In [1]:

```
# Step 1 - Import all the libraries

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.stem.porter import PorterStemmer
nltk.download('stopwords')
from nltk.corpus import stopwords
STOPWORDS = set(stopwords.words('english'))

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import accuracy_score
from wordcloud import WordCloud
import pickle
import re
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.stem import PorterStemmer
import regex as re
import string
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\swaro\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [2]:

```
%pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\interr .ai\lib\site-packages (1.9.3)
Requirement already satisfied: numpy>=1.6.1 in c:\interr .ai\lib\site-packages (from
wordcloud) (1.26.4)
Requirement already satisfied: pillow in c:\interr .ai\lib\site-packages (from wordc
loud) (10.3.0)
Requirement already satisfied: matplotlib in c:\interr .ai\lib\site-packages (from w
ordcloud) (3.8.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\interr .ai\lib\site-packages
(from matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\interr .ai\lib\site-packages (from
matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\interr .ai\lib\site-packages
(from matplotlib->wordcloud) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\interr .ai\lib\site-packages
(from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\interr .ai\lib\site-packages (f
rom matplotlib->wordcloud) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in c:\interr .ai\lib\site-packages
(from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\interr .ai\lib\site-packag
es (from matplotlib->wordcloud) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\interr .ai\lib\site-packages (from pyt
hon-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
Data=pd.read_csv(r"C:\Users\swaro\OneDrive\Desktop\data science swaroop ky 1\Projec
print(Data)
```

```
        rating_review                                        review_full
0                   5  Totally in love with the Auro of the place, re...
1                   5  I went this bar 8 days regularly with my husba...
2                   5  We were few friends and was a birthday celebra...
3                   5  Fatjar Cafe and Market is the perfect place fo...
4                   5  Hey Guys, if you are craving for pizza and sea...
...               ...                                                ...
147576              5  Near by airport very calm and cool environment...
147577              5  My favourite place to stay. Great service. Ash...
147578              5  good food with nice decoration, drinks list al...
147579              4  Near to airport .it is fine property.  Staff i...
147580              5  Amazing food ..  Excellent ambience ..  Great ...

[147581 rows x 2 columns]
```

In [4]:
```
# step 2
Data.shape
```

Out[4]:
```
(147581, 2)
```

In [5]:
```
# Coloumns
Data.columns
```

Out[5]:
```
Index(['rating_review', 'review_full'], dtype='object')
```

In [6]:
```
#Check for null values

Data.isnull().sum()
```

```
Out[6]: rating_review    0
        review_full      2
        dtype: int64
```

```
In [7]: #We will drop the null record

        Data.dropna(inplace=True)
```

```
In [9]: # shape after dropping null values
        Data.shape
```

```
Out[9]: (147579, 2)
```

```
In [10]: #Creating a new column 'length' that will contain the length of the string in 'veri
         Data['length'] = Data['review_full'].apply(len)
         Data.head()
```

Out[10]:

	rating_review	review_full	length
0	5	Totally in love with the Auro of the place, re...	720
1	5	I went this bar 8 days regularly with my husba...	202
2	5	We were few friends and was a birthday celebra...	144
3	5	Fatjar Cafe and Market is the perfect place fo...	435
4	5	Hey Guys, if you are craving for pizza and sea...	533

Datatypes of the features

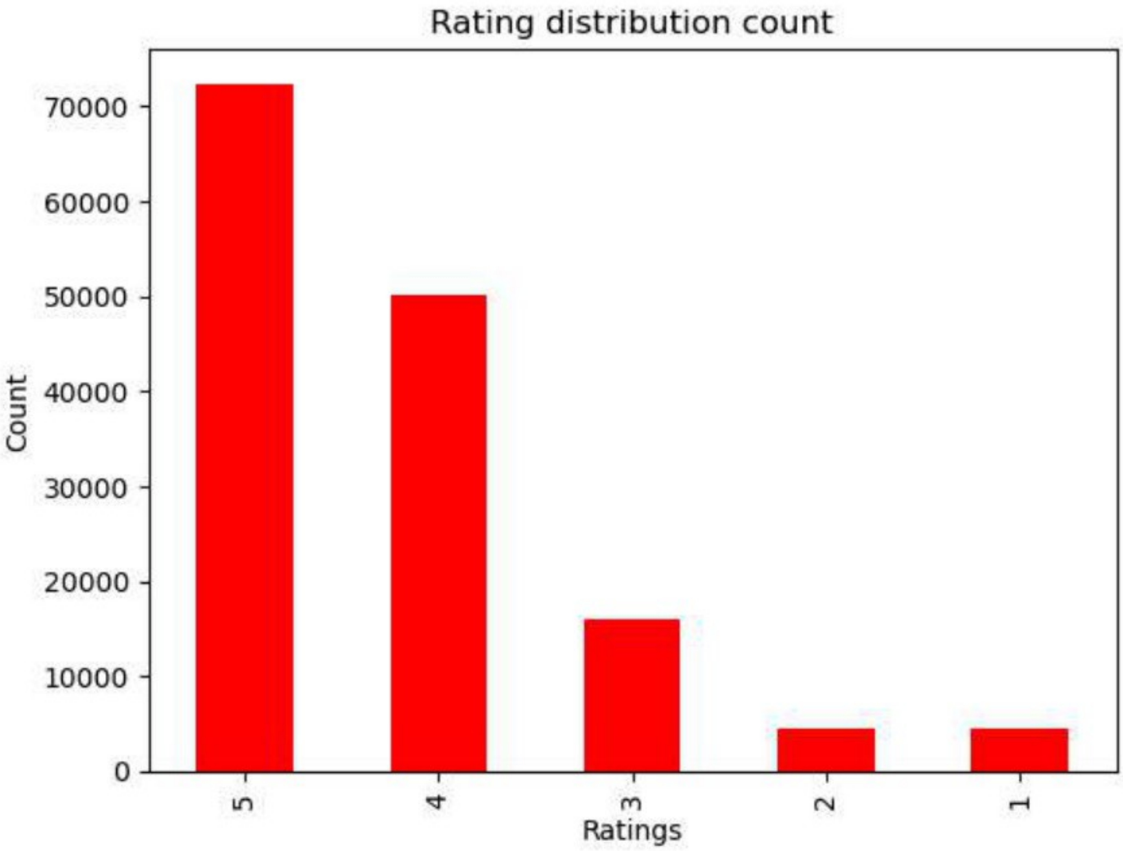
```
In [11]: # step 3
         Data.dtypes
```

