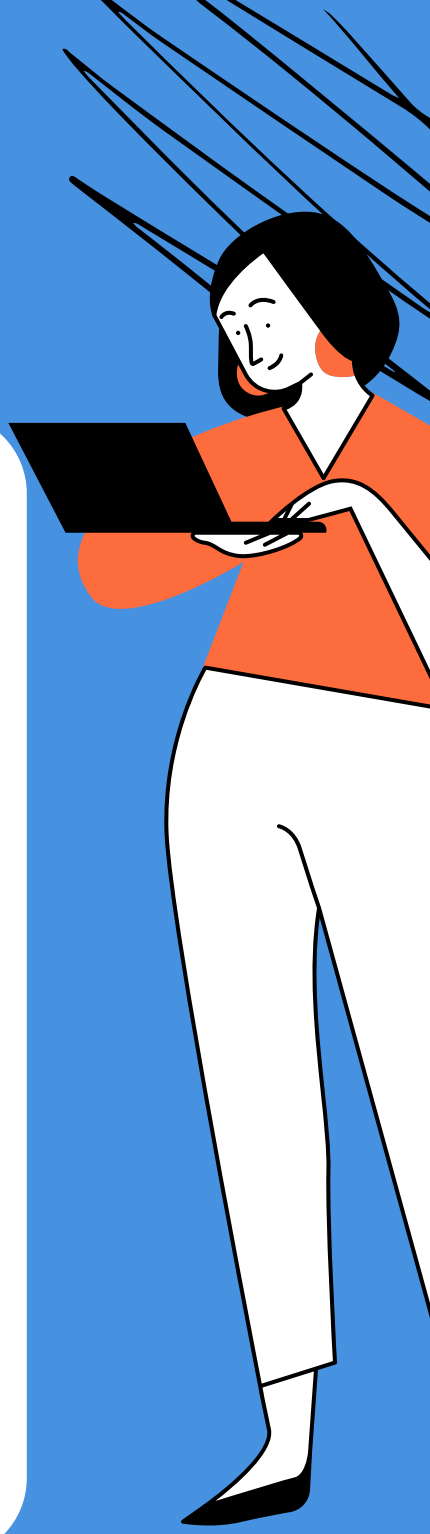
جامعة أم القرى
UMM AL-QURA UNIVERSITY



Natural Language Processing Project

Summarize Restaurant Reviews

Supervised By:
Dr. Ashwag Magrhaby

PROJECT GROUP MEMBERS

Amal Hamed Alobidi	441001849	s441001849@st.uqu.edu.sa	G2
Hala Muhammed Alshokri	441009248	s441009248@st.uqu.edu.sa	G2
Areej Adel Bawazir	439016266	s439016266@st.uqu.edu.sa	G2
Khulood AbdulMajeed	441009999	s441009999@st.uqu.edu.sa	G2
Najla atiah Alfadhli	441016659	s441016659@st.uqu.edu.sa	G2

Chapter 1: Introduction

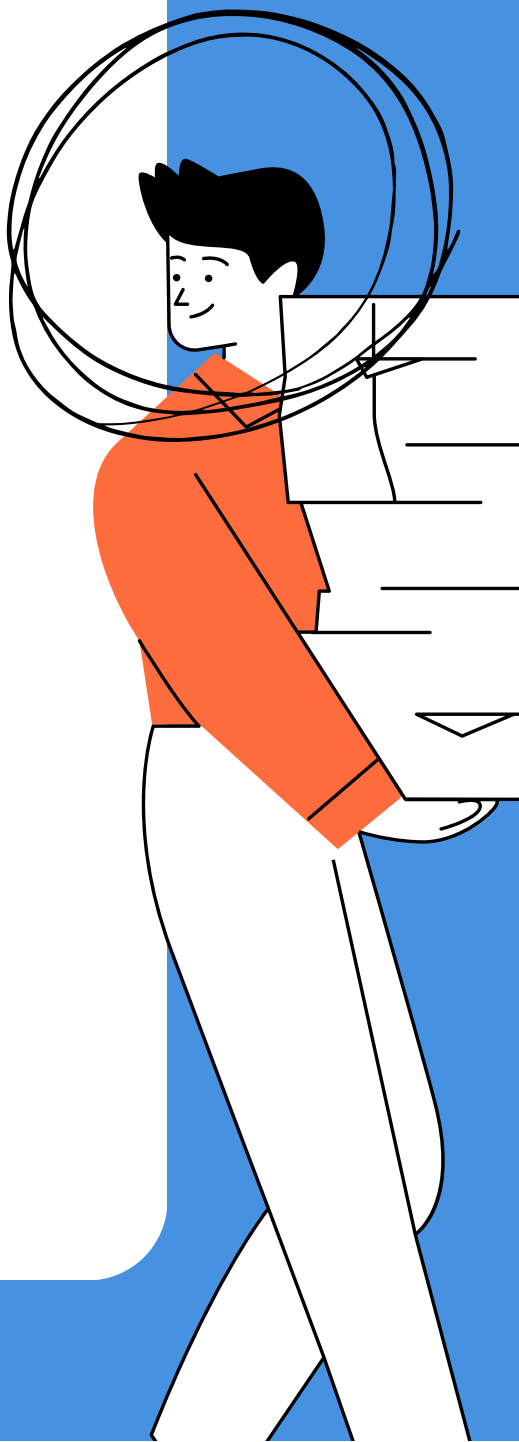
NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Summarize Restaurant Reviews

WHAT IS THE PROBLEM YOU'RE TRYING TO SOLVE?

There are alot of restaurant reviews that are in English which some considered very long, Therefore the Arab users who can't understand English find it difficult to read and understand .



WHY IS THE PROBLEM IMPORTANT TO SOLVE?

