```
from IPython.display import HTML
```

Sentiment Analysis on Restaurant Reviews

-

The second feature is Liked which contains boolean-like information about the review or comment that customer may given to a restaurant ... Generally we can understand as positive or negative feedback or something else

Importing the libraries

Library Loading

```
# Importing essential Libraries

# NumPy is used for numerical computations and working with arrays.
import numpy as np

# Pandas is used for data manipulation and analysis.
import pandas as pd
```

Exploratory Data Analysis (EDA)

Data loading is the process of importing or reading datasets from various sources, such as files (e.g., CSV, Excel), databases, or APIs, into a data analysis environment or software for further processing and analysis.

```
df = 
pd.read_csv("/kaggle/input/restaurant-reviews/Restaurant_Reviews.tsv",
delimiter='\t',quoting=3)
```

Descriptive Data Analysis is the process of using statistical and visual techniques to summarize and present key features, patterns, and insights from a dataset, typically involving measures like means, medians, standard deviations, histograms, and scatter plots to provide an overview of the data's characteristics.

```
df.shape

(1000, 2)

df.columns

Index(['Review', 'Liked'], dtype='object')
```

```python
df.head()
```

```
                                              Review  Liked
0                           Wow... Loved this place.      1
1                                 Crust is not good.      0
2          Not tasty and the texture was just nasty.      0
3  Stopped by during the late May bank holiday of...      1
4  The selection on the menu was great and so wer...      1
```

```python
df.tail()
```

```
                                                Review  Liked
995  I think food should have flavor and texture an...      0
996                            Appetite instantly gone.      0
997  Overall I was not impressed and would not go b...      0
998  The whole experience was underwhelming, and I ...      0
999  Then, as if I hadn't wasted enough of my life ...      0
```

```python
df.sample(5)
```

```
                                                Review  Liked
830  Do yourself a favor and stay away from this dish.      0
877                             Go To Place for Gyros.      1
949  Maybe it's just their Vegetarian fare, but I'v...      0
555  I know this is not like the other restaurants ...      0
950          It wasn't busy at all and now we know why.      0
```

```python
df.info()
```

```
<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
```

```python
df['Liked'].value_counts()
```

```
Liked
1    500
0    500
Name: count, dtype: int64
```

```python
df.describe()
```

```
            Liked
count  1000.00000
mean      0.50000
std       0.50025
```

```
min      0.00000
25%      0.00000
50%      0.50000
75%      1.00000
max      1.00000
```

```python
df.isnull().sum()
```

```
Review    0
Liked     0
dtype: int64
```

```python
df.duplicated().sum()
```

```
4
```

Feature Engineering

Feature engineering is the process of creating new, meaningful, and informative features (variables) from existing data or transforming existing features to improve the performance of machine learning models and enhance their ability to make accurate predictions or classifications.

```python
df['Length'] = df['Review'].apply(len)
df.head(5)
```

	Review	Liked	Length
0	Wow... Loved this place.	1	24
1	Crust is not good.	0	18
2	Not tasty and the texture was just nasty.	0	41
3	Stopped by during the late May bank holiday of...	1	87
4	The selection on the menu was great and so wer...	1	59

Data Preprocessing

Feature engineering is the process of creating, selecting, or transforming features (variables) in a dataset to improve the performance of machine learning models by making them more informative and relevant for the specific task at hand, such as predictive modeling or classification.

```python
import nltk
import re

# Download NLTK stopwords data
nltk.download('stopwords')
```

```python
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
```

```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```python
print(list(stopwords.words('english')))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
"you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself',
'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her',
'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them',
'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom',
'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was',
'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do',
'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or',
'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with',
'about', 'against', 'between', 'into', 'through', 'during', 'before',
'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',
'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once',
'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor',
'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't',
'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now',
'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't",
'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn',
"hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma',
'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan',
"shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren',
"weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

```python
# Cleaning the Reviews and Creating a Corpus

# The purpose of this code is to clean a collection of text reviews
and create a corpus of cleaned reviews.
# The corpus is a list where each element is a cleaned version of a
review.

# Initialize an empty list to store the cleaned reviews.
corpus = []

# Loop through the first 1000 reviews (adjust the range as needed) in
the DataFrame 'df'.
for i in range(0, 1000):

    # Step 1: Removing non-alphabetical characters
    # Using regular expressions, substitute any characters that are
not in the range 'a' to 'z' with a space.
    review = re.sub(pattern='[^a-zA-Z]', repl=' ', string=df['Review']
[i])
```

```
    # Step 2: Converting text to lowercase
    # Convert the entire review to lowercase to ensure consistency.
    review = review.lower()

    # Step 3: Tokenization
    # Split the review into individual words.
    review_words = review.split()

    # Step 4: Removing Stop Words
    # Remove common English stop words (e.g., 'the', 'and', 'in') from
the list of words.
    # Stop words do not carry significant meaning for text analysis.
    review_words = [word for word in review_words if not word in
set(stopwords.words('english'))]

    # Step 5: Stemming
    # Apply stemming to reduce words to their root form (e.g.,
'running' becomes 'run').
    # This helps in reducing the dimensionality of the text data.
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review_words]

    # Step 6: Rejoining Tokens
    # Join the cleaned and stemmed words back into a single string.
    review = ' '.join(review)

    # Step 7: Append to Corpus
    # Add the cleaned and processed review to the corpus list.
    corpus.append(review)

# After running this code, 'corpus' will contain a list of cleaned
reviews suitable for further text analysis.
```

<!DOCTYPE html> Explaining Data Cleaning and Corpus Creation body { background-color: lightblue; padding: 20px; } Cleaning the Reviews and Creating a Corpus

The provided code snippet exemplifies a crucial phase in text data preprocessing, where raw text reviews undergo a series of transformations to create a cleaned and structured corpus. This corpus is a fundamental resource for various natural language processing (NLP) and text analysis tasks.

Code Purpose:

The primary objective of this code is to prepare textual data for subsequent analysis by eliminating extraneous information, standardizing text, and breaking it down into manageable components. Let's delve into the step-by-step explanation of what this code accomplishes:

1. Initialization:

To begin, an empty list named 'corpus' is initialized:

corpus = []

This list will serve as a container to store the cleaned reviews once they are processed.

2. Loop Through Reviews:

Next, the code initiates a loop to iterate through a collection of text reviews. In this specific instance, the loop focuses on the first 1000 reviews found within a DataFrame named 'df.' However, it's crucial to note that this range can be adjusted based on the specific dataset and requirements:

for i in range(0, 1000):

3. Data Cleaning Steps:

The heart of this code lies in the various data cleaning and preprocessing steps applied to each review. These steps ensure that the text data becomes suitable for subsequent analysis. Here are the key steps:

Step 1: Removing Non-Alphabetical Characters:

Within this step, the code employs regular expressions to identify and substitute any characters that do not fall within the 'a' to 'z' range with a space. The result is a review with non-alphabetical characters and special symbols effectively removed:

review = re.sub(pattern='[^a-zA-Z]', repl=' ', string=df['Review'][i])

Step 2: Converting Text to Lowercase:

Uniformity in text data is crucial, and to ensure this, the entire review is converted to lowercase:

review = review.lower()

Step 3: Tokenization:

Tokenization involves splitting the review into individual words, treating each word as a separate token:

review_words = review.split()

Step 4: Removing Stop Words:

Common English stop words, such as 'the,' 'and,' and 'in,' are often devoid of meaningful information in text analysis. Hence, the code filters out these stop words from the list of words:

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

Step 5: Stemming:

Stemming is employed to reduce words to their root form, helping to reduce the dimensionality of the text data. For example, 'running' becomes 'run.' Stemming aims to capture the core meaning of words:

ps = PorterStemmer() review = [ps.stem(word) for word in review_words]

Step 6: Rejoining Tokens:

After cleaning and stemming individual tokens, the code rejoins them into a single string, effectively reconstructing the review:

review = ' '.join(review)

Step 7: Append to Corpus:

Finally, the processed and cleaned review is appended to the 'corpus' list, ensuring that each element of the 'corpus' represents a cleaned review:

corpus.append(review)

Upon completion of this code, the 'corpus' will contain a structured list of cleaned reviews, which is ready for further text analysis, including tasks such as sentiment analysis, topic modeling, or document classification.

In essence, this code showcases the critical role of data cleaning and preprocessing in NLP and text analysis workflows. By transforming raw text into a clean and structured corpus, it lays the foundation for extracting valuable insights and patterns from textual data.

With this understanding, practitioners can apply similar techniques to process and analyze text data in various real-world scenarios.

