	#Description # with the rapid growth of online platforms for sharing opinions and reviews, restarunts often rely # on the customer feedback to imporve their services and attract a new customers. # Analyzing the sentiment of these reviews can provide valuable insights into customer satisfaction.
In [7]: In [9]:	#Problem statement #Develop a sentiment analysis model to classify restarunt reviews as positive or negative pip install pandas
2-1	Defaulting to user installation because normal site-packages is not writeable Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (1.4.2) Requirement already satisfied: numpy>=1.18.5 in c:\programdata\anaconda3\lib\site-packages (from pandas) (1.21.5) Requirement already satisfied: python-dateutil>=2.8.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2.8.2)
In [10]:	Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2021.3) Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0) Note: you may need to restart the kernel to use updated packages. import pandas as pd
In [6]:	<pre>data = pd.read_csv('Reviews.csv') data</pre>
Out[6]:	Review Liked Wow Loved this place. 1 Crust is not good. 0 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
	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
	 The whole experience was underwhelming, and I 0 Then, as if I hadn't wasted enough of my life 0 rows × 2 columns
In []: In []:	
In [11]: Out[11]:	data.head() # Top 5 rows of the data set Review Liked Wow Loved this place. 1
	Crust is not good. Not tasty and the texture was just nasty. Stopped by during the late May bank holiday of 1
In [12]:	4 The selection on the menu was great and so wer 1 data.tail() # Last 5 rows of the data set
Out[12]:	ReviewLiked995I think food should have flavor and texture an0996Appetite instantly gone.0
	 Overall I was not impressed and would not go b The whole experience was underwhelming, and I Then, as if I hadn't wasted enough of my life
In [13]:	<pre>data.info() # information of the dat set like , data type , memory usage <class 'pandas.core.frame.dataframe'=""> RangeIndex: 1000 entries, 0 to 999 Data columns (total 2 columns):</class></pre>
	# Column Non-Null Count Dtype
In [14]: Out[14]:	memory usage: 15.8+ KB data.describe() # stastical information of the data set Liked
	count 1000.00000 mean 0.50000 std 0.50025
	min 0.00000 25% 0.00000 50% 0.50000 75% 1.00000
In [15]:	max 1.00000 #checking the null values of the data set
Out[15]:	<pre>data.isnull().sum() Review 0 Liked 0 dtype: int64</pre>
In [16]: Out[16]:	<pre>data.duplicated() 0 False 1 False 2 False 3 False</pre>
	4 False 995 False 996 False 997 False
In [17]:	998 False 999 False Length: 1000, dtype: bool #checking the value counts
L I	<pre>value_counts = data['Liked'].value_counts() print(value_counts) 1 500 0 500 Name: Liked, dtype: int64</pre>
In [18]:	pip install matplotlib Defaulting to user installation because normal site-packages is not writeableNote: you may need to restart the kernel to use updated packages. Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-packages (3.5.1)
	Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (1.3.2) Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (21.3) Requirement already satisfied: pyparsing>=2.2.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (3.0.4) Requirement already satisfied: numpy>=1.17 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (1.21.5) Requirement already satisfied: python-dateutil>=2.7 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (2.8.2) Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
	Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (4.25.0) Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (0.11.0) Requirement already satisfied: pillow>=6.2.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (9.0.1) Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
In [19]: In [20]:	<pre>import matplotlib.pyplot as plt import seaborn as sns value_counts.plot(kind = 'bar' , color = ['blue', 'green']) plt.title("Sentiment value counts")</pre>
	<pre>plt.xlabel('Liked') plt.ylabel('Count') plt.xticks(ticks=[0,1] , labels=['Postive','Negative'],rotation=0) plt.show()</pre>
	Sentiment value counts 500 - 400 -
	300 - 200 -
	100 - Postive Negative
In []:	Postive Negative Liked
In []: In [23]: In [20]:	<pre>from collections import Counter target_words = ['food', 'place', 'restaurant']</pre>
رحا]:	<pre>all_words = " ".join(data['Review']).lower().split() word_counts = Counter(all_words) target_word_counts = {word:word_counts[word] for word in target_words} plt.figure(figsize=(8,6)) plt.bar(target_word_counts.keys(),target_word_counts.values() , color = ['blue','green','orange'])</pre>