- Since there is no great interest in Arabic language, therefore, there are no attempts to translate other languages into it.
- There are some Arab users faces a problem because they need to learn English.
- Some customers may write comments containing colloquial ,slang, and specialized terms that hard to understand for someone who is not fluent in English

WHAT ARE THE MAIN RESULTS?

The system will generate a single Arabic summary of a restaurant's reviews which are written in English

Dataset details :

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

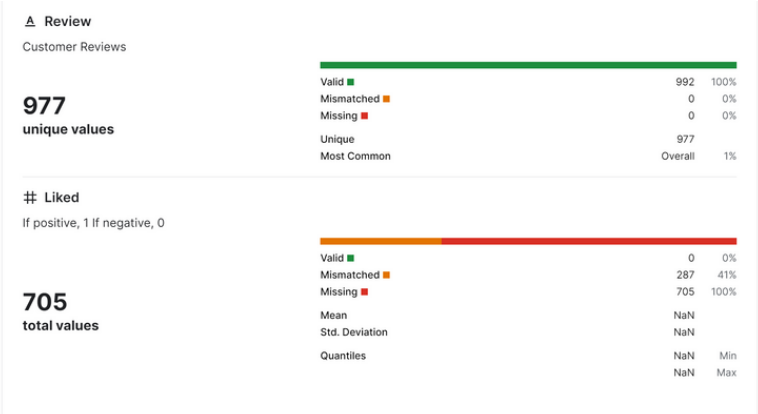
Dataset details

The dataset posted on kaggle for the purpose of applying machine learning to it

1000

Consumer reviews on Restorant

There are two partitions and 1000 rows in the dataset. Customer ratings are listed in the review column, and 0's and 1's are found in the like column. 1 if the rating is positive, 0 if it is negative. As the data size (61.33 kB).
Customer feedback review is also shown in the image.



Review
Customer Reviews

977
unique values

Valid	992	100%
Mismatched	0	0%
Missing	0	0%
Unique	977	
Most Common	Overall	1%

Liked
If positive, 1 If negative, 0

705
total values

Valid	0	0%
Mismatched	287	41%
Missing	705	100%
Mean	NaN	
Std. Deviation	NaN	
Quantiles	NaN	Min
	NaN	Max

Code, Outputs :

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Code, Outputs

1-Removing punctuation,Standardisation

```
def remove_punct(text): #Function for removing punct
  text_nopunct ="".join([char for char in text if char not in string.punctuation])
  return text_nopunct\

Reviews['Review_nopunc']= Reviews['Review'].apply(lambda x: remove_punct(x.lower()))
#Reviews.head(10)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Review  1000 non-null   object
 1   Liked   1000 non-null   int64
dtypes: int64(1), object(1)
memory usage: 15.8+ KB
```

	Review	Liked	Review_nopunc
0	Wow... Loved this place.	1	wow loved this place
1	Crust is not good.	0	crust is not good
2	Not tasty and the texture was just nasty.	0	not tasty and the texture was just nasty
3	Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1	stopped by during the late may bank holiday off rick steve recommendation and loved it
4	The selection on the menu was great and so were the prices.	1	the selection on the menu was great and so were the prices
5	Now I am getting angry and I want my damn pho.	0	now i am getting angry and i want my damn pho
6	Honestly it didn't taste THAT fresh.)	0	honestly it didnt taste that fresh
7	The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer.	0	the potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer
8	The fries were great too.	1	the fries were great too
9	A great touch.	1	a great touch

2- Tokenization using: 1- regex and split 2- NLTK

```
#  Code2:
nltk.download('punkt')

def tokenize(text):
 from nltk import word_tokenize,sent_tokenize
 token = word_tokenize(text)
 return token

Reviews['Review_nopunc_tokenize']= Reviews['Review_nopunc'].apply(lambda x:tokenize(x) )

#  Code3:
#tokenization using regex & split
import re
Reviews['Review_nopunc_token_RE']= Reviews['Review_nopunc'].apply(lambda x: re.split('\W+', x))
```

Review	Liked	Review_nopunc	Review_nopunc_tokenize	Review_nopunc_token_RE
Wow... Loved this place.	1	wow loved this place	[wow, loved, this, place]	[wow, loved, this, place]
Crust is not good.	0	crust is not good	[crust, is, not, good]	[crust, is, not, good]
Not tasty and the texture was just nasty.	0	not tasty and the texture was just nasty	[not, tasty, and, the, texture, was, just, nasty]	[not, tasty, and, the, texture, was, just, nasty]
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1	stopped by during the late may bank holiday off rick steve recommendation and loved it	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]
The selection on the menu was great and so were the prices.	1	the selection on the menu was great and so were the prices	[the, selection, on, the, menu, was, great, and, so, were, the, prices]	[the, selection, on, the, menu, was, great, and, so, were, the, prices]
Now I am getting angry and I want my	0	now i am getting angry and i want my damn pho	[now, i, am, getting, angry, and, i, want, my, damn, pho]	[now, i, am, getting, angry, and, i, want, my, damn, pho]

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Code, Outputs

3- Removing Stop- words using NLTK

```
#Removing stopwords using nltk
nltk.download('stopwords')

def remove_stop(text):
  text_nostop=[char for char in text if char not in nltk.corpus.stopwords.words('english')]
  return text_nostop

Reviews['Reviews_text_nostopword']=Reviews['Review_nopunc_tokenize'].apply(lambda x: remove_stop(x))
```

