A Text Mining Journey through Hotel and Restaurant Reviews in Thailand.

Dataset:

The dataset contains 53,644 rows and 5 columns of customer reviews of hotels and restaurants in 25 different locations in the province of Phuket, Thailand. In this report we'll select reviews of hotels and Restaurants from two major districts in the Phuket province, each representing a customer's experience. We carefully selected 15 hotels/Restaurants located in Choeng thale in Thalang, in the Northern part of Phuket,

as well as another 15 hotels/Restaurants in Cape Panwa which is situated in Mueang Phuket in the southern part of Phuket which is also the capital of the province. The dataset used for this task is the tourist_accommodation_reviews.csv dataset.

Explanation and preparation of dataset (Exploratory Data Analysis)

1. Open Jupyter Notebook, either from the Anaconda Prompt or the Anaconda Navigator. Open a new notebook by clicking the New tab button and select Python 3 (ipykernel) to create a new notebook workspace.
2. For this report, I will use the Natural Language Toolkit (NLTK) library, which provides a range of text-processing tools for tasks such as classification, tokenization, stemming, tagging, and sentiment analysis. Run the code below to import the libraries.

```
In [1]: #importing the necessary libraries and Packages

        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import re
        import seaborn as sns
        from wordcloud import WordCloud
        import nltk
        nltk.download(['stopwords',
                       'punkt',
                       'wordnet',
                       'omw-1.4',
                       'vader_lexicon'
                      ])
        %matplotlib inline
```

3. Import data using the read_csv() function in the pandas library to read the dataset into a pandas data frame, and then analyse and visualise some of the results.

```
In [2]: #importing the dataset

        reviews = pd.read_csv('tourist_review.csv')
```

4. To gain a better understanding of the dataset and its columns, execute the following code:

   (a) reviews.head()

This function retrieves the first few rows of the dataset.

```
In [3]: #Exploration of the dataset

        reviews.head()
```

Out[3]:

	ID	Review Date	Location	Hotel/Restaurant name	Review
0	rn579769293	Reviewed 1 week ago	Cape Panwa	The Cove Phuket	I finally made to to The Cove after hearing fr...
1	rn578446147	Reviewed 1 week ago	Cape Panwa	The Cove Phuket	The food and the service were both excellent a...
2	rn578261388	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	Almost confused with the corner bar but what a...
3	rn578201696	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	I know your probably going to reply with all t...
4	rn577322860	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	Super service and great food! Nice views and c...

(b) reviews.tail()

This function retrieves the last few rows of the dataset. To get more rows, simply specify a different number inside the parentheses. Example: "reviews.tail(20)" which would return the last 20 rows.

```
In [4]: reviews.tail()
```

Out[4]:

	ID	Review Date	Location	Hotel/Restaurant name	Review
2988	rn145486095	Reviewed November 15, 2012	Choeng Thale	9' Sea Breeze	this great little place on surin beach serves ...
2989	rn145450467	Reviewed November 15, 2012	Choeng Thale	9' Sea Breeze	If you're looking for a good food, great atmos...
2990	rn145136705	Reviewed November 11, 2012	Choeng Thale	9' Sea Breeze	Meeting different people from all over the wor...
2991	rn145067924	Reviewed November 10, 2012	Choeng Thale	9' Sea Breeze	This has to be the best sports bar restaurant ...
2992	rn144513470	Reviewed November 4, 2012	Choeng Thale	9' Sea Breeze	This is a great little restaurant with loads o...

(c) reviews.describe()

This function is used to return the summary of the data.

```
In [5]: reviews.describe()
```

Out[5]:

	ID	Review Date	Location	Hotel/Restaurant name	Review
count	2993	2993	2993	2993	2993
unique	2963	1129	2	30	2963
top	rn579563747	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	We had dinner here. Just stumbled across it. S...
freq	3	32	1500	100	3

5. To identify and define stop words in the English language, run the following code:

```
In [6]: #identying and defining stop word in english langauge

        stop_words = nltk.corpus.stopwords.words('english')
        print(stop_words)

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'y
ourself', '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', 'a
n', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'b
etween', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'of
f', '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', 'ar
en', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "have
n't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "should
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

6. Having explored the pre-processing steps, the texts were split into two separate parts called tokens and cleaned out by removing the stop words using the below code:

```
In [7]: #preprocessing the texts by splitting them into seperate parts named tokens and cleaning out these tokens by removing the stop wo

def preprocess_text(text):
  tokenized_document = nltk.tokenize.RegexpTokenizer('[a-zA-Z0-9\']+').tokenize(text)
  cleaned_tokens = [word.lower() for word in tokenized_document if word.lower() not in stop_words]
  stemmed_text = [nltk.stem.PorterStemmer().stem(word) for word in cleaned_tokens]
  return stemmed_text
```

For **Implementation in Python**, We'll import the **SentimentIntensityAnalyzer** class from **NLTK's VADER**, which returns a dictionary of sentiment scores, including the overall compound score, a positive score, a negative score, and neutral scores, and retrieve the first few rows with *reviews.head()* and *reviews.tail()* for last few rows.

```
In [8]: from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiment = SentimentIntensityAnalyzer()
```

```
In [9]: reviews['compound'] = [sentiment.polarity_scores(review)['compound'] for review in reviews['Review']]
reviews['neg'] = [sentiment.polarity_scores(review)['neg'] for review in reviews['Review']]
reviews['neu'] = [sentiment.polarity_scores(review)['neu'] for review in reviews['Review']]
reviews['pos'] = [sentiment.polarity_scores(review)['pos'] for review in reviews['Review']]

reviews.head()
```

Out[9]:

	ID	Review Date	Location	Hotel/Restaurant name	Review	compound	neg	neu	pos
0	rn579769293	Reviewed 1 week ago	Cape Panwa	The Cove Phuket	I finally made to to The Cove after hearing fr...	0.9388	0.000	0.685	0.315
1	rn578446147	Reviewed 1 week ago	Cape Panwa	The Cove Phuket	The food and the service were both excellent a...	0.9215	0.061	0.643	0.296
2	rn578261388	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	Almost confused with the corner bar but what a...	0.8944	0.030	0.768	0.202
3	rn578201696	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	I know your probably going to reply with all t...	0.8563	0.058	0.727	0.215
4	rn577322860	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	Super service and great food! Nice views and c...	0.9549	0.000	0.479	0.521

```
In [10]: reviews.tail()
```

Out[10]:

	ID	Review Date	Location	Hotel/Restaurant name	Review	compound	neg	neu	pos
2988	rn145486095	Reviewed November 15, 2012	Choeng Thale	9' Sea Breeze	this great little place on surin beach serves ...	0.8292	0.000	0.792	0.208
2989	rn145450467	Reviewed November 15, 2012	Choeng Thale	9' Sea Breeze	If you're looking for a good food, great atmos...	0.9459	0.000	0.640	0.360
2990	rn145136705	Reviewed November 11, 2012	Choeng Thale	9' Sea Breeze	Meeting different people from all over the wor...	0.7269	0.053	0.667	0.280
2991	rn145067924	Reviewed November 10, 2012	Choeng Thale	9' Sea Breeze	This has to be the best sports bar restaurant ...	0.9796	0.000	0.572	0.428
2992	rn144513470	Reviewed November 4, 2012	Choeng Thale	9' Sea Breeze	This is a great little restaurant with loads o...	0.9578	0.000	0.662	0.338

7. To gain more insights into the sentiment scores of the new columns, run the following code:

```
In [11]: #Exploring the new columns for more insights into sentiment scores for review data

reviews[['compound', 'neg', 'neu', 'pos']].describe()
```

Out[11]:

	compound	neg	neu	pos
count	2993.000000	2993.000000	2993.000000	2993.000000
mean	0.708503	0.022771	0.730721	0.246509
std	0.378836	0.045676	0.126106	0.133492
min	-0.959400	0.000000	0.286000	0.000000
25%	0.662400	0.000000	0.648000	0.147000
50%	0.865800	0.000000	0.737000	0.241000
75%	0.933600	0.037000	0.819000	0.336000
max	0.990900	0.407000	1.000000	0.714000

8. We will create a function to compute negative, neutral, and positive sentiments. Then, add a new column named Sentiment to the dataset using the code below:

```
In [12]: #creating a funtion to compute negative, neutral and positive sentiments and add a new colum named sentiment to our dataset

def getAnalysis(score):
 if score < 0:
    return 'Negative'
 elif score == 0:
    return 'Neutral'
 else:
    return 'Positive'

reviews['sentiment'] = reviews['compound'].apply(getAnalysis)
```

9. View the dataset again to see the new columns.

