Project Name

Culinary Sentiments: Analysing Zomato's Customer Reviews

Project-Type: Unsupervised
Contribution – Individual
Name – Charu Kumar

"Torture the data, and it will confess to anything." – Ronald Coase

PROJECT SUMMARY:

In this project, we performed sentiment analysis on customer reviews from the Zomato dataset using Python and Natural Language Processing (NLP) techniques. The primary goals of this project were to understand sentiment analysis, process text data, and classify customer reviews into positive or negative sentiments.

Key Project Steps and Concepts:

1. Data Collection: We started by collecting a dataset of customer reviews from Zomato, which included text data and associated ratings or sentiments provided by customers.
2. Sentiment Analysis in Python: We learned how to perform sentiment analysis in Python, a natural language processing (NLP) technique that involves determining the sentiment (positive, negative, or neutral) expressed in a piece of text.

3. Text Sentiment Analysis: Text sentiment analysis involves analyzing text data, extracting features, and using machine learning models to classify the sentiment expressed in the text. We used Python libraries such as pandas, scikit-learn, and NLTK to perform text sentiment analysis.

4. Preprocessing Text Data: To prepare the text data for sentiment analysis, we applied various pre-processing steps, including tokenization, removing stop words, and converting text to numerical features using techniques like TF-IDF (Term Frequency-Inverse Document Frequency).

5. Sentiment Analysis using Python for Text Data: We used Multinomial Naive Bayes classifier to perform sentiment analysis on text data. The model is trained on labeled data, and its accuracy is evaluated on a test dataset. The trained model can also be used to make predictions on new, unseen text data.

6. NLP Sentiment Analysis in Python: Natural Language Processing (NLP) techniques were applied to analyze customer reviews. This included feature extraction, model training, and evaluation of sentiment classification accuracy.

By the end of this project, we were able to classify customer reviews into different sentiment categories, gaining valuable insights into the overall sentiment expressed by Zomato customers regarding various restaurants and their experiences. This information can be used by businesses to improve customer satisfaction and make data-driven decisions.

Overall, this project not only introduced us to sentiment analysis in Python but also equipped us with the tools and knowledge to apply NLP techniques to real-world datasets for sentiment classification. Sentiment analysis is a crucial application of NLP with diverse use cases, including customer feedback analysis, social media monitoring, and more.

Data Description

Zomato Restaurant names and Metadata

Name : Name of Restaurants
Links : URL Links of Restaurants
Cost : Per person estimated Cost of dining
Collection : Tagging of Restaurants w.r.t. Zomato categories
Cuisines : Cuisines served by Restaurants
Timings : Restaurant Timings

Zomato Restaurant reviews

Restaurant : Name of the Restaurant
Reviewer : Name of the Reviewer
Review : Review Text
Rating : Rating Provided by Reviewer
Meta Data : Reviewer Metadata - No. of Reviews and followers
Time: Date and Time of Review
Pictures : No. of pictures posted with review

Problem Statement:

This problem contain two data frames -

- Zomato restaurant data which contains all information about the restaurants that are available on Zomato, Cuisines they serve, per person cost of dining.
- User review collection which contains the ratings, reviews given by users to different restaurants.

Zomato, an Indian restaurant aggregator and food delivery start-up, was founded by Deepinder Goyal and Pankaj Chaddah in 2008. Zomato serves as a platform for providing information, menus, and user reviews of various restaurants. Additionally, it offers food delivery services from partner restaurants in select cities.

India is renowned for its rich and diverse culinary offerings, available in numerous restaurants and hotel resorts, reflecting the country's unity in diversity. The restaurant industry in India is continuously evolving, with more Indians embracing the idea of dining out or having food delivered. The proliferation of restaurants across every state in India has prompted a comprehensive analysis of data to extract valuable insights, intriguing facts, and statistics about the Indian food industry in each city. As a result, this project centres on the analysis of Zomato's restaurant data for each city in India.

This project focuses on two primary aspects: the customers and the company. The objective is to analyse customer sentiments expressed in their reviews and draw meaningful conclusions through visualizations. Additionally, it involves clustering Zomato restaurants into distinct segments. Visualizing the data facilitates instant data analysis. This analysis also addresses specific business cases that can aid customers in discovering the best restaurants in their locality and assist the company in addressing areas where improvement is needed for growth.

Note : I have used Microsoft Excel, before performing data merging and manipulation in Python to meet our specific requirements.

Let’s begin!

Getting to know our data.