```
corpus[:1500]
```

```
['wow love place',
 'crust good',
 'tasti textur nasti',
 'stop late may bank holiday rick steve recommend love',
 'select menu great price',
 'get angri want damn pho',
 'honeslti tast fresh',
 'potato like rubber could tell made ahead time kept warmer',
 'fri great',
 'great touch',
 'servic prompt',
 'would go back',
 'cashier care ever say still end wayyy overpr',
 'tri cape cod ravoli chicken cranberri mmmm',
 'disgust pretti sure human hair',
 'shock sign indic cash',
 'highli recommend',
 'waitress littl slow servic',
 'place worth time let alon vega',
 'like',
 'burritto blah',
 'food amaz',
 'servic also cute',
 'could care less interior beauti',
 'perform',
 'right red velvet cake ohhh stuff good',
```

```
 'never brought salad ask',
 'hole wall great mexican street taco friendli staff',
 'took hour get food tabl restaur food luke warm sever run around like
total overwhelm',
 'worst salmon sashimi',
 'also combo like burger fri beer decent deal',
 'like final blow',
 'found place accid could happier',
 'seem like good quick place grab bite familiar pub food favor look
elsewher',
 'overal like place lot',
 'redeem qualiti restaur inexpens',
 'ampl portion good price',
 'poor servic waiter made feel like stupid everi time came tabl',
 'first visit hiro delight',
 'servic suck',
 'shrimp tender moist',
 'deal good enough would drag establish',
 'hard judg whether side good gross melt styrofoam want eat fear get
sick',
 'posit note server attent provid great servic',
 'frozen puck disgust worst peopl behind regist',
 'thing like prime rib dessert section',
 'bad food damn gener',
 'burger good beef cook right',
 'want sandwich go firehous',
 'side greek salad greek dress tasti pita hummu refresh',
 'order duck rare pink tender insid nice char outsid',
 'came run us realiz husband left sunglass tabl',
 'chow mein good',
 'horribl attitud toward custom talk one custom enjoy food',
 'portion huge',
 'love friendli server great food wonder imagin menu',
 'heart attack grill downtown vega absolut flat line excus restaur',
 'much seafood like string pasta bottom',
 'salad right amount sauc power scallop perfectli cook',
 'rip banana rip petrifi tasteless',
 'least think refil water struggl wave minut',
 'place receiv star appet',
 'cocktail handmad delici',
 'definit go back',
 'glad found place',
 'great food servic huge portion give militari discount',
 'alway great time do gringo',
 'updat went back second time still amaz',
 'got food appar never heard salt batter fish chewi',
 'great way finish great',
 'deal includ tast drink jeff went beyond expect',
 'realli realli good rice time',
```

```
 'servic meh',
 'took min get milkshak noth chocol milk',
 'guess known place would suck insid excalibur use common sens',
 'scallop dish quit appal valu well',
 'time bad custom servic',
 'sweet potato fri good season well',
 'today second time lunch buffet pretti good',
 'much good food vega feel cheat wast eat opportun go rice compani',
 'come like experienc underwhelm relationship parti wait person ask
break',
 'walk place smell like old greas trap other eat',
 'turkey roast beef bland',
 'place',
 'pan cake everyon rave tast like sugari disast tailor palat six year
old',
 'love pho spring roll oh yummi tri',
 'poor batter meat ratio made chicken tender unsatisfi',
 'say food amaz',
 'omelet die',
 'everyth fresh delici',
 'summari larg disappoint dine experi',
 'like realli sexi parti mouth outrag flirt hottest person parti',
 'never hard rock casino never ever step forward',
 'best breakfast buffet',
 'say bye bye tip ladi',
 'never go',
 'back',
 'food arriv quickli',
 'good',
 'side cafe serv realli good food',
 'server fantast found wife love roast garlic bone marrow ad extra
meal anoth marrow go',
 'good thing waiter help kept bloddi mari come',
 'best buffet town price cannot beat',
 'love mussel cook wine reduct duck tender potato dish delici',
 'one better buffet',
 'went tigerlilli fantast afternoon',
 'food delici bartend attent person got great deal',
 'ambienc wonder music play',
 'go back next trip',
 'sooooo good',
 'real sushi lover let honest yama good',
 'least min pass us order food arriv busi',
 'realli fantast thai restaur definit worth visit',
 'nice spici tender',
 'good price',
 'check',
 'pretti gross',
 'better atmospher',
```

```
 'kind hard mess steak',
 'although much like look sound place actual experi bit disappoint',
 'know place manag serv blandest food ever eaten prepar indian
cuisin',
 'worst servic boot least worri',
 'servic fine waitress friendli',
 'guy steak steak love son steak best worst place said best steak ever
eaten',
 'thought ventur away get good sushi place realli hit spot night',
 'host staff lack better word bitch',
 'bland like place number reason want wast time bad review leav',
 'phenomen food servic ambianc',
 'return',
 'definit worth ventur strip pork belli return next time vega',
 'place way overpr mediocr food',
 'penn vodka excel',
 'good select food includ massiv meatloaf sandwich crispi chicken wrap
delish tuna melt tasti burger',
 'manag rude',
 'delici nyc bagel good select cream chees real lox caper even',
 'great subway fact good come everi subway meet expect',
 'serious solid breakfast',
 'one best bar food vega',
 'extrem rude realli mani restaur would love dine weekend vega',
 'drink never empti made realli great menu suggest',
 '',
 'waiter help friendli rare check us',
 'husband ate lunch disappoint food servic',
 'red curri much bamboo shoot tasti',
 'nice blanket moz top feel like done cover subpar food',
 'bathroom clean place well decor',
 'menu alway chang food qualiti go servic extrem slow',
 'servic littl slow consid serv peopl server food come slow pace',
 'give thumb',
 'watch waiter pay lot attent tabl ignor us',
 'fianc came middl day greet seat right away',
 'great restaur mandalay bay',
 'wait forti five minut vain',
 'crostini came salad stale',
 'highlight great qualiti nigiri',
 'staff friendli joint alway clean',
 'differ cut piec day still wonder tender well well flavor',
 'order voodoo pasta first time realli excel pasta sinc go gluten free
sever year ago',
 'place good',
 'unfortun must hit bakeri leftov day everyth order stale',
 'came back today sinc reloc still impress',
 'seat immedi',
 'menu divers reason price',
```

```
 'avoid cost',
 'restaur alway full never wait',
 'delici',
 'place hand one best place eat phoenix metro area',
 'go look good food',
 'never treat bad',
 'bacon hella salti',
 'also order spinach avocado salad ingredi sad dress liter zero tast',
 'realli vega fine dine use right menu hand ladi price list',
 'waitress friendli',
 'lordi khao soi dish miss curri lover',
 'everyth menu terrif also thrill made amaz accommod vegetarian
daughter',
 'perhap caught night judg review inspir go back',
 'servic leav lot desir',
 'atmospher modern hip maintain touch cozi',
 'weekli haunt definit place come back everi',
 'liter sat minut one ask take order',
 'burger absolut flavor meat total bland burger overcook charcoal
flavor',
 'also decid send back waitress look like verg heart attack',
 'dress treat rude',
 'probabl dirt',
 'love place hit spot want someth healthi lack quantiti flavor',
 'order lemon raspberri ice cocktail also incred',
 'food suck expect suck could imagin',
 'interest decor',
 'realli like crepe station',
 'also serv hot bread butter home made potato chip bacon bit top
origin good',
 'watch prepar delici food',
 'egg roll fantast',
 'order arriv one gyro miss',
 'salad wing ice cream dessert left feel quit satisfi',
 'realli sure joey vote best hot dog valley reader phoenix magazin',
 'best place go tasti bowl pho',
 'live music friday total blow',
 'never insult felt disrespect',
 'friendli staff',
 'worth drive',
 'heard good thing place exceed everi hope could dream',
 'food great serivc',
 'warm beer help',
 'great brunch spot',
 'servic friendli invit',
 'good lunch spot',
 'live sinc first last time step foot place',
 'worst experi ever',
 'must night place',
```