```
In [13]: #viewing the new dataset
reviews.head()
```

Out[13]:

	ID	Review Date	Location	Hotel/Restaurant name	Review	compound	neg	neu	pos	sentiment
0	rn579769293	Reviewed 1 week ago	Cape Panwa	The Cove Phuket	I finally made to to The Cove after hearing fr...	0.9388	0.000	0.685	0.315	Positive
1	rn578446147	Reviewed 1 week ago	Cape Panwa	The Cove Phuket	The food and the service were both excellent a...	0.9215	0.061	0.643	0.296	Positive
2	rn578261388	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	Almost confused with the corner bar but what a...	0.8944	0.030	0.768	0.202	Positive
3	rn578201696	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	I know your probably going to reply with all t...	0.8563	0.058	0.727	0.215	Positive
4	rn577322860	Reviewed 2 weeks ago	Cape Panwa	The Cove Phuket	Super service and great food! Nice views and c...	0.9549	0.000	0.479	0.521	Positive

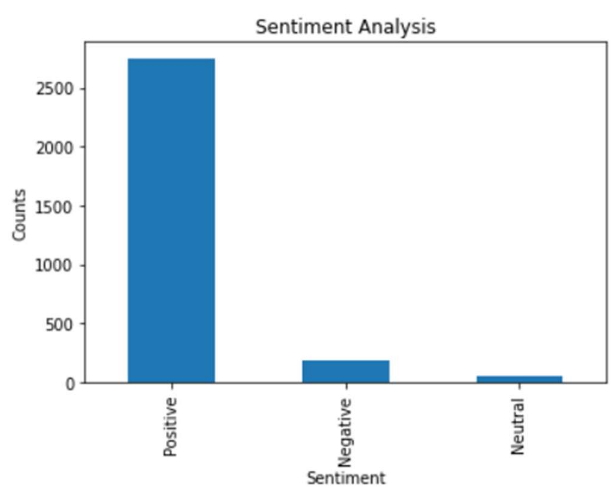
10. To view the counts for each sentiment type, run the below code.

```
In [14]: #Counts for each sentiment type

reviews['sentiment'].value_counts()
```

```
Out[14]: Positive    2749
Negative     191
Neutral       53
Name: sentiment, dtype: int64
```

11. We can visualise the counts for each sentiment type in a bar graph using the following code:

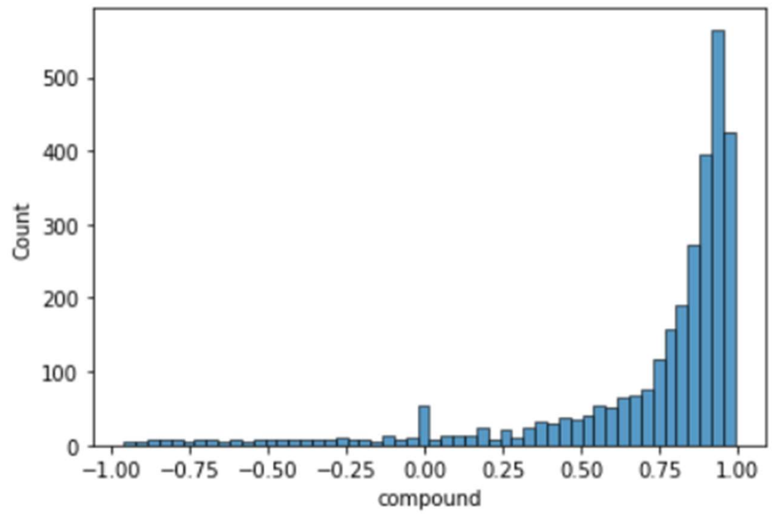
```
In [15]: #visualise the counts for each sentiment type