Importing libraries

```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.naive_bayes import MultinomialNB
        from sklearn.metrics import accuracy_score
        import matplotlib.pyplot as plt
        %matplotlib inline
```

Dataset First View

```
In [3]: df=pd.read_excel(r"/Users/charukumar/Desktop/Metadata.xlsx")
        display(df)
```

	Name	Rating	Cost	Collections	Cuisines	Timings	Reviews
0	10 Downing Street	3.0	1900	Trending This Week	North Indian, Chinese, Continental	12 Noon to 12 Midnight	I've been to this place about two times and i ...
1	13 Dhaba	4.0	450	Veggie Friendly	North Indian	12:30 PM to 10 PM (Tue-Sun), Mon Closed	I didn't go and eat at the Dhaba.\nI had order...
2	3B's - Buddies, Bar & Barbecue	5.0	1100	Barbecue & Grill, Live Sports Screenings	North Indian, Mediterranean, European	12 Noon to 4 PM, 6:30 PM to 11:30 PM	We go their for a team dinner.The name of the ...
3	AB's - Absolute Barbecues	5.0	1500	Barbecue & Grill, Great Buffets, Corporate Fav...	European, Mediterranean, North Indian	12 Noon to 4:30 PM, 6:30 PM to 11:30 PM	It was excellent experience spiced thank Krish...
4	Absolute Sizzlers	2.0	750	Great Buffets	Continental, American, Chinese	11:30 AM to 1 AM	Service was pathetic. Ordered a sizzler with l...
...	...	...	...	...	...	...	...
99	Urban Asia - Kitchen & Bar	5.0	1100	NaN	Asian, Thai, Chinese, Sushi, Momos	12 Noon to 3 PM, 7 PM to 11 PM	This place is highly recommended. It is workin...
100	Wich Please	NaN	250	NaN	Fast Food	8am to 12:30AM (Mon-Sun)	NaN
101	Yum Yum Tree - The Arabian Food Court	5.0	1200	Food Hygiene Rated Restaurants in Hyderabad	North Indian, Hyderabadi	12 Noon to 12 Midnight	It is at 6th floor of Act Boutique building th...
102	Zega - Sheraton Hyderabad Hotel	5.0	1750	NaN	Asian, Sushi	12Noon to 2AM (Mon-Sun)	My husband and I, visited Zega for their dimsu...
103	Zing's Northeast Kitchen	4.0	550	NaN	North Eastern, Momos	11:30 AM to 4 PM, 7 PM to 11 PM	The food is tooooooooo good. The interior and...

104 rows × 7 columns

Counting dataset's rows and columns

```
In [5]: df.shape
Out[5]: (104, 7)
```

Our data contains 104 features and 7 records

Dataset Information

```
In [6]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 104 entries, 0 to 103
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Name         104 non-null    object
 1   Rating       100 non-null    float64
 2   Cost         104 non-null    int64
 3   Collections  51 non-null     object
 4   Cuisines     104 non-null    object
 5   Timings      103 non-null    object
 6   Reviews      100 non-null    object
dtypes: float64(1), int64(1), object(5)
memory usage: 5.8+ KB
```

Duplicate Values

```
In [7]: #hotel Dataset Duplicate Value Count
print(len(df[df.duplicated()]))
0
```

Our dataset has no duplicate values

Statistical Summary of Data

In [8]:
```
df.describe().transpose()
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
Rating	100.0	3.470000	1.480206	1.0	2.0	4.0	5.0	5.0
Cost	104.0	864.903846	511.369749	150.0	500.0	700.0	1200.0	2800.0

Count and percentage of missing values in dataset

In [9]:
```
count = df.isnull().sum().sort_values(ascending=False)
percentage = ((df.isnull().sum()/len(df)*100)).sort_values(ascending=False
missing_data= pd.concat([count, percentage,], axis=1,
keys=['Count','Percentage'])
print('Count and %age of missing values for columns: ')
missing_data
```

```
Count and %age of missing values for columns:
```

Out[9]:

	Count	Percentage
Collections	53	50.961538
Rating	4	3.846154
Reviews	4	3.846154
Timings	1	0.961538
Name	0	0.000000
Cost	0	0.000000
Cuisines	0	0.000000

Filling missing values (numerical)

In [12]:
```
# Calculate the mean rating (excluding missing values)
mean_rating = df['Rating'].mean()