```
Out[11]: rating_review     int64
         review_full      object
         length            int64
         dtype: object
```

```
In [13]:  #Step 4 #Distinct values of 'rating' and its count

         print(f"Rating value count: \n{Data["rating_review"].value_counts()}")
```

```
Rating value count:
rating_review
5    72389
4    50248
3    15935
2     4552
1     4455
Name: count, dtype: int64
```

```
In [14]: #Bar plot to visualize the total counts of each rating

         Data["rating_review"].value_counts().plot.bar(color = 'red')
         plt.title('Rating distribution count')
         plt.xlabel('Ratings')
plt.ylabel('Count')
plt.show()
```



In [15]:

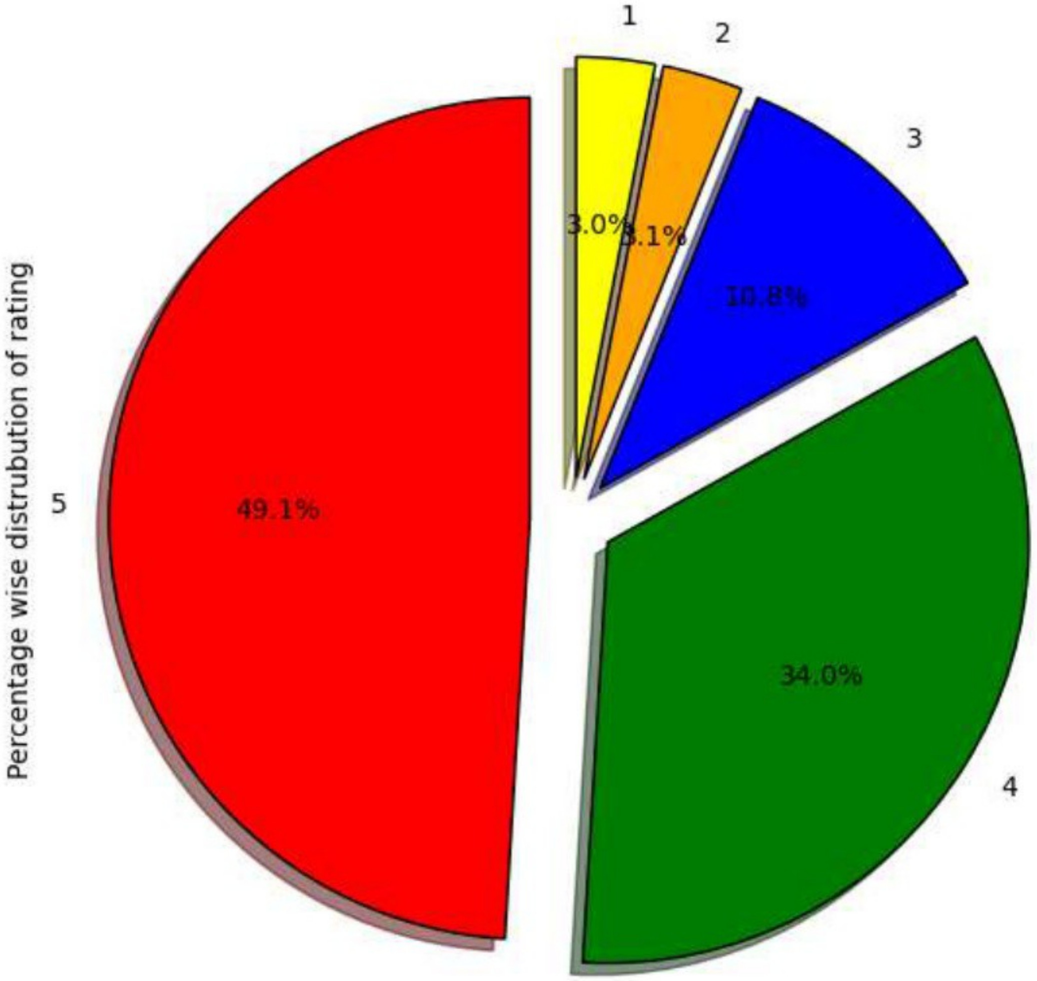
```
#Finding the percentage distribution of each rating

print(f"Rating value count - percentage distribution: \n{round(Data['rating_review'
```