Review	Liked	Review_nopunc	Review_nopunc_tokenize	Review_nopunc_token_RE	Reviews_text_nostopword
Wow... Loved this place.	1	wow loved this place	[wow, loved, this, place]	[wow, loved, this, place]	[wow, loved, place]
Crust is not good.	0	crust is not good	[crust, is, not, good]	[crust, is, not, good]	[crust, good]
Not tasty and the texture was just nasty.	0	not tasty and the texture was just nasty	[not, tasty, and, the, texture, was, just, nasty]	[not, tasty, and, the, texture, was, just, nasty]	[tasty, texture, nasty]
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1	stopped by during the late may bank holiday off rick steve recommendation and loved it	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]	[stopped, late, may, bank, holiday, rick, steve, recommendation, loved]
The selection on the menu was great and so were the prices.	1	the selection on the menu was great and so were the prices	[the, selection, on, the, menu, was, great, and, so, were, the, prices]	[the, selection, on, the, menu, was, great, and, so, were, the, prices]	[selection, menu, great, prices]
Now I am getting angry and I want my	0	now i am getting angry and i want my damn pho	[now, i, am, getting, angry, and, i, want, my, damn, pho]	[now, i, am, getting, angry, and, i, want, my, damn, pho]	[getting, angry, want, damn, pho]

4- Stemming

```
#Porter Stemming tokenized list
ps=nltk.PorterStemmer()

def stemming(sentence):
  text_stem=[ps.stem(word) for word in sentence]
  return text_stem

Reviews['Reviews_text_stemmed']=Reviews['Review_nopunc_tokenize'].apply(lambda x: stemming(x))
```

Review	Liked	Review_nopunc	Review_nopunc_tokenize	Review_nopunc_token_RE	Reviews_text_nostopword	Reviews_text_stemmed
Wow... Loved this place.	1	wow loved this place	[wow, loved, this, place]	[wow, loved, this, place]	[wow, loved, place]	[wow, love, thi, place]
Crust is not good.	0	crust is not good	[crust, is, not, good]	[crust, is, not, good]	[crust, good]	[crust, is, not, good]
Not tasty and the texture was just nasty.	0	not tasty and the texture was just nasty	[not, tasty, and, the, texture, was, just, nasty]	[not, tasty, and, the, texture, was, just, nasty]	[tasty, texture, nasty]	[not, tasti, and, the, textur, wa, just, nasti]
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1	stopped by during the late may bank holiday off rick steve recommendation and loved it	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]	[stopped, late, may, bank, holiday, rick, steve, recommendation, loved]	[stop, by, dure, the, late, may, bank, holiday, off, rick, steve, recommend, and, love, it]
The selection on the menu was great and so were the prices.	1	the selection on the menu was great and so were the prices	[the, selection, on, the, menu, was, great, and, so, were, the, prices]	[the, selection, on, the, menu, was, great, and, so, were, the, prices]	[selection, menu, great, prices]	[the, select, on, the, menu, wa, great, and, so, were, the, price]
Now I am getting angry and I want my	0	now i am getting angry and i want my damn pho	[now, i, am, getting, angry, and, i, want, my, damn, pho]	[now, i, am, getting, angry, and, i, want, my, damn, pho]	[getting, angry, want, damn, pho]	[now, i, am, get, angri, and, i, want, my, damn, pho]

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Code, Outputs

5- Lemmatization

```
#Lemmatization of tokenized list
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer
wn = WordNetLemmatizer()

def lemmatizeWords(sentence):
  text_lem=[wn.lemmatize(word) for word in sentence]
  return text_lem

Reviews['Reviews_text_Lemmatized']=Reviews['Review_nopunc_tokenize'].apply(lambda x: lemmatizeWords(x))
```

Review	Liked	Review_nopunc	Review_nopunc_tokenize	Review_nopunc_token_RE	Reviews_text_nostopword	Reviews_text_stemmed	Reviews_text_Lemmatized
... Loved is place.	1	wow loved this place	[wow, loved, this, place]	[wow, loved, this, place]	[wow, loved, place]	[wow, love, thi, place]	[wow, loved, this, place]
ust is not good.	0	crust is not good	[crust, is, not, good]	[crust, is, not, good]	[crust, good]	[crust, is, not, good]	[crust, is, not, good]
asty and ture was st nasty.	0	not tasty and the texture was just nasty	[not, tasty, and, the, texture, was, just, nasty]	[not, tasty, and, the, texture, was, just, nasty]	[tasty, texture, nasty]	[not, tasti, and, the, textur, wa, just, nasti]	[not, tasty, and, the, texture, wa, just, nasty]
opped by the late lay bank off Rick Steve endation loved it.	1	stopped by during the late may bank holiday off rick steve recommendation and loved it	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]	[stopped, late, may, bank, holiday, rick, steve, recommendation, loved]	[stop, by, dure, the, late, may, bank, holiday, off, rick, steve, recommend, and, love, it]	[stopped, by, during, the, late, may, bank, holiday, off, rick, steve, recommendation, and, loved, it]
ction on enu was t and so e prices.	1	the selection on the menu was great and so were the prices	[the, selection, on, the, menu, was, great, and, so, were, the, prices]	[the, selection, on, the, menu, was, great, and, so, were, the, prices]	[selection, menu, great, prices]	[the, select, on, the, menu, wa, great, and, so, were, the, price]	[the, selection, on, the, menu, wa, great, and, so, were, the, price]
low I am g angry want my	0	now i am getting angry and i want my damn pho	[now, i, am, getting, angry, and, i, want, my, damn, pho]	[now, i, am, getting, angry, and, i, want, my, damn, pho]	[getting, angry, want, damn, pho]	[now, i, am, get, angri, and, i, want, my, damn, pho]	[now, i, am, getting, angry, and, i, want, my, damn, pho]