# Fill missing values in the 'Rating' column with the mean rating
df['Rating'].fillna(mean_rating, inplace=True)

# Print the resulting DataFrame
display(df)
```

	Name	Rating	Cost	Collections	Cuisines	Timings	Reviews
0	10 Downing Street	3.00	1900	Trending This Week	North Indian, Chinese, Continental	12 Noon to 12 Midnight	I've been to this place about two times and i ...
1	13 Dhaba	4.00	450	Veggie Friendly	North Indian	12:30 PM to 10 PM (Tue-Sun), Mon Closed	I didn't go and eat at the Dhaba.\nI had order...
2	3B's - Buddies, Bar & Barbecue	5.00	1100	Barbecue & Grill, Live Sports Screenings	North Indian, Mediterranean, European	12 Noon to 4 PM, 6:30 PM to 11:30 PM	We go their for a team dinner.The name of the ...
3	AB's - Absolute Barbecues	5.00	1500	Barbecue & Grill, Great Buffets, Corporate Fav...	European, Mediterranean, North Indian	12 Noon to 4:30 PM, 6:30 PM to 11:30 PM	It was excellent experience spiced thank Krish...
4	Absolute Sizzlers	2.00	750	Great Buffets	Continental, American, Chinese	11:30 AM to 1 AM	Service was pathetic. Ordered a sizzler with l...

Filling missing values (Text)

```
In [16]: df.drop(['Collections', 'Timings'], axis = 1, inplace = True)
```

```
In [17]: display(df)
```

	Name	Rating	Cost	Cuisines	Reviews
0	10 Downing Street	3.00	1900	North Indian, Chinese, Continental	I've been to this place about two times and i ...
1	13 Dhaba	4.00	450	North Indian	I didn't go and eat at the Dhaba.\nI had order...
2	3B's - Buddies, Bar & Barbecue	5.00	1100	North Indian, Mediterranean, European	We go their for a team dinner.The name of the ...
3	AB's - Absolute Barbecues	5.00	1500	European, Mediterranean, North Indian	It was excellent experience spiced thank Krish...
4	Absolute Sizzlers	2.00	750	Continental, American, Chinese	Service was pathetic. Ordered a sizzler with I...
...	...	...	...	...	...
99	Urban Asia - Kitchen & Bar	5.00	1100	Asian, Thai, Chinese, Sushi, Momos	This place is highly recommended. It is workin...
100	Wich Please	3.47	250	Fast Food	No reviews available
101	Yum Yum Tree - The Arabian Food Court	5.00	1200	North Indian, Hyderabadi	It is at 6th floor of Act Boutique building th...
102	Zega - Sheraton Hyderabad Hotel	5.00	1750	Asian, Sushi	My husband and I, visited Zega for their dimsu...
103	Zing's Northeast Kitchen	4.00	550	North Eastern, Momos	The food is tooooooooooo good. The interior and...

104 rows × 5 columns

Dropping columns that are not needed

```
In [13]: # Fill missing values in the 'Reviews' column with a default text (e.g., 'No reviews available')
df['Reviews'].fillna('No reviews available', inplace=True)

# Print the resulting DataFrame
display(df)
```

	Name	Rating	Cost	Collections	Cuisines	Timings	Reviews
0	10 Downing Street	3.00	1900	Trending This Week	North Indian, Chinese, Continental	12 Noon to 12 Midnight	I've been to this place about two times and i ...
1	13 Dhaba	4.00	450	Veggie Friendly	North Indian	12:30 PM to 10 PM (Tue-Sun), Mon Closed	I didn't go and eat at the Dhaba.\nI had order...
2	3B's - Buddies, Bar & Barbecue	5.00	1100	Barbecue & Grill, Live Sports Screenings	North Indian, Mediterranean, European	12 Noon to 4 PM, 6:30 PM to 11:30 PM	We go their for a team dinner.The name of the ...
3	AB's - Absolute Barbecues	5.00	1500	Barbecue & Grill, Great Buffets, Corporate Fav...	European, Mediterranean, North Indian	12 Noon to 4:30 PM, 6:30 PM to 11:30 PM	It was excellent experience spiced thank Krish...
4	Absolute Sizzlers	2.00	750	Great Buffets	Continental, American, Chinese	11:30 AM to 1 AM	Service was pathetic. Ordered a sizzler with I...
...	...	...	...	...	...	...	...
99	Urban Asia - Kitchen & Bar	5.00	1100	NaN	Asian, Thai, Chinese, Sushi, Momos	12 Noon to 3 PM, 7 PM to 11 PM	This place is highly recommended. It is workin...
100	Wich Please	3.47	250	NaN	Fast Food	8am to 12:30AM (Mon-Sun)	No reviews available
101	Yum Yum Tree - The Arabian Food Court	5.00	1200	Food Hygiene Rated Restaurants in Hyderabad	North Indian, Hyderabadi	12 Noon to 12 Midnight	It is at 6th floor of Act Boutique building th...
102	Zega - Sheraton Hyderabad Hotel	5.00	1750	NaN	Asian, Sushi	12Noon to 2AM (Mon-Sun)	My husband and I, visited Zega for their dimsu...
103	Zing's Northeast Kitchen	4.00	550	NaN	North Eastern, Momos	11:30 AM to 4 PM, 7 PM to 11 PM	The food is tooooooooooo good. The interior and...

Calculating average rating of each restaurant

In [19]:
```
# Group the data by "Restaurant Name" and calculate the mean rating for each
average_ratings = df.groupby('Name')['Rating'].mean().reset_index()