```
Rating value count - percentage distribution:
rating_review
5    49.05
4    34.05
3    10.80
2     3.08
1     3.02
Name: count, dtype: float64
```

In [16]:

```
fig = plt.figure(figsize=(7,7))

colors = ('red', 'green', 'blue','orange','yellow')

wp = {'linewidth':1, "edgecolor":'black'}

tags = Data["rating_review"].value_counts()/Data.shape[0]

explode=(0.1,0.1,0.1,0.1,0.1)

tags.plot(kind='pie', autopct="%1.1f%%", shadow=True, colors=colors, startangle=90,

from io import  BytesIO
```

```
graph = BytesIO()

fig.savefig(graph, format="png")
```



In [17]:

```
# Analyzing 'review_full' column
Data[ "length"].describe()
```

Out[17]:

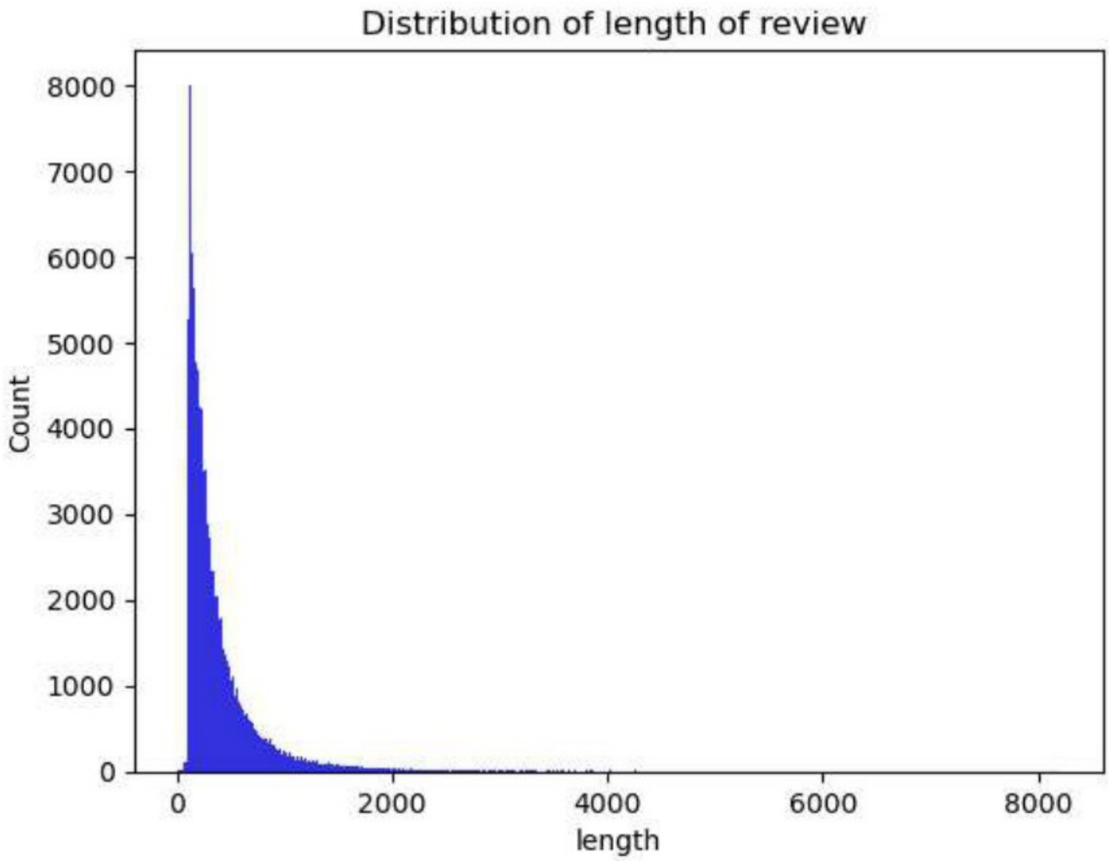
```
count    147579.000000
mean        404.745445
std         463.877369
min           6.000000
25%         163.000000
50%         263.000000
75%         462.000000
max        8192.000000
Name: length, dtype: float64
```

In [18]:

```
sns.histplot(Data['length'],color='blue').set(title='Distribution of length of revi
```

Out[18]:

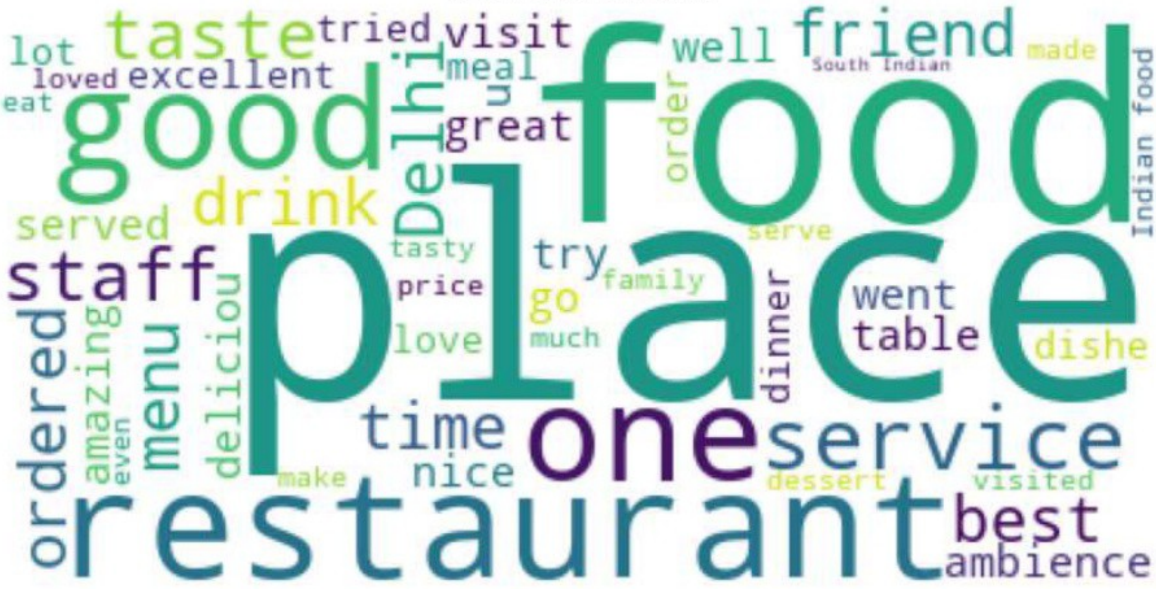
```
[Text(0.5, 1.0, 'Distribution of length of review ')]
```



In [19]:

```
# Combine all reviews
reviews = " ".join([review for review in Data['review_full']])

# Initialize wordcloud object
wc = WordCloud(background_color='white', max_words=50)

# Generate and plot wordcloud
plt.figure(figsize=(10,10))
plt.imshow(wc.generate(reviews))
plt.title('Wordcloud for all reviews', fontsize=10)
plt.axis('off')
plt.show()
```

Wordcloud for all reviews

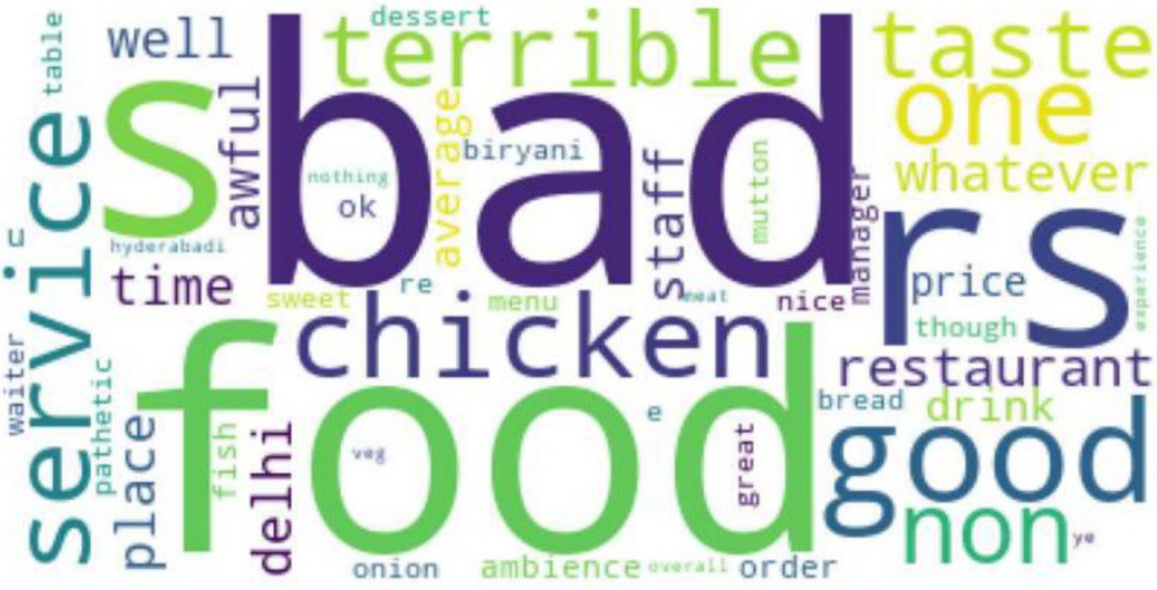


In [20]:
```
neg_reviews = " ".join([review for review in Data[Data['review_full'].str.contains(
neg_reviews = neg_reviews.lower().split()

# Combine all positive reviews into a single string
pos_reviews = " ".join([review for review in Data[~Data['review_full'].str.contains
pos_reviews = pos_reviews.lower().split()
```

In [21]:
```
# Assuming neg_reviews and pos_reviews are lists of words from the reviews

# Convert lists to sets for unique words
set_neg_reviews = set(neg_reviews)
set_pos_reviews = set(pos_reviews)

# Find unique negative and positive words using set difference
unique_negative = set_neg_reviews - set_pos_reviews
unique_positive = set_pos_reviews - set_neg_reviews

