popular techniques of obtaining the root/stem words are :-

- Stemming
- Lemmatization

The key difference is Stemming often gives some meaningless root words as it simply chops off some characters in the end. Lemmatization gives meaningful root words, however, it requires POS tags of the words.

KODAK PORTRA 400	
Review	Lemma
d food quantity must be increased along wi...	Fried food quantity must increase along so
Okay not bad food	Okay bad fo
nice food	nice fo
chicken is nice	chicken n
nice food	nice fo
...	
Never again will I be dining at this place!	Never din pla
od was excellent and service was very good.	food excellent service go
good food selection.	good food select
Food was really boring.	Food really bor
Good Service-check!	Good Service che
columns	

Algorithm Development

01 TextBlob

TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

The two measures that are used to analyze the sentiment are:-

- Polarity – talks about how positive or negative the opinion is
- Subjectivity – talks about how subjective the opinion is

TextBlob(text).sentiment gives us the Polarity, Subjectivity values.

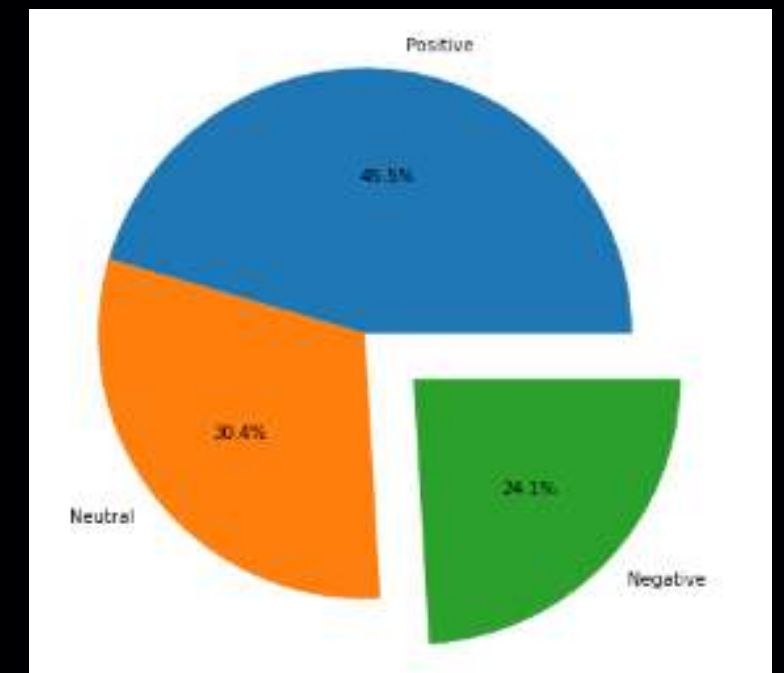
- Polarity ranges from -1 to 1 (1 is more positive, 0 is neutral, -1 is more negative)
- Subjectivity ranges from 0 to 1 (0 being very objective and 1 being very subjective)

```
from textblob import TextBlob

# function to calculate subjectivity
def getSubjectivity(Review):
    return TextBlob(Review).sentiment.subjectivity

# function to calculate polarity
def getPolarity(Review):
    return TextBlob(Review).sentiment.polarity

# function to analyze the reviews
def analysis(score):
    if score < 0:
        return 'Negative'
    elif score == 0:
        return 'Neutral'
    else:
        return 'Positive'
```



Algorithm Development

02

VADER stands for Valence Aware Dictionary and Sentiment Reasoner.

- VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.
- VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative.
- VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive).

positive sentiment : (compound score \geq 0.05)

neutral sentiment : (compound score $>$ -0.05) and (compound score $<$ 0.05)