# Display the average ratings for each restaurant
display(average_ratings)
```

	Name	Rating
0	10 Downing Street	3.00
1	13 Dhaba	4.00
2	3B's - Buddies, Bar & Barbecue	5.00
3	AB's - Absolute Barbecues	5.00
4	Absolute Sizzlers	2.00
...	...	...
99	Wich Please	3.47
100	Yum Yum Tree - The Arabian Food Court	5.00
101	Zega - Sheraton Hyderabad Hotel	5.00
102	Zing's Northeast Kitchen	4.00
103	eat.fit	3.00

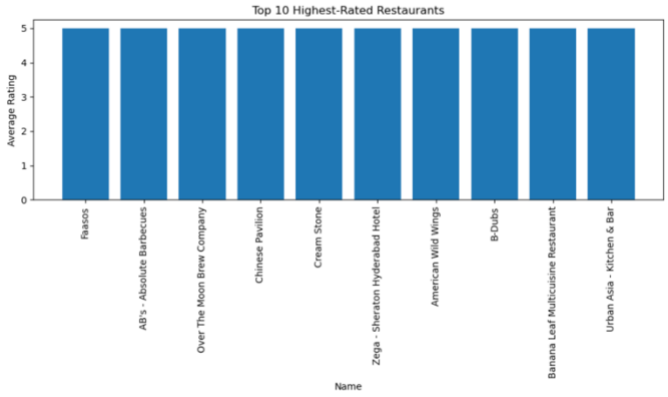
Visualising

In [23]:
```
# Sort the DataFrame by rating in descending order to visualize the highest-rated restaurants first
average_ratings = average_ratings.sort_values(by='Rating', ascending=False)

# Select the top 10 highest-rated restaurants
top_10_ratings = average_ratings.head(10)

# Create a bar chart to visualize the top 10 highest-rated restaurants
plt.figure(figsize=(10, 6))
plt.bar(top_10_ratings['Name'], top_10_ratings['Rating'])
plt.xlabel('Name')
plt.ylabel('Average Rating')
plt.title('Top 10 Highest-Rated Restaurants')
plt.xticks(rotation=90)  # Rotate x-axis labels for readability

# Show the chart
plt.tight_layout()
plt.show()
```



Finding top 10 most expensive restaurants

In [25]:
```
# Sort the DataFrame by cost in descending order to find the most expensive restaurants
top_10_expensive = df.sort_values(by='Cost', ascending=False).head(10)

# Display the top 10 most expensive restaurants
display(top_10_expensive)
```

	Name	Rating	Cost	Cuisines	Reviews
22	Collage - Hyatt Hyderabad Gachibowli	3.00	2800	Continental, Italian, North Indian, Chinese, A...	Good ambiance with wide range food options , p...
35	Feast - Sheraton Hyderabad Hotel	1.00	2500	Modern Indian, Asian, Continental, Italian	With the kind of price they have for the buffe...
48	Jonathan's Kitchen - Holiday Inn Express & Suites	1.00	1900	North Indian, Japanese, Italian, Salad, Sushi	Very bad taste including Vegetarian and Non-Ve...
0	10 Downing Street	3.00	1900	North Indian, Chinese, Continental	I've been to this place about two times and i ...
19	Cascade - Radisson Hyderabad Hitec City	5.00	1800	North Indian, Italian, Continental, Asian	This is a nice place for family as well as off...
102	Zega - Sheraton Hyderabad Hotel	5.00	1750	Asian, Sushi	My husband and I, visited Zega for their dimsu...
60	Mazzo - Marriott Executive Apartments	2.00	1700	Italian, North Indian, South Indian, Asian	I am a big soup fan and both hot and sour and ...
74	Republic Of Noodles - Lemon Tree Hotel	3.47	1700	Thai, Asian, Chinese, Malaysian	No reviews available
13	Barbeque Nation	5.00	1600	Mediterranean, North Indian, Kebab, BBQ	#Foodengineeringg\n#RamadanSpecial\n#Reviewmod...
8	Arena Eleven	4.00	1600	Continental	Located in midst of SLN terminus this place is...

Finding top 10 cheapest restaurants

In [27]:
```
# Sort the DataFrame by cost in ascending order to find the cheapest restaurants
top_10_cheap = df.sort_values(by='Cost').head(10)