```
 'side delish mix mushroom yukon gold pure white corn beateou',
 'bug never show would given sure side wall bug climb kitchen',
 'minut wait salad realiz come time soon',
 'friend love salmon tartar',
 'go back',
 'extrem tasti',
 'waitress good though',
 'soggi good',
 'jamaican mojito delici',
 'small worth price',
 'food rich order accordingli',
 'shower area outsid rins take full shower unless mind nude everyon
see',
 'servic bit lack',
 'lobster bisqu bussel sprout risotto filet need salt pepper cours
none tabl',
 'hope bode go busi someon cook come',
 'either cold enough flavor bad',
 'love bacon wrap date',
 'unbeliev bargain',
 'folk otto alway make us feel welcom special',
 'main also uninspir',
 'place first pho amaz',
 'wonder experi made place must stop whenev town',
 'food bad enough enjoy deal world worst annoy drunk peopl',
 'fun chef',
 'order doubl cheeseburg got singl patti fall apart pictur upload yeah
still suck',
 'great place coupl drink watch sport event wall cover tv',
 'possibl give zero star',
 'descript said yum yum sauc anoth said eel sauc yet anoth said spici
mayo well none roll sauc',
 'say would hardest decis honestli dish tast suppos tast amaz',
 'roll eye may stay sure go back tri',
 'everyon attent provid excel custom servic',
 'horribl wast time money',
 'dish quit flavour',
 'time side restaur almost empti excus',
 'busi either also build freez cold',
 'like review said pay eat place',
 'drink took close minut come one point',
 'serious flavor delight folk',
 'much better ayc sushi place went vega',
 'light dark enough set mood',
 'base sub par servic receiv effort show gratitud busi go back',
 'owner realli great peopl',
 'noth privileg work eat',
 'greek dress creami flavor',
 'overal think would take parent place made similar complaint silent
```

```
felt',
 'pizza good peanut sauc tasti',
 'tabl servic pretti fast',
 'fantast servic',
 'well would given godfath zero star possibl',
 'know make',
 'tough short flavor',
 'hope place stick around',
 'bar vega ever recal charg tap water',
 'restaur atmospher exquisit',
 'good servic clean inexpens boot',
 'seafood fresh gener portion',
 'plu buck',
 'servic par either',
 'thu far visit twice food absolut delici time',
 'good year ago',
 'self proclaim coffe cafe wildli disappoint',
 'veggitarian platter world',
 'cant go wrong food',
 'beat',
 'stop place madison ironman friendli kind staff',
 'chef friendli good job',
 'better dedic boba tea spot even jenni pho',
 'like patio servic outstand',
 'goat taco skimp meat wow flavor',
 'think',
 'mac salad pretti bland get',
 'went bachi burger friend recommend disappoint',
 'servic stink',
 'wait wait',
 'place qualiti sushi qualiti restaur',
 'would definit recommend wing well pizza',
 'great pizza salad',
 'thing went wrong burn saganaki',
 'wait hour breakfast could done time better home',
 'place amaz',
 'hate disagre fellow yelper husband disappoint place',
 'wait hour never got either pizza mani around us came later',
 'know slow',
 'staff great food delish incred beer select',
 'live neighborhood disappoint back conveni locat',
 'know pull pork could soooo delici',
 'get incred fresh fish prepar care',
 'go gave star rate pleas know third time eat bachi burger write
review',
 'love fact everyth menu worth',
 'never dine place',
 'food excel servic good',
 'good beer drink select good food select',
```

```
 'pleas stay away shrimp stir fri noodl',
 'potato chip order sad could probabl count mani chip box probabl
around',
 'food realli bore',
 'good servic check',
 'greedi corpor never see anoth dime',
 'never ever go back',
 'much like go back get pass atroci servic never return',
 'summer dine charm outdoor patio delight',
 'expect good',
 'fantast food',
 'order toast english muffin came untoast',
 'food good',
 'never go back',
 'great food price high qualiti hous made',
 'bu boy hand rude',
 'point friend basic figur place joke mind make publicli loudli
known',
 'back good bbq lighter fare reason price tell public back old way',
 'consid two us left full happi go wrong',
 'bread made hous',
 'downsid servic',
 'also fri without doubt worst fri ever',
 'servic except food good review',
 'coupl month later return amaz meal',
 'favorit place town shawarrrrrrma',
 'black eye pea sweet potato unreal',
 'disappoint',
 'could serv vinaigrett may make better overal dish still good',
 'go far mani place never seen restaur serv egg breakfast especi',
 'mom got home immedi got sick bite salad',
 'server pleasant deal alway honor pizza hut coupon',
 'truli unbeliev good glad went back',
 'fantast servic pleas atmospher',
 'everyth gross',
 'love place',
 'great servic food',
 'first bathroom locat dirti seat cover replenish plain yucki',
 'burger got gold standard burger kind disappoint',
 'omg food delicioso',
 'noth authent place',
 'spaghetti noth special whatsoev',
 'dish salmon best great',
 'veget fresh sauc feel like authent thai',
 'worth drive tucson',
 'select probabl worst seen vega none',
 'pretti good beer select',
 'place like chipotl better',
 'classi warm atmospher fun fresh appet succul steak basebal steak',
```

```
 'star brick oven bread app',
 'eaten multipl time time food delici',
 'sat anoth ten minut final gave left',
 'terribl',
 'everyon treat equal special',
 'take min pancak egg',
 'delici',
 'good side staff genuin pleasant enthusiast real treat',
 'sadli gordon ramsey steak place shall sharpli avoid next trip vega',
 'alway even wonder food delici',
 'best fish ever life',
 'bathroom next door nice',
 'buffet small food offer bland',
 'outstand littl restaur best food ever tast',
 'pretti cool would say',
 'definit turn doubt back unless someon els buy',
 'server great job handl larg rowdi tabl',
 'find wast food despic food',
 'wife lobster bisqu soup lukewarm',
 'would come back sushi crave vega',
 'staff great ambianc great',
 'deserv star',
 'left stomach ach felt sick rest day',
 'drop ball',
 'dine space tini elegantli decor comfort',
 'custom order way like usual eggplant green bean stir fri love',
 'bean rice mediocr best',
 'best taco town far',
 'took back money got outta',
 'interest part town place amaz',
 'rude inconsider manag',
 'staff friendli wait time serv horribl one even say hi first minut',
 'back',
 'great dinner',
 'servic outshin definit recommend halibut',
 'food terribl',
 'never ever go back told mani peopl happen',
 'recommend unless car break front starv',
 'come back everi time vega',
 'place deserv one star food',
 'disgrac',
 'def come back bowl next time',
 'want healthi authent ethic food tri place',
 'continu come ladi night andddd date night highli recommend place
anyon area',
 'sever time past experi alway great',
 'walk away stuf happi first vega buffet experi',
 'servic excel price pretti reason consid vega locat insid crystal
shop mall aria',
```