plt.title('Sentiment Analysis')
plt.xlabel('Sentiment')
plt.ylabel('Counts')
reviews['sentiment'].value_counts().plot(kind = 'bar')
plt.show()
```



12. We can further view the distribution of compound, positive, and negative scores by executing the codes shown in the images below:

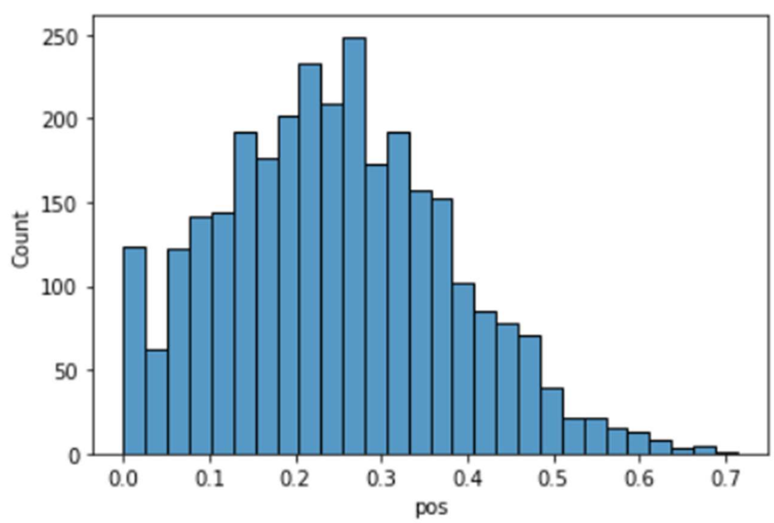
(a) Compound score

```
In [16]: #visualise distribution of compound scores

         sns.histplot(reviews['compound'])

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7f22c79bd8b0>
```



(b) Positive score

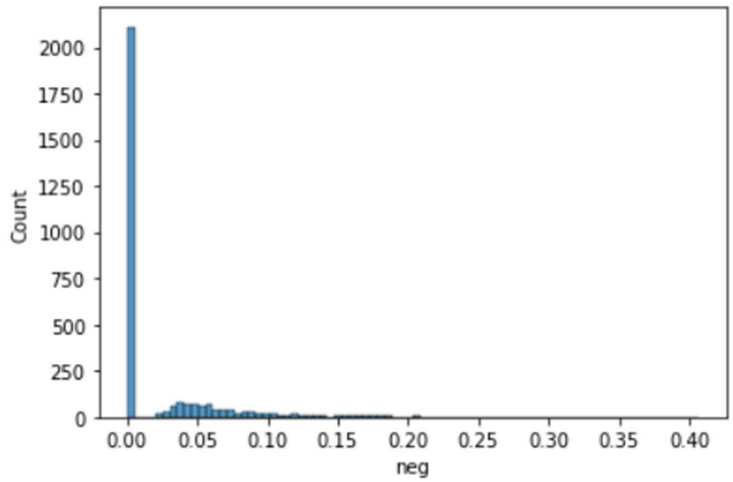
```
In [17]: #visualise distribution of positive scores

         sns.histplot(reviews['pos'])

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f22c78cce80>
```



(c) Negative score

```
In [18]: #visualise distribution of negative scores

         sns.histplot(reviews['neg'])
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7f22c77ff6a0>
```



13. Let's look at how many negative reviews there are per hotel/restaurant using the following code below:

```
In [19]: #negative reviews per hotel/Restaurant

         (reviews['compound']<=0).groupby(reviews['Hotel/Restaurant name']).sum()
```

```
Out[19]: Hotel/Restaurant name
         360 ° Bar                          5
         9' Sea Breeze                     11
         Ann's Kitchen Bar and Grill        5
         Audy Restaurant                    5
         Baan Ra Tree Restaurant           12
         Baba Soul Food                     4
         Bamboo Bar                         9
         Bampot Kitchen & Bar               3
         Benny's American Bar & Grill       5
         Black Cat                         10
         Bocconcino                         7
         Bodega & Grill                     9
         Cafe de Bangtao                   10
         Chilli Kitchen                     7
         Curry Night Indian Restaurant      3
         Cut Grill & Lounge                11
         D Restaurant                      11
         DaVinci Restaurant                 5
         DeDos                              4
         Dino Park                         16
         Live India Indian Restaurant      10
         Mali Chic Restaurant              12
         Panwa House                        8
         Plum Prime Steakhouse              7
         Sabai Sabai                       12
         Sea Breeze                         9
         The Cove Phuket                    2
         The Grill                          5
         Tree Top Restaurant and Bar        6
         Uncle Nan's Italian Restaurant    21
         Name: compound, dtype: int64
```

14. Apparently, we should also look at the number of negative reviews as a proportion of the total number for each hotel/restaurant, unless it's already known. To do this, run the code in the image below:

In [20]:
```
#calculate the negative as a percentage of the total reviews