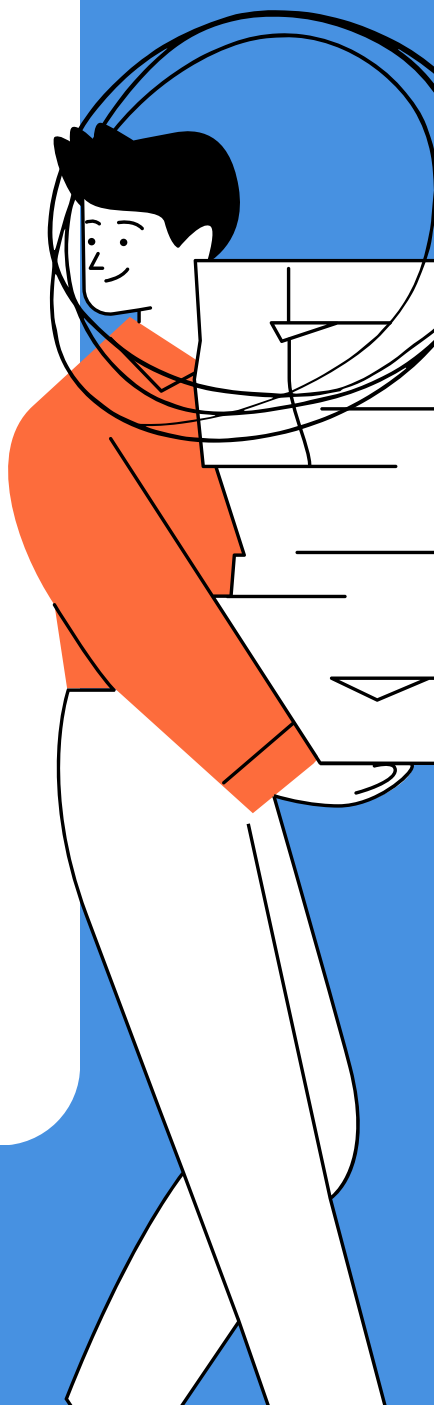
Chapter 2 :

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Project main objective

Our program aims to shorten the time it takes customers to decide whether to visit a restaurant by providing a summary of three categories: food, service, and price. It also provides statistical graphics showing customers' opinions about the three categories. Additionally, it provides a service for translating a summary of comments into Arabic in case the readers do not know English.



Target user:

- Individuals interested in dining out.
- Customers looking to make informed decisions about restaurants in a time-efficient manner.
- Individuals with limited English proficiency who require Arabic translation of the comments and summary.

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Project methodology

Steps :

- Clean the text of the dataset (by removing punctuation, stop words, and tokenizes them).
- Summarize the Clean text.
- Translate the summary to Arabic.
- Classify them as positive, negative, and neutral in the Sentiment column
- Create new columns for price, service, and food.
- We calculate the percentage of positive and negative comments for the three categories.
- Create an ML model by using random forest classifier and Sentiment as a feature

Libraries :

- pandas
- os
- nltk
- string
- nltk.stem
- sklearn.ensemble
- sklearn.model_selection
- sumy.parsers.plaintext
- sumy.nlp.tokenizers
- sumy.summarizers.lsa
- textblob
- deep_translator

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Project methodology (NLP steps and libraries) & Screenshot for each NLP steps with some explanation

1- Creates a Pandas DataFrame of our dataset then we created a new column called "CleanData" that removes punctuation marks, stop words (excluding negatives), and tokenizes text from the "Review" column.

Used Libraries :

-string

-nltk (corpus.stopwords &word_tokenize)

```
# Load the dataset
df = pd.read_csv("/Users/areejbaw/Library/CloudStorage/OneDrive-UMMAL-QURAUNIVERSITY/Desktop/معالجات/project/Restaurant_Review.tsv",
```

```
def remove_punctuation(text):

  text_nopunct ="".join([char for char in text if char not in string.punctuation])
  return text_nopunct


def remove_stopWords(text):
  nltk.download('stopwords')

  negative_words = ["no", "not", "don't", "aren't", "couldn't", "didn't", "doesn't",
                    "hadn't", "hasn't", "haven't", "isn't", "mightn't", "mustn't", "needn't", "shan't", "shouldn't",
                    "wasn't", "weren't", "won't", "wouldn't"]
  stop_words = set(nltk.corpus.stopwords.words('english'))
  stop_words = stop_words.difference(set(negative_words))
  text_nostop=[char for char in text if char not in stop_words ]

  return text_nostop

nltk.download('punkt')

def tokenize(text):
 token = word_tokenize(text)
 return token
```

```
df['CleanData']=df['Review'].apply(lambda x: remove_punctuation(x))

df['CleanData']=df['CleanData'].apply(lambda x:tokenize(x))

df['CleanData']=df['CleanData'].apply(lambda x: remove_stopWords(x))
```

Project methodology (NLP steps and libraries) & Screenshot for each NLP steps with some explanation

2- We created a new column called "Summary" that summarizes the text from the "CleanData" column using LSA summarize.

Used Libraries :
-sumy.parsers.plaintext (PlaintextParser)
sumy.summarizers.lsa (LsaSummarizer)
sumy.nlp.tokenizers (Tokenizer)

```
def summarize_text(text, num_sentences=1):
    # Create a plaintext parser
    parser = PlaintextParser.from_string(text, Tokenizer("english"))

    # Create an LSA summarizer
    summarizer = LsaSummarizer()

    # Summarize the document with the specified number of sentences
    summary = summarizer(parser.document, num_sentences)

    # Combine the sentences into a single string and return it
    return " ".join(str(sentence) for sentence in summary)

df["Summary"] = df['CleanData'].apply(lambda x: summarize_text(" ".join([summarize_text(word,1) for word in x])))
```

3- WE CREATED A NEW COLUMN CALLED "ARABIC" THAT TRANSLATES THE TEXT FROM THE "SUMMARY" COLUMN INTO ARABIC USING THE DEEP_TRANSLATOR LIBRARIES.
USED LIBRARIES :