```
 'summar food incred nay transcend noth bring joy quit like memori
pneumat condiment dispens',
 'probabl one peopl ever go ian like',
 'kid pizza alway hit lot great side dish option kiddo',
 'servic perfect famili atmospher nice see',
 'cook perfect servic impecc',
 'one simpli disappoint',
 'overal disappoint qualiti food bouchon',
 'account know get screw',
 'great place eat remind littl mom pop shop san francisco bay area',
 'today first tast buldogi gourmet hot dog tell ever thought possibl',
 'left frustrat',
 'definit soon',
 'food realli good got full petti fast',
 'servic fantast',
 'total wast time',
 'know kind best ice tea',
 'come hungri leav happi stuf',
 'servic give star',
 'assur disappoint',
 'take littl bad servic food suck',
 'gave tri eat crust teeth still sore',
 'complet gross',
 'realli enjoy eat',
 'first time go think quickli becom regular',
 'server nice even though look littl overwhelm need stay profession
friendli end',
 'dinner companion told everyth fresh nice textur tast',
 'ground right next tabl larg smear step track everywher pile green
bird poop',
 'furthermor even find hour oper websit',
 'tri like place time think done',
 'mistak',
 'complaint',
 'serious good pizza expert connisseur topic',
 'waiter jerk',
 'strike want rush',
 'nicest restaur owner ever come across',
 'never come',
 'love biscuit',
 'servic quick friendli',
 'order appet took minut pizza anoth minut',
 'absolutley fantast',
 'huge awkward lb piec cow th gristl fat',
 'definit come back',
 'like steiner dark feel like bar',
 'wow spici delici',
 'familiar check',
 'take busi dinner dollar elsewher',
```

```
 'love go back',
 'anyway fs restaur wonder breakfast lunch',
 'noth special',
 'day week differ deal delici',
 'mention combin pear almond bacon big winner',
 'back',
 'sauc tasteless',
 'food delici spici enough sure ask spicier prefer way',
 'ribey steak cook perfectli great mesquit flavor',
 'think go back anytim soon',
 'food gooodd',
 'far sushi connoisseur definit tell differ good food bad food
certainli bad food',
 'insult',
 'last time lunch bad',
 'chicken wing contain driest chicken meat ever eaten',
 'food good enjoy everi mouth enjoy relax venu coupl small famili
group etc',
 'nargil think great',
 'best tater tot southwest',
 'love place',
 'definit worth paid',
 'vanilla ice cream creami smooth profiterol choux pastri fresh
enough',
 'im az time new spot',
 'manag worst',
 'insid realli quit nice clean',
 'food outstand price reason',
 'think run back carli anytim soon food',
 'due fact took minut acknowledg anoth minut get food kept forget
thing',
 'love margarita',
 'first vega buffet disappoint',
 'good though',
 'one note ventil could use upgrad',
 'great pork sandwich',
 'wast time',
 'total letdown would much rather go camelback flower shop cartel
coffe',
 'third chees friend burger cold',
 'enjoy pizza brunch',
 'steak well trim also perfectli cook',
 'group claim would handl us beauti',
 'love',
 'ask bill leav without eat bring either',
 'place jewel la vega exactli hope find nearli ten year live',
 'seafood limit boil shrimp crab leg crab leg definit tast fresh',
 'select food best',
 'delici absolut back',
 'small famili restaur fine dine establish',
```

```
 'toro tartar cavier extraordinari like thinli slice wagyu white
truffl',
 'dont think back long time',
 'attach ga station rare good sign',
 'awesom',
 'back mani time soon',
 'menu much good stuff could decid',
 'wors humili worker right front bunch horribl name call',
 'conclus fill meal',
 'daili special alway hit group',
 'tragedi struck',
 'pancak also realli good pretti larg',
 'first crawfish experi delici',
 'monster chicken fri steak egg time favorit',
 'waitress sweet funni',
 'also tast mom multi grain pumpkin pancak pecan butter amaz fluffi
delici',
 'rather eat airlin food serious',
 'cant say enough good thing place',
 'ambianc incred',
 'waitress manag friendli',
 'would recommend place',
 'overal impress noca',
 'gyro basic lettuc',
 'terribl servic',
 'thoroughli disappoint',
 'much pasta love homemad hand made pasta thin pizza',
 'give tri happi',
 'far best cheesecurd ever',
 'reason price also',
 'everyth perfect night',
 'food good typic bar food',
 'drive get',
 'first glanc love bakeri cafe nice ambianc clean friendli staff',
 'anyway think go back',
 'point finger item menu order disappoint',
 'oh thing beauti restaur',
 'gone go',
 'greasi unhealthi meal',
 'first time might last',
 'burger amaz',
 'similarli deliveri man say word apolog food minut late',
 'way expens',
 'sure order dessert even need pack go tiramisu cannoli die',
 'first time wait next',
 'bartend also nice',
 'everyth good tasti',
 'place two thumb way',
 'best place vega breakfast check sat sun',
```

```
 'love authent mexican food want whole bunch interest yet delici meat
choos need tri place',
 'terribl manag',
 'excel new restaur experienc frenchman',
 'zero star would give zero star',
 'great steak great side great wine amaz dessert',
 'worst martini ever',
 'steak shrimp opinion best entre gc',
 'opportun today sampl amaz pizza',
 'wait thirti minut seat although vacant tabl folk wait',
 'yellowtail carpaccio melt mouth fresh',
 'tri go back even empti',
 'go eat potato found stranger hair',
 'spici enough perfect actual',
 'last night second time dine happi decid go back',
 'even hello right',
 'dessert bit strang',
 'boyfriend came first time recent trip vega could pleas qualiti food
servic',
 'realli recommend place go wrong donut place',
 'nice ambianc',
 'would recommend save room',
 'guess mayb went night disgrac',
 'howev recent experi particular locat good',
 'know like restaur someth',
 'avoid establish',
 'think restaur suffer tri hard enough',
 'tapa dish delici',
 'heart place',
 'salad bland vinegrett babi green heart palm',
 'two felt disgust',
 'good time',
 'believ place great stop huge belli hanker sushi',
 'gener portion great tast',
 'never go back place never ever recommend place anyon',
 'server went back forth sever time even much help',
 'food delici',
 'hour serious',
 'consid theft',
 'eew locat need complet overhaul',
 'recent wit poor qualiti manag toward guest well',
 'wait wait wait',
 'also came back check us regularli excel servic',
 'server super nice check us mani time',
 'pizza tast old super chewi good way',
 'swung give tri deepli disappoint',
 'servic good compani better',
 'staff also friendli effici',
 'servic fan quick serv nice folk',
```

```
 'boy sucker dri',
 'rate',
 'look authent thai food go els',
 'steak recommend',
 'pull car wait anoth minut acknowledg',
 'great food great servic clean friendli set',
 'assur back',
 'hate thing much cheap qualiti black oliv',
 'breakfast perpar great beauti present giant slice toast lightli dust
powder sugar',
 'kid play area nasti',
 'great place fo take eat',
 'waitress friendli happi accomod vegan veggi option',
 'omg felt like never eaten thai food dish',
 'extrem crumbi pretti tasteless',
 'pale color instead nice char flavor',
 'crouton also tast homemad extra plu',
 'got home see driest damn wing ever',
 'regular stop trip phoenix',
 'realli enjoy crema caf expand even told friend best breakfast',
 'good money',
 'miss wish one philadelphia',
 'got sit fairli fast end wait minut place order anoth minut food
arriv',
 'also best chees crisp town',
 'good valu great food great servic',
 'ask satisfi meal',
 'food good',
 'awesom',
 'want leav',
 'made drive way north scottsdal one bit disappoint',
 'eat',
 'owner realli realli need quit soooooo cheap let wrap freak sandwich
two paper one',
 'check place coupl year ago impress',
 'chicken got definit reheat ok wedg cold soggi',
 'sorri get food anytim soon',
 'absolut must visit',
 'cow tongu cheek taco amaz',
 'friend like bloodi mari',
 'despit hard rate busi actual rare give star',
 'realli want make experi good one',
 'return',
 'chicken pho tast bland',
 'disappoint',
 'grill chicken tender yellow saffron season',
 'drive thru mean want wait around half hour food somehow end go make
us wait wait',
 'pretti awesom place',
```