# Display the top 10 cheapest restaurants
display(top_10_cheap)
```

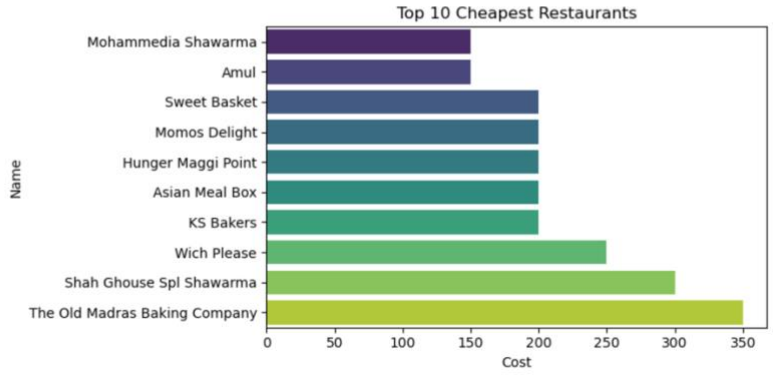
	Name	Rating	Cost	Cuisines	Reviews
61	Mohammedia Shawarma	4.00	150	Street Food, Arabian	Best shawarma served at reasonable price and t...
7	Amul	5.00	150	Ice Cream, Desserts	The place I prefer most for good taste and enj...
83	Sweet Basket	3.47	200	Bakery, Mithai	No reviews available
62	Momos Delight	1.00	200	Momos	Ordered because I wanted to eat momos so much ...
43	Hunger Maggi Point	5.00	200	Fast Food	It is good but i want some spicy if it is spic...
10	Asian Meal Box	4.00	200	Asian	Meal box are value for money. The tossed noodl...
55	KS Bakers	2.00	200	Bakery, Desserts, Fast Food	Just Average. Ordered some 7-8 times. They cou...
100	Wich Please	3.47	250	Fast Food	No reviews available
78	Shah Ghouse Spl Shawarma	4.00	300	Lebanese	The biryani here is very delicious, one should...
93	The Old Madras Baking Company	4.00	350	Bakery	Nestled between the hustle bustle of Gachibowl...

Visualising most expensive and cheapest restaurants

In [34]:
```
# Sort the DataFrame by cost in ascending order (optional)
top_10_cheap = top_10_cheap.sort_values(by='Cost')

# Create a bar plot using Seaborn
plt.figure(figsize=(8, 4))
sns.barplot(x='Cost', y='Name', data=top_10_cheap, palette='viridis')
plt.xlabel('Cost')
plt.ylabel('Name')
plt.title('Top 10 Cheapest Restaurants')
plt.tight_layout()

# Show the chart
plt.show()
```

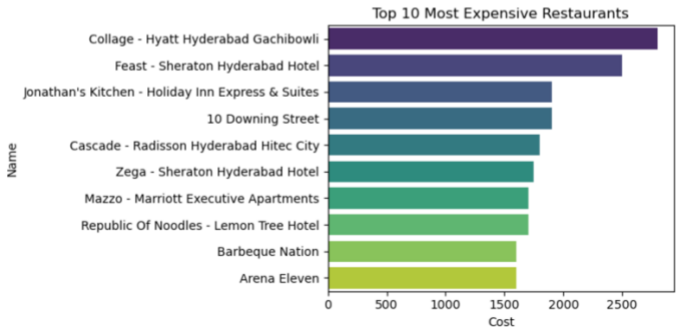


In [33]:
```
import seaborn as sns

# Sort the DataFrame by cost in descending order (optional)
top_10_expensive = top_10_expensive.sort_values(by='Cost', ascending=False)

# Create a bar plot using Seaborn
plt.figure(figsize=(8, 4))
sns.barplot(x='Cost', y='Name', data=top_10_expensive, palette='viridis')
plt.xlabel('Cost')
plt.ylabel('Name')
plt.title('Top 10 Most Expensive Restaurants')
plt.tight_layout()

# Show the chart
plt.show()
```



Creating wordcloud for expensive restaurants

In [35]:
```
#CREATING WORDCLOUD FOR EXPENSIVE RESTAURANT
from wordcloud import WordCloud

plt.figure(figsize=(10,6))
text = " ".join(name for name in df.sort_values('Cost', ascending=False)['Name

# Creating word_cloud with text as an argument in .generate() method
word_cloud = WordCloud(width=2000, height=2000, collocations=False,
                       colormap='gist_earth', background_color='white').generat