# Join the unique words back into strings
unique_negative = " ".join(unique_negative)
unique_positive = " ".join(unique_positive)
```

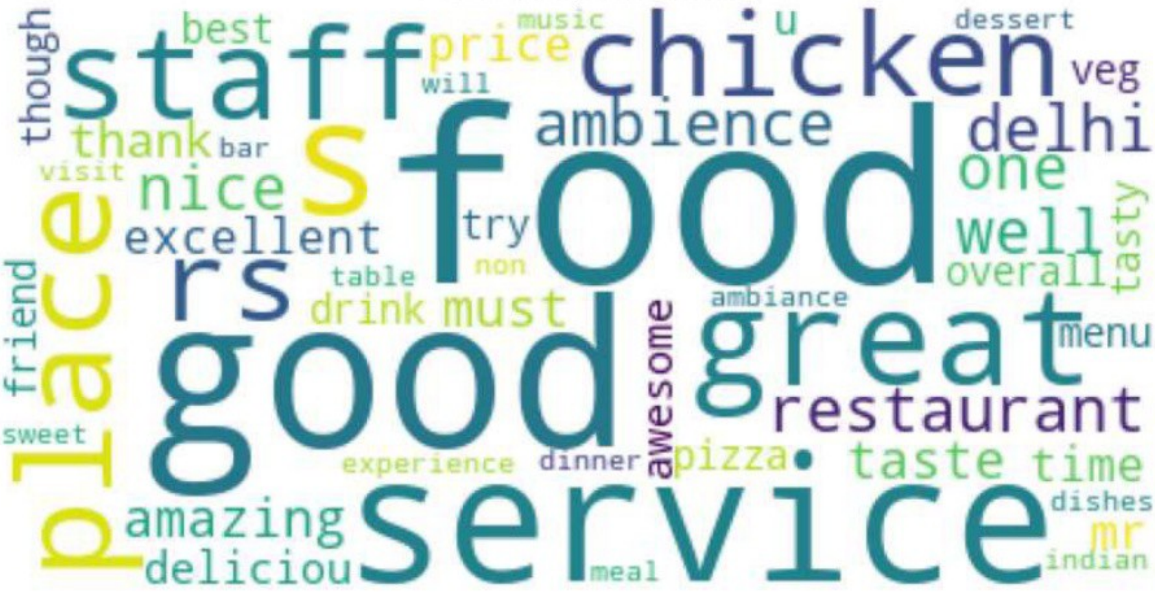
In [22]:
```
wc = WordCloud(background_color='white', max_words=50)

# Generate and plot wordcloud
plt.figure(figsize=(10,10))
plt.imshow(wc.generate(unique_negative))
plt.title('Wordcloud for negative reviews', fontsize=10)
plt.axis('off')
plt.show()
```

Wordcloud for negative reviews



```
In [23]: wc = WordCloud(background_color='white', max_words=50)

         # Generate and plot wordcloud
         plt.figure(figsize=(10,10))
         plt.imshow(wc.generate(unique_positive))
         plt.title('Wordcloud for positive reviews', fontsize=10)
         plt.axis('off')
         plt.show()
```

Wordcloud for positive reviews



Preprocessing and Modelling

```
In [24]: from nltk.stem import WordNetLemmatizer
         # Initialize lemmatizer
         lemmatizer = WordNetLemmatizer()
         stop_words = set(stopwords.words('english'))

         def clean_text(text):
```

```
    # Lowercase
    text = text.lower()
    # Remove punctuation
    text = re.sub(r'[^\w\s]', '', text)
    # Remove stop words and Lemmatize
    text = ' '.join(lemmatizer.lemmatize(word) for word in text.split() if word not
    return text

# Apply cleaning function
Data['cleaned_reviews'] = Data['review_full'].apply(clean_text)
```

In [25]:

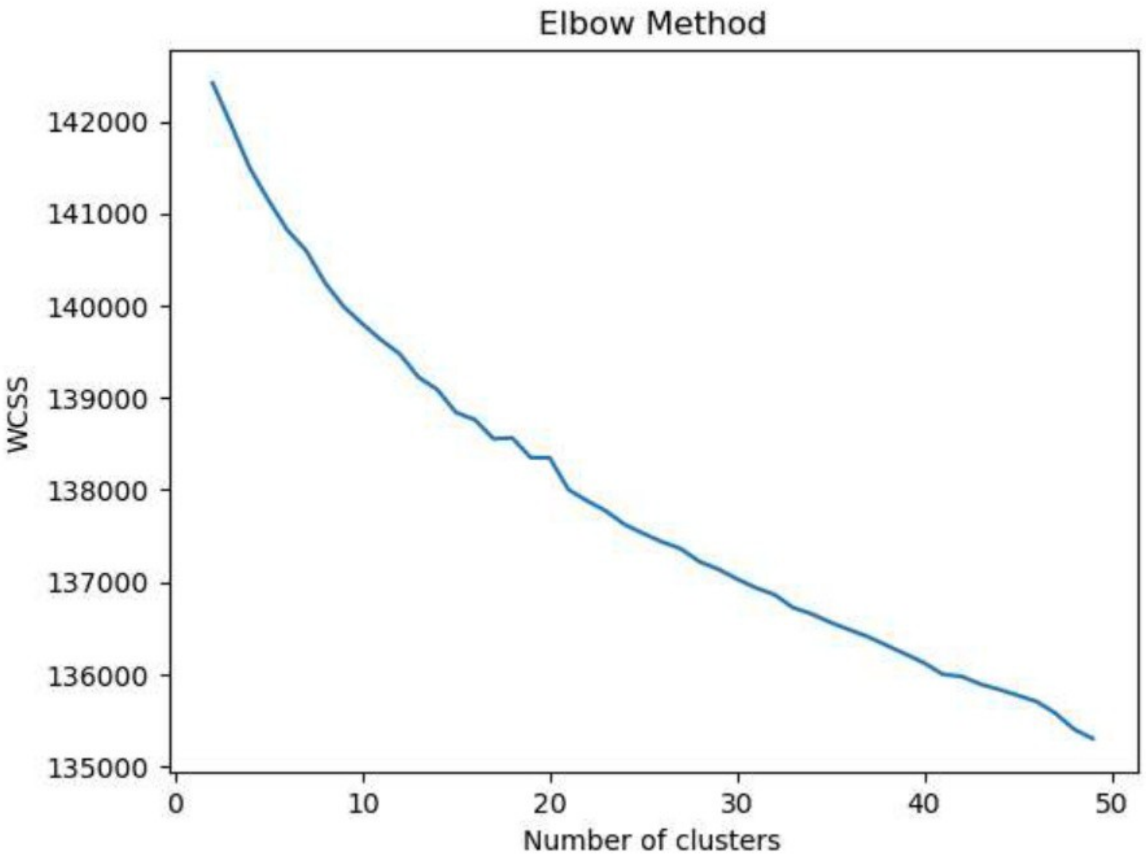
```
from sklearn.feature_extraction.text import TfidfVectorizer