	<pre>plt.xlabel('words') plt.ylabel('Frequency') plt.title('Frequency of specific words in Reviews') plt.show()</pre>
	Frequency of specific words in Reviews
	80 -
	60 - South Programme
	Ledneueck 40 -
	20 -
	food place restaurant
In [24]:	
In [25]: In [26]:	<pre>#convert a data set into lower case lowercased_text = data['Review'].str.lower() print(lowercased_text)</pre>
	wow loved this place. crust is not good. not tasty and the texture was just nasty. stopped by during the late may bank holiday of the selection on the menu was great and so wer
	i think food should have flavor and texture an 996 appetite instantly gone. 997 overall i was not impressed and would not go b 998 the whole experience was underwhelming, and i 999 then, as if i hadn't wasted enough of my life
In [27]:	Name: Review, Length: 1000, dtype: object #tokinization from nltk.tokenize import word_tokenize
In []: In []:	<pre>print(data['Tokens']) data_info()</pre>
In []: In []: In []:	<pre>data.info() data['Review'].value_counts() import string</pre>
In []: In []:	<pre>import string data['Review'] = data['Review'].str.replace(f"[{string.punctuation}]"," ",regex = True) print(data['Review'])</pre>
In []:	<pre>data['Review'].value_counts() #Removing the stop words like this, is , are ,was from nltk.corpus import stopwords</pre>
Tn [].	Trom Hier. con pus import scopwords
In []:	<pre>stop_words = set(stopwords.words('english')) data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words])</pre>
	<pre>data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) #stemming</pre>
In []:	<pre>data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) #stemming #stemming is the process of reducing the a word into root or base word form by removig suffix #example : driving stemmed is drive #Stemming from nltk.stem import PorterStemmer</pre>
In []: In []: In []: In []:	<pre>data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) #stemming #stemming is the process of reducing the a word into root or base word form by removig suffix #example : driving stemmed is drive #Stemming from nltk.stem import PorterStemmer from nltk.tokenize import word_tokenize stemmer = PorterStemmer()</pre>
In []: In []: In []:	<pre>data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) #stemming #stemming is the process of reducing the a word into root or base word form by removig suffix #example : driving stemmed is drive #Stemming from nltk.stem import PorterStemmer from nltk.tokenize import word_tokenize stemmer = PorterStemmer() data['stemmed'] = data['Review'].apply(lambda x: ' '.join([stemmer.stem(word) for word in word_tokenize(x)])) #Lemmatization</pre>
In []:	<pre>data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) #stemming #stemming is the process of reducing the a word into root or base word form by removig suffix #example : driving stemmed is drive #Stemming from nltk.stem import PorterStemmer from nltk.tokenize import word_tokenize stemmer = PorterStemmer() data['stemmed'] = data['Review'].apply(lambda x: ' '.join([stemmer.stem(word) for word in word_tokenize(x)]))</pre>
In []: In []: In []: In []: In []: In []: In [31]: In [44]:	data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) #stemming #stemming is the process of reducing the a word into root or base word form by removig suffix #example : driving stemmed is drive #stemming from nltk.stem import PorterStemmer from nltk.stenkenize import word_tokenize stemmer = PorterStemmer() data['stemmed'] = data['Review'].apply(lambda x: ' '.join([stemmer.stem(word) for word in word_tokenize(x)])) #Lemmatization #Lemmatization is the process transforming a word into its base or dictionary form #example is better is lemmtized to good from nltk.stem import WordNetLemmatizer from nltk.stem import WordNetLemmatizer from nltk.corpus import word_tokenize from nltk.corpus import wordnet
In []: In []: In []: In []: In []: In []: In [44]: In [46]:	data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) #stemming
In []: In []: In []: In []: In []: In []: In [44]: In [45]:	data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) estemming estemming as the process of reducing the a word into root or base word form by removig suffix example: driving stemmed is drive #stemming from nitk.stem import PorterStemmer from nitk.stem import word_tokenize stemmer = PorterStemmer() data['Stemmed'] = data['Review'].apply(lambda x: ' '.join([stemmer.stem(word) for word in word_tokenize(x)])) #lemmatization #lemmatization is the process transforming a word into its base or dictionary form #example is better is lemmized to good from nitk.stem import wordwetLemmatizer from nitk.cokenize import wordwetLemmatizer from nitk.cokenize import wordwetLemmatizer lemmatizer = WordWetLemmatizer() data['Lemmatized'] = data['Review'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word , pos = wordnet.VER8) for word in word_tokenize(x)])) print(data['Lemmatized']) Not tasty and the texture be just nasty stopped by during the late May bash holiday of