DEEP_TRANSLATOR (GOOGLETRANSLATOR)

```
df["Arabic"] = df.iloc[:10,:]['Summary'].apply(lambda x: GoogleTranslator(source='en', target='ar').translate((x)))
```

Project methodology (NLP steps and libraries) & Screenshot for each NLP steps with some explanation

4- We created a new column named "Sentiment" and it contains 3 values, either positive, neutral, or negative, and the value is determined based on the score of the text, and the score is calculated using textblob libraries, and it has a method that calculates the score for each sentence and classifies it

Used Libraries :
textblob(TextBlob)

```
def analyze_sentiment(text):
    blob = TextBlob(text)
    sentiment_score = blob.sentiment.polarity

    # Return 'positive' if sentiment score is greater than 0, 'negative' if it's less than 0, and 'neutral' if it's 0
    if sentiment_score > 0:
        return 'positive'
    elif sentiment_score < 0:
        return 'negative'
    else:
        return 'neutral'


df['Sentiment'] = df['Review'].apply(lambda x : analyze_sentiment(x))
```

5- WE CREATED A NEW COLUMN NAMED "ROOT" THAT TAKES THE TEXT IN THE "CLEANDATA" COLUMN AND RETURNS EACH WORD TO ITS ROOT, IN ORDER TO SEARCH FOR SPECIFIC WORDS THAT ARE PRESENT IN THE TEXT

USED LIBRARIES :
NLTK.STEM(PORTERSTEMMER)

```
# Create a column for the root of each word in the reviews
ps = PorterStemmer()
df["root"] = df["CleanData"].apply(lambda x: " ".join([ps.stem(word) for word in x]))
```

Project methodology (NLP steps and libraries) & Screenshot for each NLP steps with some explanation

6- We created 3 new columns named "price", "service", "food" in order to store any description of one of the three categories by searching in Set that consider a word after stem

```
# Create new columns for price, service, and food
df["price"] = ""
df["service"] = ""
df["food"] = ""


def extract_descriptions(row):
  # Define some keywords for each category



  price_keywords = ["afford", "budget-friendli", "good valu", "worth the price", "within budget", "within reach", "within me
  service_keywords = ["great", "friend", "help", "welcom", "kind", "gentl", "respect", "smile", "accommod", "knowledg", "po
  food_keywords = ["great", "delici", "tasti", "flavor", "fresh", "yummi", "appet", "cook", "crisp", "crunchi", "juici", "me
 # Split the root column into a list of words
  words = row["root"].split()

  # Loop through the words and check if they match any keyword
  for word in words:
    if word in price_keywords:
      row["price"] += word + " "
    elif word in service_keywords:
      row["service"] += word + " "
    elif word in food_keywords:
      row["food"] += word + " "

  # Return the row with the updated columns
  return row

# Apply the function to the dataframe
df = df.apply(extract_descriptions, axis=1)
```

Project methodology (NLP steps and libraries) & Screenshot for each NLP steps with some explanation

7- We calculated the percentage of positive and negative for each of the three categories

Used Libraries :
pandas

```
positiveComments=""
negativeComments=""
def calculate_sentiment_percentage(df, column):

  # Create variables to store the number of positive and negative comments
  positive_count = 0
  negative_count = 0


  for index, row in df.iterrows():

      if row[column] and pd.notna(row[column]):

        if row['Sentiment'] == 'positive':
            positive_count += 1
        elif row['Sentiment'] == 'negative':
            negative_count += 1


  total_count = positive_count + negative_count
  positive_percentage = (positive_count / total_count )* 100

  negative_percentage = (negative_count / total_count) * 100

  positiveComments = "The positive comments about the " + column + " was: ", round(positive_percentage,3), "%"
  negativeComments= "The negative comments about the " + column + " was: ", round(negative_percentage,3), "%"
  return str(positiveComments)+"\n"+ str(negativeComments),positive_percentage,negative_percentage
```

Project methodology (NLP steps and libraries) & Screenshot for each NLP steps with some explanation

8- We created a machine learning module using the random forest classifier, and the feature used here is the "Sentiment" column and print the accuracy.

Used Libraries :
sklearn.model_selection(train_test_split)
sklearn.ensemble (RandomForestClassifier) pandas

```
# vectorizer = CountVectorizer()

X = pd.get_dummies(df['Sentiment'])
# X = pd.concat([pd.DataFrame(X_sentiment)], axis=1)
# X.columns = X.columns.astype(str) #
target = df['Liked']

X_train, X_test, y_train, y_test = train_test_split(X, target, test_size=0.2)

# Train a random forest classifier on the training set
rfc = RandomForestClassifier(n_jobs=-1)
rfc.fit(X_train, y_train)

# Evaluate the classifier on the testing set
score = rfc.score(X_test, y_test)
print('Accuracy:', score)
```

Project methodology (NLP steps and libraries) & Screenshot for each NLP steps with some explanation

9- We created interfaces that show the acronym for the three categories in both Arabic and English, and the user can also see the statistical graphics for the three categories.