# Display the generated Word Cloud
plt.imshow(word_cloud, interpolation='bilinear')
plt.axis("off")
```

Out[35]:
```
(-0.5, 1999.5, 1999.5, -0.5)
```

Finding popular cuisines

In [38]:
```
#Most popular cuisines
cuisine_list=[]
cuisines=df.Cuisines.str.split(',')

#Get all the cuisines in a list
for i in cuisines:
  for j in i:
    cuisine_list.append(j)

# converting it to dataframe
cuisine_series=pd.Series(cuisine_list)
cuisine_df=pd.DataFrame(cuisine_series,columns=['Cuisines'])
cuisine_df[cuisine_df['Cuisines']==' North Indian']='North Indian'
```

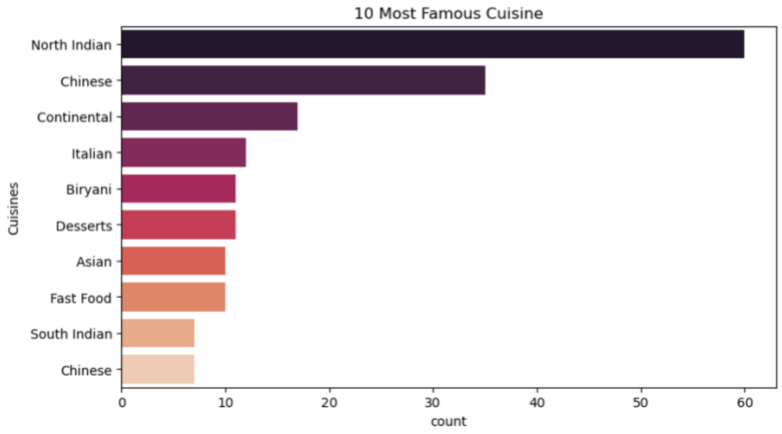
In [39]:
```
#Let's Find the count of each cuisine
cuisine_=pd.DataFrame(cuisine_df.groupby(by='Cuisines',as_index=False).value_counts())
cuisine_
```

Out[39]:

	Cuisines	count
0	American	2
1	Andhra	3
2	Arabian	1
3	Asian	10
4	BBQ	1
...	...	...
64	North Indian	60
65	Seafood	1
66	South Indian	2
67	Street Food	2
68	Thai	1

Visualising top 10 popular cuisines

```
In [40]: plt.rcParams['figure.figsize']= (9,5)
sns.barplot(x='count', y='Cuisines', data=cuisine_.sort_values(ascending=False, by='count')[:10],palette=
plt.title('10 Most Famous Cuisine')
plt.show()
```



SENTIMENT ANALYSIS

Multinomial Naive Bayes classifier to perform sentiment analysis on text data

```
In [41]: # Create a 'Sentiment' column based on the 'Rating' column
df['Sentiment'] = df['Rating'].apply(lambda x: 'good' if x >= 4 else 'bad')

# Split data into training and testing sets
X = df['Reviews']
y = df['Sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a TF-IDF vectorizer to convert text data into numerical features
tfidf_vectorizer = TfidfVectorizer(max_features=5000)  # You can adjust the number of fea

# Fit and transform the vectorizer on the training data
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Train a Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(X_train_tfidf, y_train)

# Predict sentiments on the test data
y_pred = classifier.predict(X_test_tfidf)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Now, you can use the classifier to predict sentiment for new reviews
new_reviews = ["This restaurant was amazing!", "The food was terrible."]
new_reviews_tfidf = tfidf_vectorizer.transform(new_reviews)
sentiments = classifier.predict(new_reviews_tfidf)
print("Predicted Sentiments:", sentiments)
```

```
Accuracy: 0.5714285714285714
Predicted Sentiments: ['good' 'good']
```

View of the updated table