```
 'ambienc perfect',
 'best luck rude non custom servic focus new manag',
 'grandmoth make roast chicken better one',
 'ask multipl time wine list time ignor went hostess got one',
 'staff alway super friendli help especi cool bring two small boy
babi',
 'four star food guy blue shirt great vibe still let us eat',
 'roast beef sandwich tast realli good',
 'even drastic sick',
 'high qualiti chicken chicken caesar salad',
 'order burger rare came done',
 'promptli greet seat',
 'tri go lunch madhous',
 'proven dead wrong sushi bar qualiti great servic fast food impecc',
 'wait hour seat greatest mood',
 'good joint',
 'macaron insan good',
 'eat',
 'waiter attent friendli inform',
 'mayb cold would somewhat edibl',
 'place lot promis fail deliv',
 'bad experi',
 'mistak',
 'food averag best',
 'great food',
 'go back anytim soon',
 'disappoint order big bay plater',
 'great place relax awesom burger beer',
 'perfect sit famili meal get togeth friend',
 'much flavor poorli construct',
 'patio seat comfort',
 'fri rice dri well',
 'hand favorit italian restaur',
 'scream legit book somethat also pretti rare vega',
 'fun experi',
 'atmospher great love duo violinist play song request',
 'person love hummu pita baklava falafel baba ganoush amaz eggplant',
 'conveni sinc stay mgm',
 'owner super friendli staff courteou',
 'great',
 'eclect select',
 'sweet potato tot good onion ring perfect close',
 'staff attent',
 'chef gener time even came around twice take pictur',
 'owner use work nobu place realli similar half price',
 'googl mediocr imagin smashburg pop',
 'dont go',
 'promis disappoint',
 'sushi lover avoid place mean',
```

```
 'great doubl cheeseburg',
 'awesom servic food',
 'fantast neighborhood gem',
 'wait go back',
 'plantain worst ever tast',
 'great place highli recommend',
 'servic slow attent',
 'gave star give star',
 'staff spend time talk',
 'dessert panna cotta amaz',
 'good food great atmospher',
 'damn good steak',
 'total brunch fail',
 'price reason flavor spot sauc home made slaw drench mayo',
 'decor nice piano music soundtrack pleasant',
 'steak amaz rge fillet relleno best seafood plate ever',
 'good food good servic',
 'absolut amaz',
 'probabl back honest',
 'definit back',
 'sergeant pepper beef sandwich auju sauc excel sandwich well',
 'hawaiian breez mango magic pineappl delight smoothi tri far good',
 'went lunch servic slow',
 'much say place walk expect amaz quickli disappoint',
 'mortifi',
 'needless say never back',
 'anyway food definit fill price pay expect',
 'chip came drip greas mostli edibl',
 'realli impress strip steak',
 'go sinc everi meal awesom',
 'server nice attent serv staff',
 'cashier friendli even brought food',
 'work hospit industri paradis valley refrain recommend cibo longer',
 'atmospher fun',
 'would recommend other',
 'servic quick even go order like like',
 'mean realli get famou fish chip terribl',
 'said mouth belli still quit pleas',
 'thing',
 'thumb',
 'read pleas go',
 'love grill pizza remind legit italian pizza',
 'pro larg seat area nice bar area great simpl drink menu best brick
oven pizza homemad dough',
 'realli nice atmospher',
 'tonight elk filet special suck',
 'one bite hook',
 'order old classic new dish go time sore disappoint everyth',
 'cute quaint simpl honest',
```

```
'chicken delici season perfect fri outsid moist chicken insid',
'food great alway compliment chef',
'special thank dylan recommend order yummi tummi',
'awesom select beer',
'great food awesom servic',
'one nice thing ad gratuiti bill sinc parti larger expect tip',
'fli appl juic fli',
'han nan chicken also tasti',
'servic thought good',
'food bare lukewarm must sit wait server bring us',
'ryan bar definit one edinburgh establish revisit',
'nicest chines restaur',
'overal like food servic',
'also serv indian naan bread hummu spici pine nut sauc world',
'probabl never come back recommend',
'friend pasta also bad bare touch',
'tri airport experi tasti food speedi friendli servic',
'love decor chines calligraphi wall paper',
'never anyth complain',
'restaur clean famili restaur feel',
'way fri',
'sure long stood long enough begin feel awkwardli place',
'open sandwich impress good way',
'back',
'warm feel servic felt like guest special treat',
'extens menu provid lot option breakfast',
'alway order vegetarian menu dinner wide array option choos',
'watch price inflat portion get smaller manag attitud grow rapidli',
'wonder lil tapa ambienc made feel warm fuzzi insid',
'got enjoy seafood salad fabul vinegrett',
'wonton thin thick chewi almost melt mouth',
'level spici perfect spice whelm soup',
'sat right time server get go fantast',
'main thing enjoy crowd older crowd around mid',
'side town definit spot hit',
'wait minut get drink longer get arepa',
'great place eat',
'jalapeno bacon soooo good',
'servic poor that nice',
'food good servic good price good',
'place clean food oh stale',
'chicken dish ok beef like shoe leather',
'servic beyond bad',
'happi',
'tast like dirt',
'one place phoenix would defin go back',
'block amaz',
'close hous low key non fanci afford price good food',
'hot sour egg flower soup absolut star',
```

```
 'sashimi poor qualiti soggi tasteless',
 'great time famili dinner sunday night',
 'food tasti say real tradit hunan style',
 'bother slow servic',
 'flair bartend absolut amaz',
 'frozen margarita way sugari tast',
 'good order twice',
 'nutshel restaraunt smell like combin dirti fish market sewer',
 'girlfriend veal bad',
 'unfortun good',
 'pretti satifi experi',
 'join club get awesom offer via email',
 'perfect someon like beer ice cold case even colder',
 'bland flavorless good way describ bare tepid meat',
 'chain fan beat place easili',
 'nacho must',
 'come back',
 'mani word say place everyth pretti well',
 'staff super nice quick even crazi crowd downtown juri lawyer court
staff',
 'great atmospher friendli fast servic',
 'receiv pita huge lot meat thumb',
 'food arriv meh',
 'pay hot dog fri look like came kid meal wienerschnitzel idea good
meal',
 'classic main lobster roll fantast',
 'brother law work mall ate day guess sick night',
 'good go review place twice herea tribut place tribut event held last
night',
 'chip salsa realli good salsa fresh',
 'place great',
 'mediocr food',
 'get insid impress place',
 'super pissd',
 'servic super friendli',
 'sad littl veget overcook',
 'place nice surpris',
 'golden crispi delici',
 'high hope place sinc burger cook charcoal grill unfortun tast fell
flat way flat',
 'could eat bruschetta day devin',
 'singl employe came see ok even need water refil final serv us food',
 'lastli mozzarella stick best thing order',
 'first time ever came amaz experi still tell peopl awesom duck',
 'server neglig need made us feel unwelcom would suggest place',
 'servic terribl though',
 'place overpr consist boba realli overpr',
 'pack',
 'love place',
```