Used Libraries :
tkinter
matplotlib.pyplot
matplotlib.backends.backend_agg(FigureCanvasAgg)
PIL(Image & ImageTk)

```
import tkinter as tk
import matplotlib.pyplot as plt
from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
from PIL import Image, ImageTk


def show_pie_chart(title,positive_percentage,negative_percentage):
    new_window = tk.Toplevel(window)


    labels = ['Positive', 'Negative']
    sizes = [positive_percentage, negative_percentage]
    colors = ['#E8A09A', '#FBE29F']
    fig, ax = plt.subplots()
    ax.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
    ax.set_title(title)
    canvas = FigureCanvas(fig)
    canvas.draw()
    pil_image = Image.frombytes('RGB', canvas.get_width_height(), canvas.tostring_rgb())
    fig_image = ImageTk.PhotoImage(master=new_window, image=pil_image)

    chart_label = tk.Label(new_window, image=fig_image)
    chart_label.image = fig_image
    chart_label.pack()


window = tk.Tk()
window.geometry("600x600")
window.title("Sentiment Analysis")

label = tk.Label(window, text=tee)
label.pack()

food_button = tk.Button(window, text="Food", command=lambda: show_pie_chart("Food",positive_percentagefo,negative_percentagefo))
food_button.pack()
service_button = tk.Button(window, text="Service", command=lambda: show_pie_chart("Service",positive_percentageSe,negative_percentageSe))
service_button.pack()
price_button = tk.Button(window, text="Price", command=lambda: show_pie_chart("Price",positive_percentagePr,negative_percentagePr))
price_button.pack()

window.mainloop()
```

Project methodology (NLP steps and libraries) & Screenshot for each NLP steps with some explanation

10- We saved the complete data set after analysis and translation into an excel file and print the path of the file .

Used Libraries :
os

```
df.to_csv('table.txt', index=False)
df.to_excel("table.xlsx")
print(os.path.join(os.getcwd(), 'table.txt'))
```

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Output :