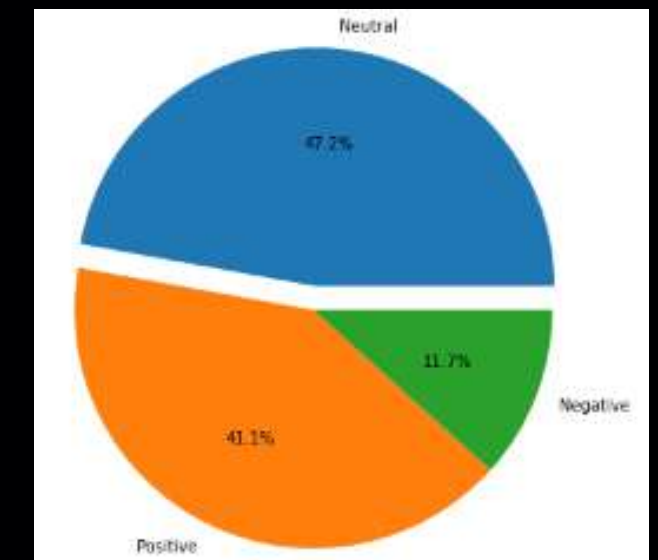
negative sentiment : (compound score \leq -0.05)

```
#in_data['Vader_Sentiment'] = #in_data['lemma'].apply(vadersentimentanalysis)

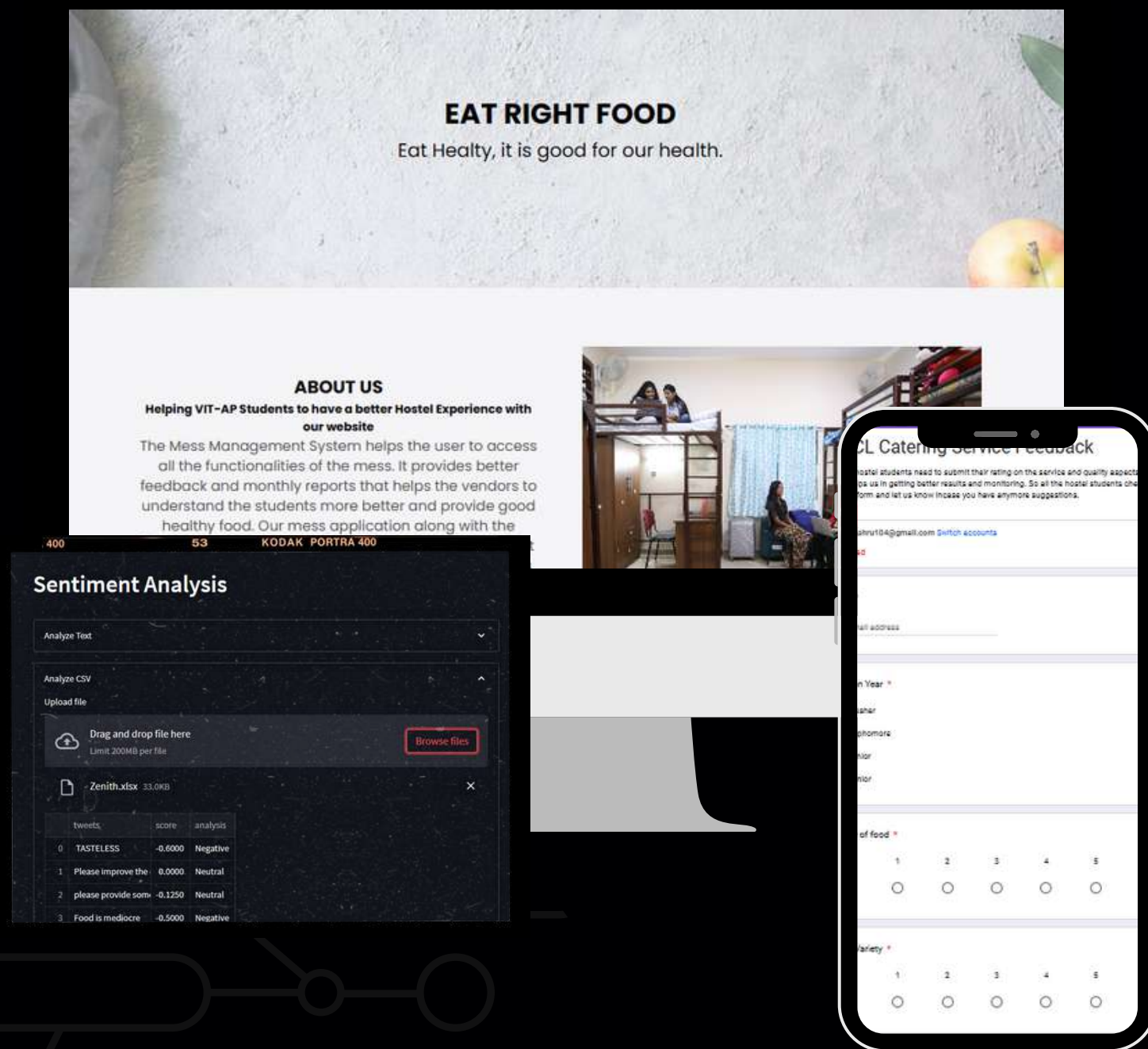
# Function to analyse
def vader_analysis(compound):
    if compound >= 0.4:
        return 'Positive'
    elif compound <= -0.5:
        return 'Negative'
    else:
        return 'Neutral'

#in_data['Vader_Analysis'] = #in_data['Vader_Sentiment'].apply(vader_analysis)
#in_data.head()
```

	Review	Lemma	Polarity	Analysis	Vader_Sentiment	Vader_Analysis
0	Fried food quantity must be increased along wi...	Fried food quantity must increase along soft...	0.1	Positive	0.5182	Neutral
1	Okay not bad food	Okay bad food	-0.1	Negative	-0.3818	Neutral
2	nice food	nice food	0.8	Positive	0.4218	Positive
3	chicken is nice	chicken nice	0.0	Neutral	0.4218	Positive
4	nice food	nice food	0.8	Positive	0.4218	Positive



FUTURE PROGRESS OF OUR WORK



- As we have developed algorithm , we will be deploying the algorithm into a website that can help the catering services .This will help the catering services to have an easy access for the sentiment analysis of the students feedback.
- We are also developing a website that will be a onestop for all the mess functionalities, this will help the students to give their feedback directly and also can check the menu and they will also be receiving the updates through email

THANK YOU