```
 'say dessert yummi',
 'food terribl',
 'season fruit fresh white peach pure',
 'kept get wors wors offici done',
 'place honestli blown',
 'definit would eat',
 'wast money',
 'love put food nice plastic contain oppos cram littl paper takeout
box',
 'cr pe delic thin moist',
 'aw servic',
 'ever go',
 'food qualiti horribl',
 'price think place would much rather gone',
 'servic fair best',
 'love sushi found kabuki price hip servic',
 'favor stay away dish',
 'poor servic',
 'one tabl thought food averag worth wait',
 'best servic food ever maria server good friendli made day',
 'excel',
 'paid bill tip felt server terribl job',
 'lunch great experi',
 'never bland food surpris consid articl read focus much spice
flavor',
 'food way overpr portion fuck small',
 'recent tri caballero back everi week sinc',
 'buck head realli expect better food',
 'food came good pace',
 'ate twice last visit especi enjoy salmon salad',
 'back',
 'could believ dirti oyster',
 'place deserv star',
 'would recommend place',
 'fact go round star awesom',
 'disbelief dish qualifi worst version food ever tast',
 'bad day low toler rude custom servic peopl job nice polit wash dish
otherwis',
 'potato great biscuit',
 'probabl would go',
 'flavor perfect amount heat',
 'price reason servic great',
 'wife hate meal coconut shrimp friend realli enjoy meal either',
 'fella got huevo ranchero look appeal',
 'went happi hour great list wine',
 'may say buffet pricey think get pay place get quit lot',
 'probabl come back',
 'worst food servic',
 'place pretti good nice littl vibe restaur',
```

```
 'talk great custom servic cours back',
 'hot dish hot cold dish close room temp watch staff prepar food bare
hand glove everyth deep fri oil',
 'love fri bean',
 'alway pleasur deal',
 'plethora salad sandwich everyth tri get seal approv',
 'place awesom want someth light healthi summer',
 'sushi strip place go',
 'servic great even manag came help tabl',
 'feel dine room colleg cook cours high class dine servic slow best',
 'start review two star edit give one',
 'worst sushi ever eat besid costco',
 'excel restaur highlight great servic uniqu menu beauti set',
 'boyfriend sat bar complet delight experi',
 'weird vibe owner',
 'hardli meat',
 'better bagel groceri store',
 'go place gyro',
 'love owner chef one authent japanes cool dude',
 'burger good pizza use amaz doughi flavorless',
 'found six inch long piec wire salsa',
 'servic terribl food mediocr',
 'defin enjoy',
 'order albondiga soup warm tast like tomato soup frozen meatbal',
 'three differ occas ask well done medium well three time got
bloodiest piec meat plate',
 'two bite refus eat anymor',
 'servic extrem slow',
 'minut wait got tabl',
 'serious killer hot chai latt',
 'allergi warn menu waitress absolut clue meal contain peanut',
 'boyfriend tri mediterranean chicken salad fell love',
 'rotat beer tap also highlight place',
 'price bit concern mellow mushroom',
 'worst thai ever',
 'stay vega must get breakfast least',
 'want first say server great perfect servic',
 'pizza select good',
 'strawberri tea good',
 'highli unprofession rude loyal patron',
 'overal great experi',
 'spend money elsewher',
 'regular toast bread equal satisfi occasion pat butter mmmm',
 'buffet bellagio far anticip',
 'drink weak peopl',
 'order correct',
 'also feel like chip bought made hous',
 'disappoint dinner went elsewher dessert',
 'chip sal amaz',
```

```
 'return',
 'new fav vega buffet spot',
 'serious cannot believ owner mani unexperienc employe run around like chicken head cut',
 'sad',
 'felt insult disrespect could talk judg anoth human like',
 'call steakhous properli cook steak understand',
 'impress concept food',
 'thing crazi guacamol like pur ed',
 'realli noth postino hope experi better',
 'got food poison buffet',
 'brought fresh batch fri think yay someth warm',
 'hilari yummi christma eve dinner rememb biggest fail entir trip us',
 'needless say go back anytim soon',
 'place disgust',
 'everi time eat see care teamwork profession degre',
 'ri style calamari joke',
 'howev much garlic fondu bare edibl',
 'could bare stomach meal complain busi lunch',
 'bad lost heart finish',
 'also took forev bring us check ask',
 'one make scene restaur get definit lost love one',
 'disappoint experi',
 'food par denni say good',
 'want wait mediocr food downright terribl servic place',
 'waaaaaayyyyyyyyyi rate say',
 'go back',
 'place fairli clean food simpli worth',
 'place lack style',
 'sangria half glass wine full ridicul',
 'bother come',
 'meat pretti dri slice brisket pull pork',
 'build seem pretti neat bathroom pretti trippi eat',
 'equal aw',
 'probabl hurri go back',
 'slow seat even reserv',
 'good stretch imagin',
 'cashew cream sauc bland veget undercook',
 'chipolt ranch dip saus tasteless seem thin water heat',
 'bit sweet realli spici enough lack flavor',
 'disappoint',
 'place horribl way overpr',
 'mayb vegetarian fare twice thought averag best',
 'busi know',
 'tabl outsid also dirti lot time worker alway friendli help menu',
 'ambianc feel like buffet set douchey indoor garden tea biscuit',
 'con spotti servic',
 'fri hot neither burger',
 'came back cold',
```

```
 'food came disappoint ensu',
 'real disappoint waiter',
 'husband said rude even apolog bad food anyth',
 'reason eat would fill night bing drink get carb stomach',
 'insult profound deuchebaggeri go outsid smoke break serv solidifi',
 'someon order two taco think may part custom servic ask combo ala
cart',
 'quit disappoint although blame need place door',
 'rave review wait eat disappoint',
 'del taco pretti nasti avoid possibl',
 'hard make decent hamburg',
 'like',
 'hell go back',
 'gotten much better servic pizza place next door servic receiv
restaur',
 'know big deal place back ya',
 'immedi said want talk manag want talk guy shot firebal behind bar',
 'ambianc much better',
 'unfortun set us disapppoint entre',
 'food good',
 'server suck wait correct server heimer suck',
 'happen next pretti put',
 'bad caus know famili own realli want like place',
 'overpr get',
 'vomit bathroom mid lunch',
 'kept look time soon becom minut yet still food',
 'place eat circumst would ever return top list',
 'start tuna sashimi brownish color obvious fresh',
 'food averag',
 'sure beat nacho movi would expect littl bit come restaur',
 'ha long bay bit flop',
 'problem charg sandwich bigger subway sub offer better amount veget',
 'shrimp unwrap live mile brushfir liter ice cold',
 'lack flavor seem undercook dri',
 'realli impress place close',
 'would avoid place stay mirag',
 'refri bean came meal dri crusti food bland',
 'spend money time place els',
 'ladi tabl next us found live green caterpillar salad',
 'present food aw',
 'tell disappoint',
 'think food flavor textur lack',
 'appetit instantli gone',
 'overal impress would go back',
 'whole experi underwhelm think go ninja sushi next time',
 'wast enough life pour salt wound draw time took bring check']
```

Word CLoud