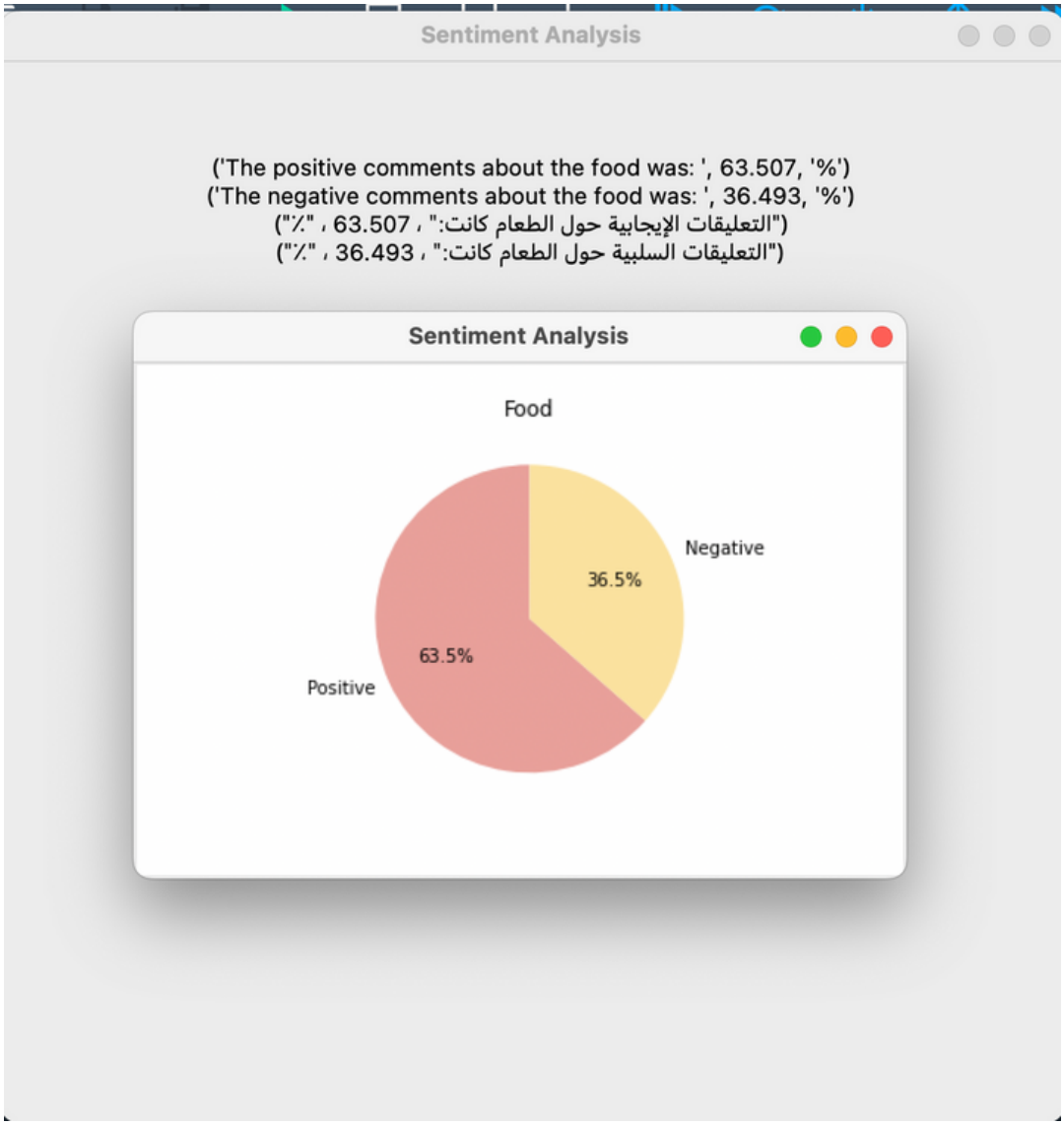
```
Accuracy: 0.815
```

	B	C	D	E	F	G	H	I	J	K
1	Review	Liked	CleanData	Summary	Arabic	Sentiment	root	price	service	food
2	Wow... Loved this place.	1	['Wow', 'Loved', 'place']	Wow Loved place	واو مكان محبوب	positive	wow love place			
3	Crust is not good.	0	['Crust', 'not', 'good']	Crust not good	القشرة ليست جيدة	negative	crust not good			
4	Not tasty and the texture was just nasty.	0	['Not', 'tasty', 'texture', 'nasty']	Not tasty texture nasty	ليس الملمس لذيذاً مقرفاً	negative	not tasti textur nasti			tasti nasti
5	Stopped by during the late May bank holiday off R	1	['Stopped', 'late', 'May', 'bank', 'holiday', 'Rick', 'S	Stopped late May bank holiday Ri	توقفت في أواخر مايو عطلة البنك أحب ريك	positive	stop late may bank holiday rick steve recommend love			recommend
6	The selection on the menu was great and so were	1	['The', 'selection', 'menu', 'great', 'prices']	The selection menu great prices	قائمة الاختيار أسعار رائعة	positive	the select menu great price		great	
7	Now I am getting angry and I want my damn pho.	0	['Now', 'I', 'getting', 'angry', 'I', 'want', 'damn', 'pho	Now getting angry want damn ph	الآن تغضب تريد pho لعنة	negative	now i get angri i want damn pho			
8	Honesity it didn't taste THAT fresh.)	0	['Honesity', 'didnt', 'taste', 'THAT', 'fresh']	Honesity didnt taste fresh	لم يكن طعم Honesity طازج	positive	honestl didnt tast that fresh			fresh
9	The potatoes were like rubber and you could tell th	0	['The', 'potatoes', 'like', 'rubber', 'could', 'tell', 'mac	The potatoes like rubber could tell	يمكن أن تدل البطاطس مثل المطاط الوقت	neutral	the potato like rubber could tell made ahead time kept warmer		time	
10	The fries were great too.	1	['The', 'fries', 'great']	The fries great	البطاطا المقلية كبيرة	positive	the fri great		great	
11	A great touch.	1	['A', 'great', 'touch']	great touch	لمسة رائعة	positive	a great touch		great	
12	Service was very prompt.	1	['Service', 'prompt']	Service prompt		positive	servic prompt		prompt	
13	Would not go back.	0	['Would', 'not', 'go', 'back']	Would not go back		neutral	would not go back			
14	The cashier had no care what so ever on what I h	0	['The', 'cashier', 'no', 'care', 'ever', 'I', 'say', 'still',	The cashier no care ever say still ended wayyy overpriced		neutral	the cashier no care ever i say still end wayyy overpr			
15	I tried the Cape Cod ravoli, chicken, with cranberr	1	['I', 'tried', 'Cape', 'Cod', 'ravoli', 'chicken', 'cranbe	tried Cape Cod ravoli chicken cranberrymmmm		negative	i tri cape cod ravoli chicken cranberrymmmm			
16	I was disgusted because I was pretty sure that wa	0	['I', 'disgusted', 'I', 'pretty', 'sure', 'human', 'hair']	disgusted pretty sure human hair		negative	i disgust i pretti sure human hair			disgust
17	I was shocked because no signs indicate cash on	0	['I', 'shocked', 'no', 'signs', 'indicate', 'cash']	shocked no signs indicate cash		negative	i shock no sign indic cash			
18	Highly recommended.	1	['Highly', 'recommended']	Highly recommended		positive	highli recommend			recommend
19	Waitress was a little slow in service.	0	['Waitress', 'little', 'slow', 'service']	Waitress little slow service		negative	waitress littl slow servic		slow	
20	This place is not worth your time, let alone Vegas.	0	['This', 'place', 'not', 'worth', 'time', 'let', 'alone', 'Ve	This place not worth time let alone Vegas		negative	thi place not worth time let alon vega		time	
21	did not like at all.	0	['not', 'like']	not like		neutral	not like			
22	The Burritos Blah!	0	['The', 'Burritos', 'Blah']	The Burritos Blah		neutral	the burritto blah			
23	The food, amazing.	1	['The', 'food', 'amazing']	The food amazing		positive	the food amaz			amaz
24	Service is also cute.	1	['Service', 'also', 'cute']	Service also cute		positive	servic also cute			
25	I could care less... The interior is just beautiful.	1	['I', 'could', 'care', 'less', 'The', 'interior', 'beautiful'	could care less The interior beautiful		positive	i could care less the interior beauti			beauti
26	So they performed.	1	['So', 'performed']	So performed		neutral	so perform			
27	That's right....the red velvet cake.....ohhh this stuff	1	['Thats', 'righthe', 'red', 'velvet', 'cakeohhh', 'stuf	Thats righthe red velvet cakeohhh stuff good		positive	that righth red velvet cakeohhh stuff good			
28	- They never brought a salad we asked for.	0	['They', 'never', 'brought', 'salad', 'asked']	They never brought salad asked		neutral	they never brought salad ask			
29	This hole in the wall has great Mexican street tacc	1	['This', 'hole', 'wall', 'great', 'Mexican', 'street', 'tac	This hole wall great Mexican street tacos friendly staff		positive	thi hole wall great mexican street taco friendli staff		great	