In []: In []: In []: In []: In []: In []: In [44]: In [46]:	data('Tokens') = data('Review').apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data('Tokens')) #stomming
In []: In []: In []: In []: In []: In []: In [44]: In [46]:	data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) #stemming #stemming is the process of reducing the a word into root or base word form by removig suffix #stemming is the process of reducing the a word into root or base word form by removig suffix #stemming is the process of reducing the a word into root or base word form by removig suffix #stemming is the process of reducing the a word into root or base word form by removig suffix #stemming is the process of reducing the sufficient of the sufficient of the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form #stemming is the process transforming a word into its base or dictionary form word. #stemming is the process transforming a word into its base or dictionary form or dictionary form is the process transforming a word into its base or dictionary form or dictionary form or dictionary form or dicti
In []: In []: In []: In []: In []: In [31]: In [44]: In [45]: In [46]: In [47]:	data['Tokens'] = data['Review'].apply(lambda x: [word for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) ### Stemming ### Stemming St. The process of reducing the a word into root or base word form by removing suffix ### Stemming ###
In []: In [44]: In [45]: In [46]: In [47]:	data['tokens'] = data['mexims'].apply[lambda x: [sord for word in word_tokenize(x) if word not in stop_words]) print(data['Tokens']) streaming streaming streaming streaming striving steemed is drive streaming striving steemed is drive from allk.tokenize upport word_tokenize streaming from allk.tokenize upport word_tokenize streaming from allk.tokenize upport word_tokenize streaming data['mexims'] apply[lambda x: ''.join[[steemer.stem[word]]]) streaming data['mexims'] apply[lambda x: ''.join[steemer.stem[word]]]) streaming data['mexims'] apply[lambda x: ''.join[steemer.stem[word]]]) streaming data['mexims'] apply[lambda x: ''.join[steemer.stem[word]]]) streaming data['mexims'] apply[lambda x: ''.join[lementizer]]] from allk.steel import wordness apply word apply[lambda x: ''.join[lementizer]]]] streaming data['mexims'] apply word apply[lambda x: ''.join[lementizer]]]] print(data['mexims'] data['mexims'] apply[lambda x: ''.join[lementizer]]]] streaming data['mexims'] apply word apply[lambda x: ''.join[lementizer]]]] streaming data['mexims'] apply word apply word
In []: In [44]: In [45]: In [46]: In [47]:	data['tokens'] = data['sevisw'].apply(lambda x: [owrd for word in word_tokenize(x) if word not in step_words) print(data['tokens']) ###################################
In []: In [44]: In [45]: In [46]: In [47]:	data "Tevens" = Sata "Revise" .apply(lambda x: [send for word in word towentrs(x) if word not in step words) print(data Towers")) ##################################
In []: In [44]: In [45]: In [46]: In [47]: In [49]:	data['lobers'] = data['Revies'].apply[lambda x: [serd for word in word_townize()) If word not in stop_words] #### Process of reducing the a word_stop rout or base word farm by removing suffix ##################################
In []: In [44]: In [45]: In [46]: In [47]: In [49]:	date("takent] = date("takent] - speriyllandia v: [aren for were in were_namentane(c) if word not in eng_words) provided Tokent() provided Tokent
In []: In [44]: In [45]: In [46]: In [47]: In [49]:	### STATE OF THE TOTAL PROPERTY OF THE TOTAL PROPERTY OF THE STATE OF THE TOTAL PROPERTY OF THE STATE OF THE
In []: In []: In []: In []: In []: In [31]: In [44]: In [45]: In [46]: In [47]: In [50]: In [51]:	incal Televist a catal Review apply limited as in part for word in word_contention; if we're not in stop_words); print(anti) Televist a catal Review apply limited as in the real or date and fine by revening settle. **Exercising face of review apply limited as in '.'. joint([clearer exercised]); **Exercised as the review a catal '.'. joint([clearer exercised]); **Exercised as the review a catal '.'. joint([clearer exercised]); **Exercised as the process from transport and as and fine it is case or decisionry from securities of the process from transport and as and fine it is case or decisionry from securities of the process from transport and as and fine it is case or decisionry from securities of the process from transport and as and fine it is case or decisionry from securities of the process from transport and as and fine it is case or decisionry from securities of the process from transport words: ***Exercised as the process from transpor