```
# import library
# positive review

from wordcloud import WordCloud
import matplotlib.pyplot as plt
word_cloud = df.loc[df['Liked'] == 1,:]
text = ' '.join([text for text in word_cloud['Review']])
# Generate a WordCloud object
wordcloud = WordCloud(width=800,
height=400,background_color='white').generate(text)
# Display the word cloud using matplotlib
plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```


```
# Creating a Bag of Words Model

# In this code, we create a Bag of Words (BoW) model from the 'corpus'
of cleaned text reviews.

# Import the CountVectorizer class from scikit-learn, which is used to
convert text data into numerical features.
from sklearn.feature_extraction.text import CountVectorizer

# Initialize a CountVectorizer with a specified maximum number of
features (max_features).
# The 'max_features' parameter controls the number of most frequent
```

```
words to keep in the vocabulary.
# Adjust this value based on your specific requirements.
cv = CountVectorizer(max_features=1500)

# Apply the CountVectorizer to the 'corpus' to transform the text data
into a numerical representation.
# The 'fit_transform' method converts the text into a sparse matrix
where rows represent reviews and columns represent words.
X = cv.fit_transform(corpus).toarray()

# 'X' now contains the BoW representation of the text data.

# Extract the target variable 'y' from the DataFrame 'df'.
# Assuming the target variable is located in the second column (index
1) of the DataFrame.
y = df.iloc[:, 1].values

# 'y' now contains the labels or target values corresponding to each
review.

# The resulting 'X' and 'y' can be used to train machine learning
models for tasks such as sentiment analysis or text classification.

# Splitting the Dataset into Training and Testing Sets

# In this code, we split the dataset into training and testing sets to
evaluate the performance of a machine learning model.

# Import the 'train_test_split' function from scikit-learn, which is
used for splitting datasets.

######################################################################
##################

from sklearn.model_selection import train_test_split

# Split the feature matrix 'X' and the target variable 'y' into
training and testing sets.
# The 'test_size' parameter specifies the proportion of the dataset to
include in the test split.
# Here, 20% of the data is reserved for testing (test_size=0.20).
# The 'random_state' parameter ensures reproducibility by fixing the
random seed for the split.
# This means that the same split will be obtained every time you run
the code with the same random_state value.

######################################################################
######################

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=0)
```

```
##########################################################################

# 'X_train' and 'y_train' contain the features and labels for the training set, respectively.
# 'X_test' and 'y_test' contain the features and labels for the testing set, respectively.

# The dataset is typically divided into a training set (used to train the model) and a testing set (used to evaluate the model's performance).
# The proportions used in this split can be adjusted based on the specific requirements of your analysis.

# In this example, the dataset contains 1000 samples, and 80% (800 samples) are used for training,
# while 20% (200 samples) are used for testing. These proportions can be modified as needed.
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((800, 1500), (200, 1500), (800,), (200,))
```

Model Training

```
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier

# Define a dictionary of models with their names as keys and model instances as values
models = {
    'MultinomialNB': MultinomialNB(),
    'RandomForestClassifier': RandomForestClassifier(),
    'GradientBoostingClassifier': GradientBoostingClassifier(),
    'SVC': SVC(),
    'LogisticRegression': LogisticRegression(),
    'KNeighborsClassifier': KNeighborsClassifier(n_neighbors=5),
```

```
    'DecisionTreeClassifier': DecisionTreeClassifier(),
    'MLPClassifier': MLPClassifier(max_iter=1000),
    'AdaBoostClassifier': AdaBoostClassifier(),
    'BaggingClassifier': BaggingClassifier(),
    'ExtraTreesClassifier': ExtraTreesClassifier(),
    # Add more models as needed
}

# Create an empty dictionary to store model accuracies
model_accuracies = {}

# Loop through each model and train/evaluate it
for model_name, model in models.items():
    # Train the model
    model.fit(X_train, y_train)

    # Make predictions on the testing data
    y_pred = model.predict(X_test)

    # Calculate accuracy for the model
    accuracy = accuracy_score(y_test, y_pred)

    # Store the accuracy in the model_accuracies dictionary
    model_accuracies[model_name] = accuracy

# Rank the models based on accuracy (highest to lowest)
ranked_models = sorted(model_accuracies.items(), key=lambda x: x[1],
reverse=True)

# Print the ranked models
print("Model Rankings (Accuracy):")
for rank, (model_name, accuracy) in enumerate(ranked_models, start=1):
    print(f"{rank}. {model_name}: {accuracy * 100:.2f}%")
```

```
Model Rankings (Accuracy):
1. MultinomialNB: 76.50%
2. MLPClassifier: 74.50%
3. GradientBoostingClassifier: 73.50%
4. SVC: 73.50%
5. LogisticRegression: 71.00%
6. AdaBoostClassifier: 71.00%
7. RandomForestClassifier: 70.00%
8. BaggingClassifier: 70.00%
9. ExtraTreesClassifier: 70.00%
10. DecisionTreeClassifier: 67.00%
11. KNeighborsClassifier: 58.50%
```

```
# Naive Bayes Classifier for Text Classification

# In this code, a Multinomial Naive Bayes classifier is used for text
```

```
classification.

# Import the Multinomial Naive Bayes classifier from scikit-learn.
from sklearn.naive_bayes import MultinomialNB

# Create an instance of the Multinomial Naive Bayes classifier.
clf = MultinomialNB()

# Train (fit) the classifier using the training data.
# 'X_train' contains the features (BoW representation of text data).
# 'y_train' contains the corresponding labels or target values.
clf.fit(X_train, y_train)

# At this point, the Multinomial Naive Bayes classifier 'clf' has been
trained on the training data.

# Naive Bayes classifiers are commonly used in NLP tasks such as text
classification (e.g., spam detection, sentiment analysis).
# They are known for their simplicity and effectiveness when dealing
with text data.
```

```
MultinomialNB()
```

```
# Predicting the Test set results

y_pred = clf.predict(X_test)
y_pred
```

```
array([0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0,
0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,
0,
       0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
0,
       1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1,
0,
       1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0,
0,
       0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1,
1,
       0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1,
1,
       1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
1,
       0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1,
1,
       0, 1])
```

```
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix, classification_report
```

```
y_true = y_test

accuracy = accuracy_score(y_true, y_pred)
precision = precision_score(y_true, y_pred)
recall = recall_score(y_true, y_pred)
f1 = f1_score(y_true, y_pred)

# Convert the metrics to percentages
accuracy_percent = accuracy * 100
precision_percent = precision * 100
recall_percent = recall * 100
f1_percent = f1 * 100

# Print the metrics as percentages
print("Accuracy: {:.2f}%".format(accuracy_percent))
print("Precision: {:.2f}%".format(precision_percent))
print("Recall: {:.2f}%".format(recall_percent))
print("F1 Score: {:.2f}%".format(f1_percent))
```

```
Accuracy: 76.50%
Precision: 76.42%
Recall: 78.64%
F1 Score: 77.51%
```

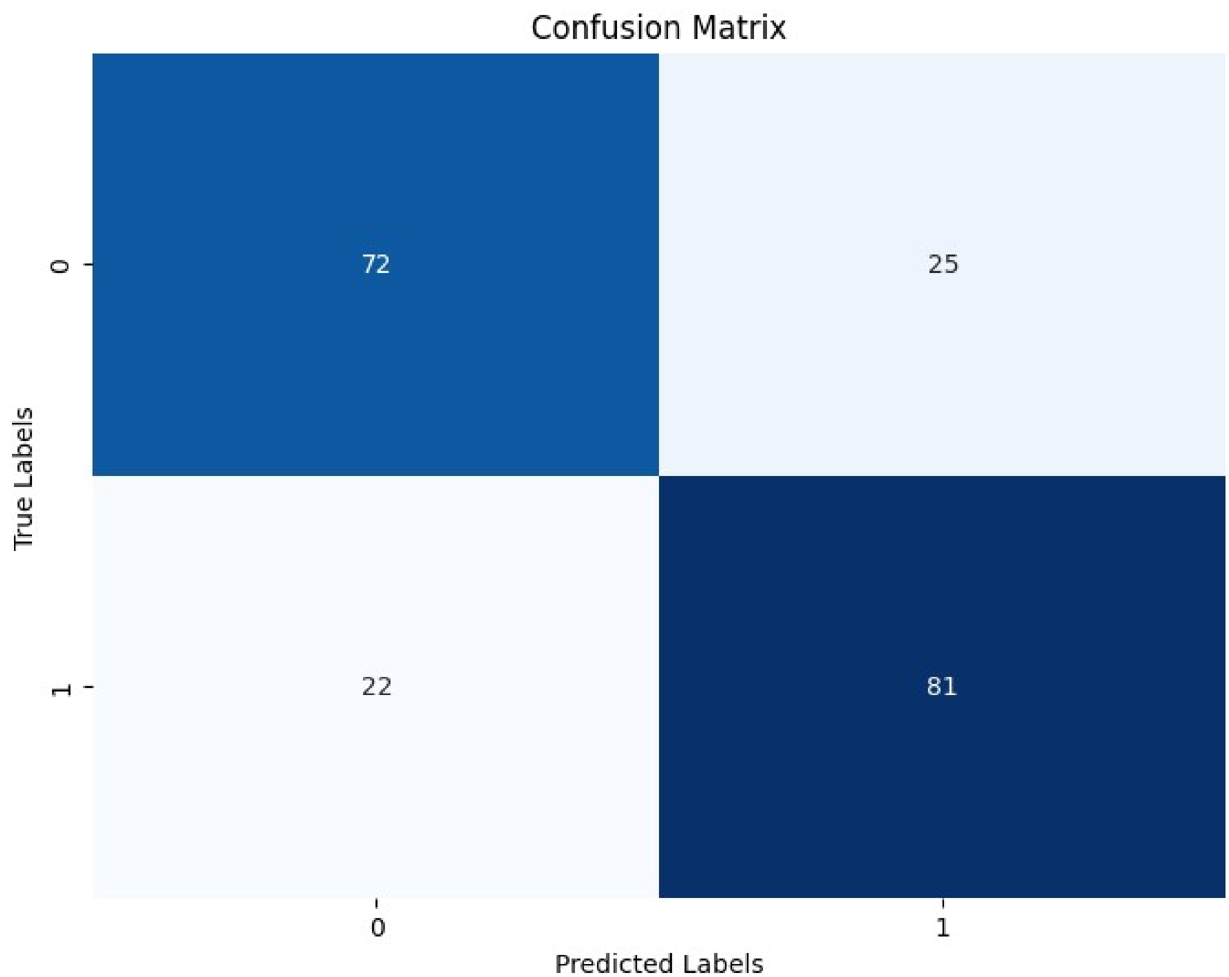
```
# Confusion Matrix: A table showing the count of true positives, true
negatives, false positives, and false negatives
conf_matrix = confusion_matrix(y_true, y_pred)
print("Confusion Matrix:\n", conf_matrix)
```

```
Confusion Matrix:
 [[72 25]
 [22 81]]
```