```
In [42]: display(df)
```

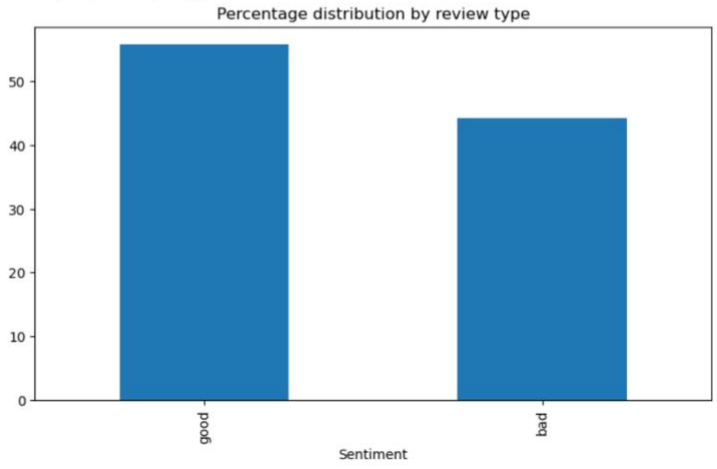
	Name	Rating	Cost	Cuisines	Reviews	Sentiment
0	10 Downing Street	3.00	1900	North Indian, Chinese, Continental	I've been to this place about two times and i ...	bad
1	13 Dhaba	4.00	450	North Indian	I didn't go and eat at the Dhaba.\nI had order...	good
2	3B's - Buddies, Bar & Barbecue	5.00	1100	North Indian, Mediterranean, European	We go their for a team dinner.The name of the ...	good
3	AB's - Absolute Barbecues	5.00	1500	European, Mediterranean, North Indian	It was excellent experience spiced thank Krish...	good
4	Absolute Sizzlers	2.00	750	Continental, American, Chinese	Service was pathetic. Ordered a sizzler with l...	bad
...	...	...	...	...	...	...
99	Urban Asia - Kitchen & Bar	5.00	1100	Asian, Thai, Chinese, Sushi, Momos	This place is highly recommended. It is workin...	good
100	Wich Please	3.47	250	Fast Food	No reviews available	bad
101	Yum Yum Tree - The Arabian Food Court	5.00	1200	North Indian, Hyderabadi	It is at 6th floor of Act Boutique building th...	good
102	Zega - Sheraton Hyderabad Hotel	5.00	1750	Asian, Sushi	My husband and I, visited Zega for their dimsu...	good
103	Zing's Northeast Kitchen	4.00	550	North Eastern, Momos	The food is tooooooooo good. The interior and...	good

Percentage visualisation of review type

```
In [43]: print('%age for default\n')
print(round(df.Sentiment.value_counts(normalize=True)*100,2))
round(df.Sentiment.value_counts(normalize=True)*100,2).plot(kind='bar')
plt.title('Percentage distribution by review type')
plt.show()
```

```
%age for default

Sentiment
good    55.77
bad     44.23
Name: proportion, dtype: float64
```



Data Cleaning

Ist Phase

In [44]:
```
#Apply first level cleaning
import re
import string

#This function will convert to lower case, removes square brackets, numbers and punctuations from reviews
def text_clean_1(text):
    text=text.lower()
    text=re.sub('\[.*?\]', '', text)
    text=re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text=re.sub('\w*\d\w*', '', text)
    return text
cleaned1=lambda x: text_clean_1(x)
```

In [45]:
```
#updated text
df['cleaned_reviews']=pd.DataFrame(df.Reviews.apply(cleaned1))
df.head(10)
```

Out[45]:

	Name	Rating	Cost	Cuisines	Reviews	Sentiment	cleaned_reviews
0	10 Downing Street	3.0	1900	North Indian, Chinese, Continental	I've been to this place about two times and i ...	bad	ive been to this place about two times and i r...
1	13 Dhaba	4.0	450	North Indian	I didn't go and eat at the Dhaba.\nI had order...	good	i didnt go and eat at the dhaba\ni had ordered...
2	3B's - Buddies, Bar & Barbecue	5.0	1100	North Indian, Mediterranean, European	We go their for a team dinner.The name of the ...	good	we go their for a team dinnerthe name of the g...
3	AB's - Absolute Barbecues	5.0	1500	European, Mediterranean, North Indian	It was excellent experience spiced thank Krish...	good	it was excellent experience spiced thank krish...
4	Absolute Sizzlers	2.0	750	Continental, American, Chinese	Service was pathetic. Ordered a sizzler with I...	bad	service was pathetic ordered a sizzler with la...
5	Al Saba Restaurant	3.0	750	North Indian, Chinese, Seafood, Biryani, Hyder...	Visited this place at night. Had chicken Birya...	bad	visited this place at night had chicken biryan...
6	American Wild Wings	5.0	600	American, Fast Food, Salad, Burger	found them on Zomato website, a very interesti...	good	found them on zomato website a very interestin...
7	Amul	5.0	150	Ice Cream, Desserts	The place I prefer most for good taste and enj...	good	the place i prefer most for good taste and enj...
8	Arena Eleven	4.0	1600	Continental	Located in midst of SLN terminus this place is...	good	located in midst of sln terminus this place is...
9	Aromas@11SIX	3.0	750	North Indian, Chinese, Mughlai, Biryani	This place is located in khajaguda. It used to...	bad	this place is located in khajaguda it used to ...

2nd Phase

In [46]:
```
#Second cleaning