# Initialize TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer(max_features=5000)  # Limit features for perform
tfidf_matrix = tfidf_vectorizer.fit_transform(Data['cleaned_reviews'])
```

In [26]:

```
from sklearn.cluster import KMeans
wcss = []
for i in range(2, 50):  # Adjusting the range to be more efficient
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(tfidf_matrix)
    wcss.append(kmeans.inertia_)
```

In [27]:

```
plt.plot(range(2, 50), wcss)  # Adjusting the range accordingly
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [28]:

```python
# Fit KMeans with a chosen number of clusters, e.g., 5
optimal_clusters = 20
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
Data['cluster'] = kmeans.fit_predict(tfidf_matrix)
```

In [29]:

```python
# Get feature names
feature_names = tfidf_vectorizer.get_feature_names_out()

# Function to get top words for each cluster
def get_top_words_per_cluster(model, n_words=10):
    order_centroids = model.cluster_centers_.argsort()[:, ::-1]
    top_words = {}
    for i in range(optimal_clusters):
        top_words[i] = [feature_names[ind] for ind in order_centroids[i, :n_words]]
    return top_words

top_words = get_top_words_per_cluster(kmeans)

# Display top words for each cluster
for cluster, words in top_words.items():
    print(f"Cluster {cluster}: {', '.join(words)}")
```

```
Cluster 0: price, reasonable, food, good, place, service, quality, restaurant, high,
great
Cluster 1: food, service, great, good, one, time, taste, place, staff, dish
Cluster 2: excellent, service, food, great, good, staff, restaurant, place, ambienc
e, nice
Cluster 3: amazing, food, place, great, service, staff, experience, ambience, reall
y, visit
Cluster 4: quick, service, food, good, place, bite, great, nice, staff, restaurant
Cluster 5: good, food, service, nice, place, ambience, also, really, staff, quality
Cluster 6: place, food, great, visit, good, nice, friend, one, must, best
Cluster 7: south, indian, food, dosa, place, good, authentic, restaurant, delhi, nor
th
Cluster 8: pizza, pasta, good, place, cheese, domino, crust, italian, food, great
Cluster 9: restaurant, food, delhi, good, service, one, indian, great, hotel, best
Cluster 10: music, place, good, live, food, great, nice, loud, ambience, drink
Cluster 11: beer, good, place, food, nice, great, service, bar, one, menu
Cluster 12: chicken, butter, place, good, food, dish, tikka, ordered, try, one
Cluster 13: biryani, mutton, kebab, chicken, karims, place, food, good, taste, try
Cluster 14: dal, paneer, makhani, bukhara, naan, food, good, tikka, makhni, place
Cluster 15: veg, non, food, good, place, restaurant, dish, also, vegetarian, taste
Cluster 16: burger, good, place, fry, shake, chicken, food, try, one, taste
Cluster 17: drink, bar, place, good, great, food, cocktail, nice, service, friend
Cluster 18: breakfast, buffet, good, service, staff, great, food, hotel, dinner, nic
e
Cluster 19: thanks, mr, special, service, great, food, team, staff, hospitality, goo
d
```

In [30]:

```
pip install textblob
```

```
Requirement already satisfied: textblob in c:\interr .ai\lib\site-packages (0.18.0.p
ost0)
Requirement already satisfied: nltk>=3.8 in c:\interr .ai\lib\site-packages (from te
xtblob) (3.8.1)
Requirement already satisfied: click in c:\interr .ai\lib\site-packages (from nltk>=
3.8->textblob) (8.1.7)
Requirement already satisfied: joblib in c:\interr .ai\lib\site-packages (from nltk>
=3.8->textblob) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\interr .ai\lib\site-packages (f
rom nltk>=3.8->textblob) (2023.10.3)
Requirement already satisfied: tqdm in c:\interr .ai\lib\site-packages (from nltk>=
3.8->textblob) (4.66.4)
Requirement already satisfied: colorama in c:\interr .ai\lib\site-packages (from cli
ck->nltk>=3.8->textblob) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

In [31]:

```
from textblob import TextBlob

# Function to calculate sentiment polarity
def get_sentiment(text):
    return TextBlob(text).sentiment.polarity

# Apply sentiment analysis on cleaned reviews
Data['sentiment'] = Data['cleaned_reviews'].apply(get_sentiment)

# Calculate average sentiment for each cluster
cluster_sentiment = Data.groupby('cluster')['sentiment'].mean().reset_index()
```