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Assuming you have calculated the confusion matrix
conf_matrix = confusion_matrix(y_true, y_pred)

# Create a heatmap to visualize the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",
cbar=False)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```




```
# Classification Report: Provides precision, recall, F1-score, and
support for each class
class_report = classification_report(y_true, y_pred)
print("Classification Report:\n", class_report)

Classification Report:
               precision    recall  f1-score   support

           0       0.77      0.74      0.75        97
           1       0.76      0.79      0.78       103

    accuracy                           0.77       200
   macro avg       0.77      0.76      0.76       200
weighted avg       0.77      0.77      0.76       200


from sklearn.model_selection import GridSearchCV
from sklearn.naive_bayes import MultinomialNB

# Create a parameter grid to specify the hyperparameters to tune
```

```python
param_grid = {
    'alpha': [0.1, 0.5, 1.0],  # Smoothing parameter (Laplace
smoothing)
    'fit_prior': [True, False]  # Whether to learn class prior
probabilities
}

# Create a Multinomial Naive Bayes classifier
nb_classifier = MultinomialNB()

# Create a GridSearchCV object with the classifier and parameter grid
grid_search = GridSearchCV(estimator=nb_classifier,
param_grid=param_grid, cv=5, scoring='accuracy')

# Fit the GridSearchCV object to your training data
grid_search.fit(X_train, y_train)

# Get the best hyperparameters from the grid search
best_params = grid_search.best_params_

# Get the best model from the grid search
best_classifier = grid_search.best_estimator_

# Print the best hyperparameters
print("Best Hyperparameters:", best_params)

# Evaluate the best model on your testing data
y_pred_best = best_classifier.predict(X_test)

# Calculate and print metrics for the best model (e.g., accuracy,
precision, recall, F1-score)
accuracy_best = accuracy_score(y_test, y_pred_best)
precision_best = precision_score(y_test, y_pred_best)
recall_best = recall_score(y_test, y_pred_best)
f1_best = f1_score(y_test, y_pred_best)

print("Best Model Metrics:")
print("Accuracy:", accuracy_best)
print("Precision:", precision_best)
print("Recall:", recall_best)
print("F1 Score:", f1_best)
```

```
Best Hyperparameters: {'alpha': 0.5, 'fit_prior': True}
Best Model Metrics:
Accuracy: 0.775
Precision: 0.7735849056603774
Recall: 0.7961165048543689
F1 Score: 0.7846889952153109
```

```python
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
```

```python
import numpy as np

# Define a range of alpha values to test
alpha_values = np.arange(0.1, 1.1, 0.1)

best_accuracy = 0.0
best_alpha = None

# Loop through each alpha value
for alpha in alpha_values:
    # Create a new Multinomial Naive Bayes classifier with the current
alpha
    classifier = MultinomialNB(alpha=alpha)

    # Train the classifier on the training data
    classifier.fit(X_train, y_train)

    # Make predictions on the testing data
    y_pred = classifier.predict(X_test)

    # Calculate the accuracy score
    accuracy = accuracy_score(y_test, y_pred)

    # Print the accuracy for the current alpha
    print("Accuracy score for alpha-{} is: {:.2f}
%".format(round(alpha, 1), accuracy * 100))

    # Check if this alpha gives a better accuracy
    if accuracy > best_accuracy:
        best_accuracy = accuracy
        best_alpha = alpha

print("\nOptimal hyperparameters:")
print("Best accuracy: {:.2f}%".format(best_accuracy * 100))
print("Best alpha: {}".format(round(best_alpha, 1)))
```

```
Accuracy score for alpha-0.1 is: 78.00%
Accuracy score for alpha-0.2 is: 78.50%
Accuracy score for alpha-0.3 is: 78.00%
Accuracy score for alpha-0.4 is: 78.00%
Accuracy score for alpha-0.5 is: 77.50%
Accuracy score for alpha-0.6 is: 77.50%
Accuracy score for alpha-0.7 is: 77.50%
Accuracy score for alpha-0.8 is: 77.00%
Accuracy score for alpha-0.9 is: 76.50%
Accuracy score for alpha-1.0 is: 76.50%

Optimal hyperparameters:
Best accuracy: 78.50%
Best alpha: 0.2
```

```
clf = MultinomialNB(alpha=0.2)
classifier.fit(X_train,y_train)
```

```
MultinomialNB()
```

Predictions

```
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

def predict_sentiment(sample_review, classifier, cv):
    # Preprocess the sample review
    sample_review = re.sub(pattern='[^a-zA-Z]', repl=' ',
string=sample_review)
    sample_review = sample_review.lower()
    sample_review_words = sample_review.split()
    sample_review_words = [word for word in sample_review_words if not
word in set(stopwords.words('english'))]
    ps = PorterStemmer()
    final_review = [ps.stem(word) for word in sample_review_words]
    final_review = ' '.join(final_review)

    # Transform the preprocessed review using the CountVectorizer (cv)
    temp = cv.transform([final_review]).toarray()

    # Use the pre-trained classifier to predict sentiment
    sentiment = classifier.predict(temp)

    return sentiment[0]  # Return the predicted sentiment (assuming
it's a single value)

# Sample reviews as strings
reviews = [
    'The food is really bad.',
    'I love their delicious dishes!',
    'Terrible experience. Avoid this place.',
    'The service was excellent.',
    'Worst place ever, but nice food'
]

# Assuming you have already defined the 'predict_sentiment' function,
classifier, and cv

for review in reviews:
    sentiment = predict_sentiment(review, classifier, cv)
    if sentiment:
        sentiment_label = 'POSITIVE'
    else:
```

```python
        sentiment_label = 'NEGATIVE'

    print(f"Review: '{review}'")
    print(f"Sentiment: {sentiment_label}")
    print()
```

```
Review: 'The food is really bad.'
Sentiment: NEGATIVE

Review: 'I love their delicious dishes!'
Sentiment: POSITIVE

Review: 'Terrible experience. Avoid this place.'
Sentiment: NEGATIVE

Review: 'The service was excellent.'
Sentiment: POSITIVE

Review: 'Worst place ever, but nice food'
Sentiment: NEGATIVE
```