def text_clean_2(text):
    text=re.sub('\n','',text)
    text=re.sub('['']', '', text)
    return text
cleaned2=lambda x: text_clean_2(x)
```

In [47]:
```
df['cleaned_reviews']=pd.DataFrame(df.Reviews.apply(cleaned2))
df.head(10)
```

Out[47]:

	Name	Rating	Cost	Cuisines	Reviews	Sentiment	cleaned_reviews
0	10 Downing Street	3.0	1900	North Indian, Chinese, Continental	I've been to this place about two times and i ...	bad	I've been to this place about two times and i ...
1	13 Dhaba	4.0	450	North Indian	I didn't go and eat at the Dhaba.\nI had order...	good	I didn't go and eat at the Dhabal had ordered ...
2	3B's - Buddies, Bar & Barbecue	5.0	1100	North Indian, Mediterranean, European	We go their for a team dinner.The name of the ...	good	We go their for a team dinnerThe name of the g...
3	AB's - Absolute Barbecues	5.0	1500	European, Mediterranean, North Indian	It was excellent experience spiced thank Krish...	good	It was excellent experience spiced thank Krish...
4	Absolute Sizzlers	2.0	750	Continental, American, Chinese	Service was pathetic. Ordered a sizzler with I...	bad	Service was pathetic Ordered a sizzler with la...
5	Al Saba Restaurant	3.0	750	North Indian, Chinese, Seafood, Biryani, Hyder...	Visited this place at night. Had chicken Birya...	bad	Visited this place at night Had chicken Biryan...
6	American Wild Wings	5.0	600	American, Fast Food, Salad, Burger	found them on Zomato website, a very interesti...	good	found them on Zomato website, a very interesti...
7	Amul	5.0	150	Ice Cream, Desserts	The place I prefer most for good taste and enj...	good	The place I prefer most for good taste and enj...
8	Arena Eleven	4.0	1600	Continental	Located in midst of SLN terminus this place is...	good	Located in midst of SLN terminus this place is...
9	Aromas@11SIX	3.0	750	North Indian, Chinese, Mughlai, Biryani	This place is located in khajaguda. It used to...	bad	This place is located in khajaguda It used to ...

Training and testing dataset

In [48]:
```
from sklearn.model_selection import train_test_split
IR=df.cleaned_reviews
DR=df.Sentiment

IV_train, IV_test, DV_train, DV_test = train_test_split(IR, DR, test
print('IV_train :', len(IV_train))
print('IV_test :', len(IV_test))
print('DV_train :', len(DV_train))
print('DV_test :', len(DV_test))
```

```
IV_train : 93
IV_test : 11
DV_train : 93
DV_test : 11
```

- 93 data points for training the machine learning model (IV_train and DV_train).
- 11 data points for testing the model's performance (IV_test and DV_test).

Text Classification

In [49]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression

tvec = TfidfVectorizer()
clf2 = LogisticRegression(solver = "lbfgs")

from sklearn.pipeline import Pipeline
```

In [50]:
```
model = Pipeline([('vectorizer', tvec), ['classifier', clf2]])
model.fit(IV_train, DV_train)
from sklearn.metrics import confusion_matrix
predictions = model.predict(IV_test)
confusion_matrix(predictions, DV_test)
```

Out[50]:
```
array([[0, 0],
       [4, 7]])
```

- The value in the top-left cell (0,0) represents the number of true negatives (TN). It's 0, which means the model correctly predicted negatives 0 times.
- The value in the bottom-right cell (1,1) represents the number of true positives (TP). It's 7, which means the model correctly predicted positives 7 times.
- The value in the top-right cell (0,1) represents the number of false positives (FP). It's 0, which means the model incorrectly predicted positives 0 times.
- The value in the bottom-left cell (1,0) represents the number of false negatives (FN). It's 4, which means the model incorrectly predicted negatives 4 times.

Experimenting with the trained dataset

In [51]:
```
example = ["I'm not happy"]
result = model.predict(example)
print(result)
```

```
['bad']
```

In [52]:
```
example = ["I'm satisfied "]
result = model.predict(example)
print(result)
```

```
['good']
```

In [53]:
```
example = ["The food was not tasty"]
result = model.predict(example)
print(result)
```

```
['bad']
```

In [54]:
```
example = ["Place was beautiful"]
result = model.predict(example)
print(result)
```

```
['good']
```

The code takes the text "I'm not happy," uses a machine learning model to predict its sentiment, and then prints the predicted sentiment label, which is "bad" in this case.