```
# Define a function to label clusters based on average sentiment
def label_cluster(row):
    if row['sentiment'] > 0:
        return 'Positive'
    elif row['sentiment'] < 0:
        return 'Negative'
    else:
        return 'Neutral'

# Apply the labeling function
cluster_sentiment['label'] = cluster_sentiment.apply(label_cluster, axis=1)

# Display the average sentiment and labels for each cluster
print(cluster_sentiment)
```

```
    cluster  sentiment     label
0         0   0.315965  Positive
1         1   0.285559  Positive
2         2   0.577745  Positive
3         3   0.475964  Positive
4         4   0.360778  Positive
5         5   0.481366  Positive
6         6   0.385212  Positive
7         7   0.394815  Positive
8         8   0.325297  Positive
9         9   0.317628  Positive
10       10   0.384022  Positive
11       11   0.277481  Positive
12       12   0.129191  Positive
13       13   0.266642  Positive
14       14   0.343193  Positive
15       15   0.356666  Positive
16       16   0.293418  Positive
17       17   0.354854  Positive
18       18   0.420457  Positive
19       19   0.471944  Positive
```

In [32]:

```
# Function to calculate sentiment polarity
def get_sentiment(text):
    return TextBlob(text).sentiment.polarity

# Apply sentiment analysis on cleaned reviews
Data['sentiment'] = Data['cleaned_reviews'].apply(get_sentiment)

# Filter for negative reviews
negative_reviews = Data[Data['sentiment'] < 0]

# Calculate average sentiment for each cluster with negative reviews
cluster_sentiment = negative_reviews.groupby('cluster')['sentiment'].mean().reset_i

# Define a function to label clusters based on average sentiment
def label_cluster(row):
    if row['sentiment'] < 0:
        return 'Negative'
    else:
        return 'Not Applicable'  # Since we're only focusing on negative reviews

# Apply the labeling function
cluster_sentiment['label'] = cluster_sentiment.apply(label_cluster, axis=1)

# Display the average sentiment and labels for each cluster
print(cluster_sentiment)
```

```
    cluster  sentiment     label
0         0  -0.145982  Negative
1         1  -0.172655  Negative
2         2  -0.122959  Negative
3         3  -0.083245  Negative
4         4  -0.094006  Negative
5         5  -0.147875  Negative
6         6  -0.168441  Negative
7         7  -0.144348  Negative
8         8  -0.153509  Negative
9         9  -0.158714  Negative
10       10  -0.134783  Negative
11       11  -0.127412  Negative
12       12  -0.154380  Negative
13       13  -0.149298  Negative
14       14  -0.149447  Negative
15       15  -0.132757  Negative
16       16  -0.166638  Negative
17       17  -0.156612  Negative
18       18  -0.156940  Negative
19       19  -0.170526  Negative
```

In [33]:

```
# Function to get top positive and negative reviews for each cluster
def get_top_reviews_per_cluster(data, cluster_labels):
    top_reviews = {}
    for label in ['Positive', 'Negative']:
        top_reviews[label] = {}
        for cluster in data['cluster'].unique():
            cluster_data = data[data['cluster'] == cluster]
            if label == 'Positive':
                top_reviews[label][cluster] = cluster_data[cluster_data['sentiment'
            else:
                top_reviews[label][cluster] = cluster_data[cluster_data['sentiment'
    return top_reviews

# Get top positive and negative reviews for each cluster
top_reviews = get_top_reviews_per_cluster(Data, cluster_sentiment)

# Display the top positive and negative reviews
for label, cluster_reviews in top_reviews.items():
    print(f"Top {label} Reviews:")
    for cluster, reviews in cluster_reviews.items():
        print(f"Cluster {cluster}:")
        for review in reviews:
            print(review)
        print()
```

```
Top Positive Reviews:
Cluster 6:
Totally in love with the Auro of the place, really beautiful and quite fancy at the 
same time. The ambience is very pure and gives a sense of positivity throughout. Out
door and indoor interior are quite quaint and cute. Love the open kitchen idea and t
here whole marketplace ideology. Due to coronovirus they specifically use disposable
cutlery to keep the pandemic in mind taking all the precautionary measures from the 
beginning of the place with the mask on their staff and using good sanitisation. The
food is really amazing specially the pizza straight from the oven and the hummus and
pita bread are quite delicious too. If you're looking for a classy yet soothing Ital
ian place in Delhi,Fatjar is a go to for you!
Fatjar Cafe and Market is the perfect place for a casual lunch with your loved ones.
The Ambience is delightful and the food is totally the best thing about the place! Y
Ou can purchase your choosing of Olives and a lot more directly from the place. They
are taking all the prevention measures for the spread of COVID-19 and I can say this
without hesitation that it is by far the most safest place felt after looking at the
ir actions ✨
If you are looking for a varied selection of delicious, high quality dishes, this is
your place. We spent three weeks traveling around India, and our best meals of the t
rip were unquestionably found here. Great place if you are tiring of the typical Ind
ian dishes and are lusting for some ethnic variety. Oh, and do yourself a favor: You
MUST get the Sticky Date Pudding with Vanilla Bean Ice Cream. It may be the best des
sert we have ever had.
was staying near by so visited this place. huge place with loads of options to eat a
nd has a small deli too. ordered for the pesto mozzarella pesto sandwich which turne
d out to be very good. shot of espresso was good too. if budget isn't a problem, thi
s place is not to be missed for the quality of the ingredients they use.
Fat Jar is one of the most hygienic and beautiful places in Delhi NCR.Pizza and coff
ee are so good here. I would also recommend the in house tea which is served hot and
is delicous.