In []: In []: In []: In []: In []: In [31]: In [44]: In [45]: In [46]: In [47]: In [50]: In [51]:	Bits Titles - dea of Townson carding been a cond one cont on how many for any carding setting and continued to the armount of controls of the armoun
In []: In []: In []: In []: In []: In [31]: In [44]: In [45]: In [46]: In [47]: In [50]: In [51]:	indiginations within Decimal properties and according count for source and according to the properties of the properties
In []: In []: In []: In []: In []: In []: In [44]: In [45]: In [46]: In [47]: In [50]: In [51]:	ACCEPTION OF THE PROPERTY OF T
In []: In []: In []: In []: In []: In [44]: In [44]: In [47]: In [50]: In [51]:	Section of the control of the contro
In []: In [44]: In [45]: In [46]: In [47]: In [53]: In [53]: In [54]: In [54]:	inter(interes) is deterd benefit in apply (landous in (both fire acts the world observation) if acts and interpretation (landous interpretation) in acts and interpretation (l
In []: In [44]: In [45]: In [46]: In [47]: In [51]: In [54]: In [54]: In [56]: In [56]: In [56]:	State Company Compan
In []: In [44]: In [46]: In [47]: In [55]: In [56]: In [57]: In [57]:	The Control of Control of Special Control of Specia
In []: In [44]: In [45]: In [46]: In [47]: In [51]: In [54]: In [54]: In [56]: In [56]: In [56]:	including the control of the control
In []: In []: In []: In []: In []: In [31]: In [44]: In [46]: In [47]: In [56]: In [56]: In [57]: In [57]: In [58]: In [59]:	The content of the
In []: In []: In []: In []: In []: In []: In [44]: In [45]: In [46]: In [56]: In [57]: In [57]: In [57]:	Transfer
In []: In [44]: In [45]: In [46]: In [56]: In [56]: In [57]: In [57]: In [60]: In [60]: In [60]: In [61]:	### Comment of the co
In []: In [44]: In [45]: In [46]: In [56]: In [56]: In [57]: In [57]: In [60]: In [60]: In [60]: In [61]:	The content of the
In []: In [44]: In [45]: In [56]: In [56]: In [57]: In [60]: In [60]: In [60]: In [62]: In [62]:	Company Comp
In []: In [44]: In [45]: In [46]: In [56]: In [57]: In [58]: In [58]: In [60]: In [60]: In [61]: In [62]:	Section of the control of the contro
In []: In [44]: In [45]: In [56]: In [56]: In [57]: In [60]: In [60]: In [60]: In [62]: In [62]:	The stand of the
In []: In [44]: In [45]: In [56]: In [56]: In [57]: In [60]: In [60]: In [60]: In [62]: In [62]:	Section of the content of the conten
In []: In [44]: In [46]: In [48]: In [56]: In [56]: In [56]: In [66]: In [66]: In [66]: In [68]: In	Company Comp
In []: In [44]: In [46]: In [48]: In [56]: In [56]: In [56]: In [66]: In [66]: In [66]: In [68]: In	The content of the
In [6]: In []: In []: In []: In []: In [44]: In [44]: In [46]: In [56]: In [56]: In [57]: In [66]: In [66]: In [66]: In [67]: In [68]: In [71]:	March Marc
In [5]: In []: In []: In []: In []: In [31]: In [44]: In [45]: In [46]: In [56]: In [57]: In [56]: In [66]: In [66]: In [66]: In [66]: In [67]: In [68]: In [68]: In [70]: In [75]: In [75]:	The content of the co
In []: In [44]: In [46]: In [47]: In [56]: In [56]: In [56]: In [66]: In [70]: In	Company Comp
In [3: In []: In []: In []: In []: In [31]: In [44]: In [46]: In [46]: In [56]: In [56]: In [56]: In [57]: In [51]: In [66]: In [66]: In [66]: In [66]: In [67]: In [76]: In [77]: In [Company Comp
In [6]: In []: In []: In []: In []: In [31]: In [44]: In [46]: In [47]: In [53]: In [53]: In [53]: In [54]: In [66]: In [66]: In [66]: In [67]: In [68]: In [68]: In [68]: In [68]: In [70]: In	The content of the
In []: In []: In []: In []: In []: In []: In [4]: In [45]: In [46]: In [47]: In [56]: In [56]: In [56]: In [56]: In [66]: In [66]: In [66]: In [66]: In [70]: In [70	March Marc
In [6]: In []: In []: In []: In []: In [31]: In [44]: In [46]: In [47]: In [53]: In [53]: In [53]: In [54]: In [66]: In [66]: In [66]: In [67]: In [68]: In [68]: In [68]: In [68]: In [70]: In	
In [6]: In []: In []: In []: In []: In [4]: In [44]: In [45]: In [46]: In [56]: In [56]: In [56]: In [56]: In [66]: In [66]: In [66]: In [66]: In [76]: In [
In []: In []: In []: In []: In []: In []: In [4]: In [45]: In [46]: In [47]: In [56]: In [56]: In [56]: In [56]: In [66]: In [66]: In [66]: In [66]: In [70]: In [70	
In [6]: In []: In []: In []: In []: In [4]: In [44]: In [45]: In [46]: In [56]: In [56]: In [56]: In [56]: In [66]: In [66]: In [66]: In [66]: In [76]: In [The state of the s
In [6]: In []: In []: In []: In []: In []: In [4]: In [4]: In [46]: In [47]: In [56]: In [56]: In [56]: In [56]: In [66]: In [66]: In [66]: In [76]: In [76]	