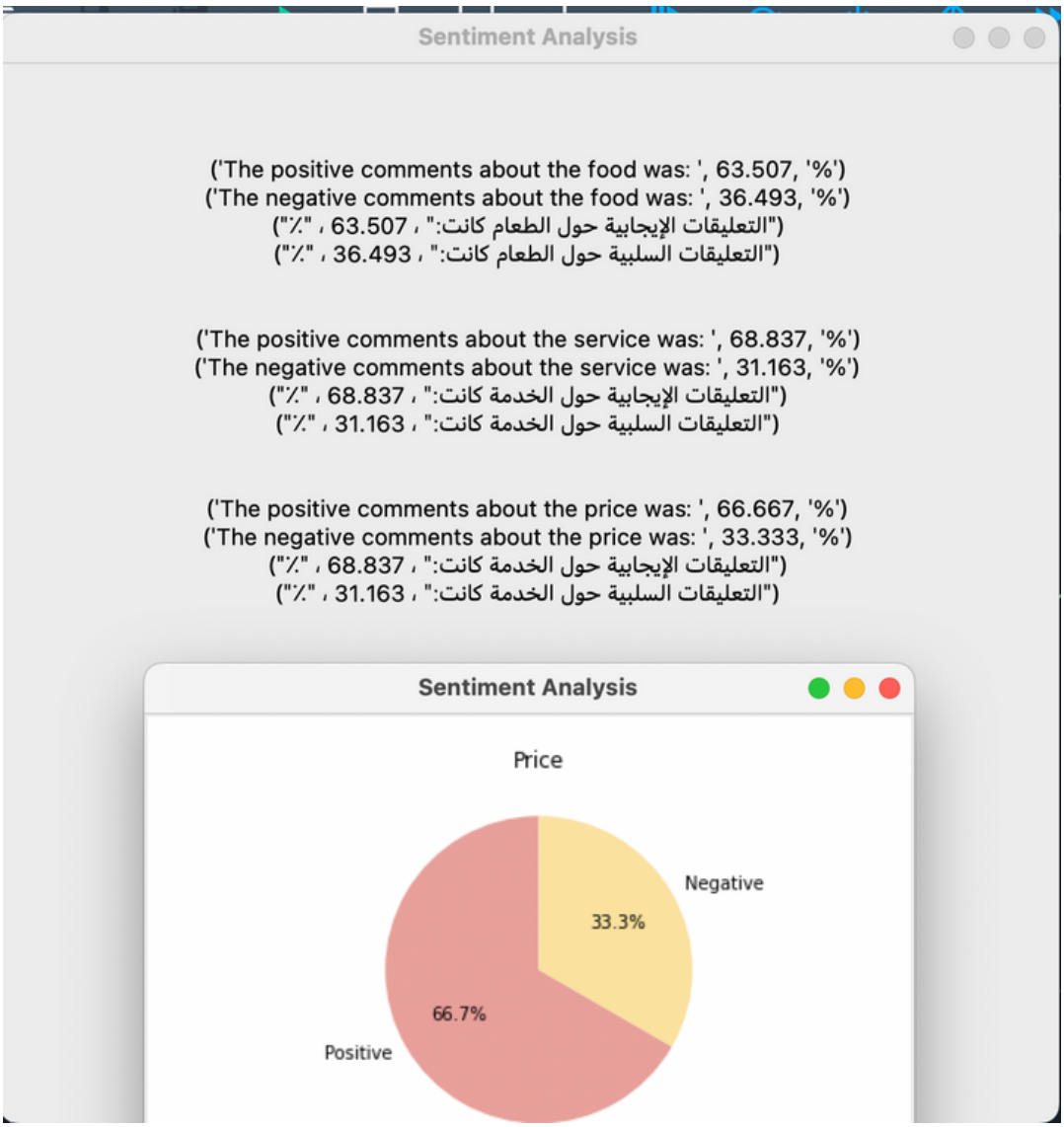
Output :

Sentiment Analysis
('The positive comments about the food was: ', 63.507, '%')
('The negative comments about the food was: ', 36.493, '%')
("التعليقات الإيجابية حول الطعام كانت:" ، 63.507 ، "٪")
("التعليقات السلبية حول الطعام كانت:" ، 36.493 ، "٪")
('The positive comments about the service was: ', 68.837, '%')
('The negative comments about the service was: ', 31.163, '%')
("التعليقات الإيجابية حول الخدمة كانت:" ، 68.837 ، "٪")
("التعليقات السلبية حول الخدمة كانت:" ، 31.163 ، "٪")
('The positive comments about the price was: ', 66.667, '%')
('The negative comments about the price was: ', 33.333, '%')
("التعليقات الإيجابية حول الخدمة كانت:" ، 68.837 ، "٪")
("التعليقات السلبية حول الخدمة كانت:" ، 31.163 ، "٪")
For more details about any of them, click on the following buttons:
Food
Service
Price

Output :


Sentiment Analysis
('The positive comments about the food was: ', 63.507, '%')
('The negative comments about the food was: ', 36.493, '%')
("التعليقات الإيجابية حول الطعام كانت:" ، 63.507 ، "٪")
("التعليقات السلبية حول الطعام كانت:" ، 36.493 ، "٪")
Sentiment Analysis
Food
Negative
36.5%
63.5%
Positive

Output :


Sentiment Analysis
('The positive comments about the food was: ', 63.507, '%')
('The negative comments about the food was: ', 36.493, '%')
("التعليقات الإيجابية حول الطعام كانت:" ، 63.507 ، "٪")
("التعليقات السلبية حول الطعام كانت:" ، 36.493 ، "٪")
('The positive comments about the service was: ', 68.837, '%')
('The negative comments about the service was: ', 31.163, '%')
("التعليقات الإيجابية حول الخدمة كانت:" ، 68.837 ، "٪")
("التعليقات السلبية حول الخدمة كانت:" ، 31.163 ، "٪")
('The positive comments about the price was: ', 66.667, '%')
('The negative comments about the price was: ', 33.333, '%')
("التعليقات الإيجابية حول الخدمة كانت:" ، 68.837 ، "٪")
("التعليقات السلبية حول الخدمة كانت:" ، 31.163 ، "٪")
Sentiment Analysis
Price
Negative
33.3%
66.7%
Positive

Task distribution

NATURAL LANGUAGE PROCESSING PROJECT

DR. ASHWAG MAGRHABY

Tasks	KHULOOD	AMAL	HALA	AREEJ	NAJLA
Code	✔	✔	✔	✔	✔
Report	✔	✔	✔	✔	✔