Cluster 17:
I went this bar 8 days regularly with my husband.. we are fully satisfied by the ser
vices. Staff is very good (vitoni and amen) they both served us daily . Will surely 
visit again. Highly recommended ❣️
We had drinks during happy hour, good selection & excellent service at the bar. We d
idn’t eat but the food looked really good.
Great service with quite polite and generous staff. Must hangout at hanger. I was se
rved by Mr. Shubham very cooperative and friendly person. The place is very neat and
clean and follows all rules of safety and cleanliness. Great food and great Drinks. 
Cheers!
Beautiful and hygienic bar with great service by Ashwin... A smart guy with lots of 
talent... Love to visit again!!!
The hotel and staff are lovely and extremely attentive. The hot drinks and snacks ar
e very nice indeed.  Had a lovely stay in the hotel and I was served by a very lovel
y staff member by the name of Laxmi.  Thsank you

Cluster 5:
We were few friends and was a birthday celebration. The food was so good in taste an
d it was really fresh. We all loved it and highly recomment.
Emmanuel is a good stirrer and shaker. They taste great and he is very knowledgeabl
e. A good place to relax.
Superb , fantastic , good service..food was tasty , Tarun and Akash was really good 
in service and very polite too...
Very good service as well as food....Nice experience....Staff is very supportive and
cooperative...
```

```
Really good atmosphere and good service from the staff. Especially surendra, really
helpful through out the match even at such a late time.

Cluster 8:
Hey Guys, if you are craving for pizza and searching for it then you should visit th
is cafe. Yes, I highly recommend you to visit this FatJar Cafe because the cafe offe
rs scrumptious pizza. I have ordered Veggie Pizza which was really awesome. Apart fr
om that, I have also ordered Cappuccino and one soft drink. If I talk about the ambi
ance then no doubt, because it is absolutely pretty. From my side I'll give 5/5 for
this cafe. It is completely amazing. I had a great experience with this cafe. Here a
re some pictures have a look
Kailash colony is brimming with small cafes now. I happened to visit this beauty for
lunch and was amazed with the 1st impression it made.  We started with Falafel & Pit
a and Margherita Pizza. Safe to say, Margherita pizza was the best choice of the da
y. Its made freshly in front of you in the Wood Fired oven. The taste was exquisite
and I would recommend this to anybody who plans to visit this outlet. I could come h
ere just to have that Pizza. Hummus, Pita and Falafel were a good side dish while we
ordered the next couple of dishes. We moved on to order Fussili Alfredo. This white
sauce fussili pasta was good but could not be compared to the pizza we had earlier.
In the end, we went for Frozen Yoghurt with Honey. It is recommended to have it with
honey, which will do wonders to the frozen yoghurt!  This place turned out to be qui
te peaceful and the plants enhanced its beauty.
Kailash Colony is the place I always head to when I’m not in the mood for GK & Nehru
Place’s hustle bustle. This time around we planned to visit Fatjar Café and Market w
hich is right on the road and was very easy to find.   Fatjar’s ambience is very ple
asing to the eyes and the light colours used makes it look very airy. The use of pla
nts further makes this place look more beautiful. Most of the cooking is done in fro
nt of you with a semi open kitchen and their food fire oven looks amazing too. The p
roducts on display establishes the word “market” in the name perfectly. The varied s
eating options are very comfortable and a small outdoor seating is great for sunny w
inter lunches. The parking though is limited in front of the café.   We started our
meal with some “Falafel & Pita” which had some moist falafels which were nicely frie
d and not at all oily. The pita breads were very fresh and so was the hummus. You co
uld literally tell that that just after your first bite. The consistency of the humm
us was noteworthy too. Next we ordered some humble “Truffle Fries” which were very c
rispy and a pinch of truffle to top them made them taste even better.   For the main
course, we ordered a “Fussili Alfredo” which was very creamy and the pasta was cooke
d to perfection. The only thing they need to look at is that usually the pasta is su
pposed to be served along with a bread which they don’t. The highlight of our meal w
as simply their “Margherita Pizza” which had a brilliant base made out of the freshe
st dough you could think of. The cheese used to top the pizza looked very premium an
d the balance between the sauces and the cheese was tremendous too. The presentation
deserves a special mention as it wasn’t the usual pizza stand they serve their pizza
on but is more a raised steel stand which again was a nice touch.   To accompany our
meal, we ordered a “Berries Shakes” which was very thick and was prepared using real
berries instead of the preservatives filled syrups. Though the sweetness was a bit t
oo low for our liking and a little more sweetness would have made that shake perfec
t.   We ended our meal with a “Frozen Yoghurt with Honey” which had two scoops of fr
ozen yoghurt which had some ice crystals too and once you top it up with honey, the
combination really does wonders to the taste.   The service was top notch and the st
ewards were very considerate about what we were ordering and didn’t shy away from re
commending their favourites and also explaining the dishes on the menu. Overall, Fat
jar simply amazed us with its varied food options on the menu and its serene ambienc
e. The place looked fantastic and is perfect for a romantic date or co working or if
you are looking forward to a quiet and a hearty meal.
The service was great. this young gentlemen Suraj gave us a awesome service. The piz
```