percent_negative = pd.DataFrame((reviews['compound']<=0).groupby(reviews['Hotel/Restaurant name']).sum()
                                /reviews['Hotel/Restaurant name'].groupby(reviews['Hotel/Restaurant name']).count()*100,
                                columns=['% negative reviews']).sort_values(by='% negative reviews')
percent_negative
```

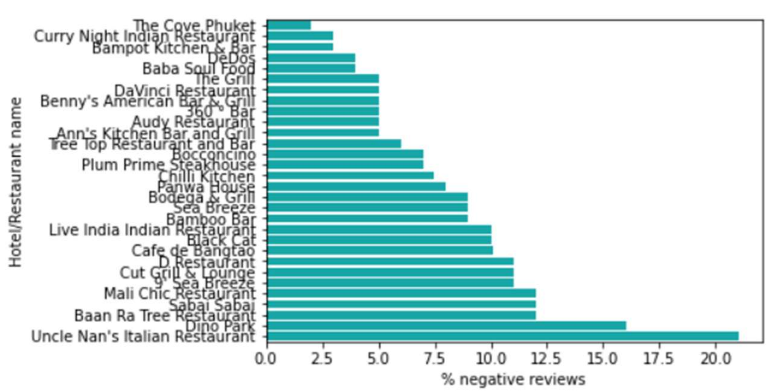
Out[20]:

Hotel/Restaurant name	% negative reviews
The Cove Phuket	2.000000
Curry Night Indian Restaurant	3.000000
Bampot Kitchen & Bar	3.000000
DeDos	4.000000
Baba Soul Food	4.000000
The Grill	5.000000
DaVinci Restaurant	5.000000
Benny's American Bar & Grill	5.000000
360 ° Bar	5.000000
Audy Restaurant	5.000000
Ann's Kitchen Bar and Grill	5.000000
Tree Top Restaurant and Bar	6.000000
Bocconcino	7.000000
Plum Prime Steakhouse	7.000000
Chilli Kitchen	7.446809
Panwa House	8.000000
Bodega & Grill	9.000000
Sea Breeze	9.000000
Bamboo Bar	9.000000
Live India Indian Restaurant	10.000000
Black Cat	10.000000
Cafe de Bangtao	10.101010
D Restaurant	11.000000
Cut Grill & Lounge	11.000000
9' Sea Breeze	11.000000
Mali Chic Restaurant	12.000000
Sabai Sabai	12.000000
Baan Ra Tree Restaurant	12.000000
Dino Park	16.000000
Uncle Nan's Italian Restaurant	21.000000

15. We can further plot the above percentage as a horizontal barplot using seaborn.

In [21]:
```
#we can also plot the above percentages as a horizontal barplot

sns.barplot(data=percent_negative, x='% negative reviews', y=percent_negative.index, color='c')
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x7f22c76ceaf0>



16. We also look into understanding what words appeared more frequently in positive or negative reviews for a particular hotel/restaurant. Then use the Wordcloud to visualise it. In this report, we will focus on Uncle Nan's Italian restaurant as this got the most negative reviews among the restaurants in the Cape Panwa location. To achieve this, execute the codes as shown in the image below.

In [22]:
```
#process the text data ready for wordcloud visualisation using the function defined earlier
#We'll focus on Uncle Nan's Italian restaurant as this got most negative reviews among the restaurants in cape panwa location


reviews['processed_review'] = reviews['Review'].apply(preprocess_text)

reviews_positive_subset = reviews.loc[(reviews['Hotel/Restaurant name']=='Uncle Nan\'s Italian Restaurant')
                              & (reviews['compound']>0),:]

reviews_negative_subset = reviews.loc[(reviews['Hotel/Restaurant name']=='Uncle Nan\'s Italian Restaurant')
                              & (reviews['compound']<=0),:]

reviews_positive_subset.head()
```

Out[22]:

	ID	Review Date	Location	Hotel/Restaurant name	Review	compound	neg	neu	pos	sentiment	processed_review
1300	rn576095102	Reviewed 3 weeks ago	Cape Panwa	Uncle Nan's Italian Restaurant	Needed some non Thai food one evening and went...	0.7878	0.000	0.655	0.345	Positive	[need, non, thai, food, one, even, went, certa...
1301	rn575028373	Reviewed 4 weeks ago	Cape Panwa	Uncle Nan's Italian Restaurant	This restaurant is at the Kantary Bay Hotel. I...	0.8495	0.030	0.796	0.174	Positive	[restaur, kantari, bay, hotel, alway, littl, s...
1302	rn572806144	Reviewed April 12, 2018	Cape Panwa	Uncle Nan's Italian Restaurant	I think my daughter made a better pizza when s...	0.0258	0.129	0.737	0.134	Positive	[think, daughter, made, better, pizza, 6yr, ol...
1306	rn564703288	Reviewed March 5, 2018	Cape Panwa	Uncle Nan's Italian Restaurant	Food and setting was lovely the only downside ...	0.5106	0.033	0.861	0.106	Positive	[food, set, love, downsid, portion, could, big...
1307	rn563337120	Reviewed February 28, 2018	Cape Panwa	Uncle Nan's Italian Restaurant	We booked through cape panwa hotel .we had an ...	0.9501	0.000	0.656	0.344	Positive	[book, cape, panwa, hotel, excel, meal, staff,...

17. We can now generate a Wordcloud using the Wordcloud library for negative reviews at Uncle Nan's Italian Restaurant using the code below.

In [23]:
```
# Wordcloud of words from negative reviews for Uncle Nan's Italian Restaurant

neg_tokens = [word for review in reviews_negative_subset['processed_review'] for word in review]

wordcloud = WordCloud(background_color='white').generate_from_text(' '.join(neg_tokens))

plt.figure(figsize=(12,12))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

We can see that some of the words aren't particularly informative. for example: 'food', 'pizza', 'Italian, and 'pasta' are all references to the restaurant menu. But we can also see words like 'disappoint', 'worst', 'expensive', and 'taste' have been mentioned a few times – Maybe these are issues that require further investigation.

18. We can also generate a wordcloud from the positive reviews using the code below.

```
In [24]: #wordcloud of words from positive reviews for Uncle Nan's Italian Restaurant

         pos_tokens = [word for review in reviews_positive_subset['processed_review'] for word in review]

         wordcloud = WordCloud(background_color='white').generate_from_text(
             ' '.join(pos_tokens))

         plt.figure(figsize=(12,12))
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis("off")
         plt.show()
```

19. Wordclouds do provide a way to visualise word frequencies, but sometimes it might be difficult to explain. we can also use the tabulate method to understand the most frequent words, and the number of occurrences in each with the use of FreqDist from **NLTK** using the code below:

(a) Positive Reviews

```
In [25]: #using FreqDist from NLTK to show the frequeny of words in a tabular form for positive reviews

         from nltk.probability import FreqDist

         pos_freqdist = FreqDist(pos_tokens)

         pos_freqdist.tabulate(10)
```

food	good	italian	pizza	servic	nice	staff	restaur	meal	hotel
59	34	31	26	21	19	18	17	17	16

(b) Negative Reviews

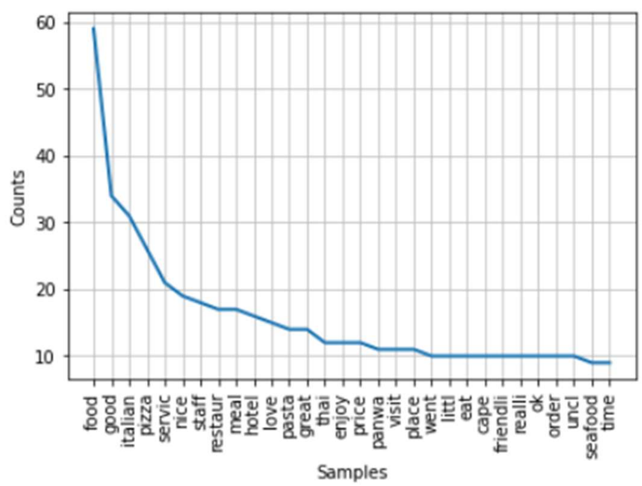
In [26]:
```
#using FreqDist from NLTK to show the frequeny of words in a tabular form for negative reviews

from nltk.probability import FreqDist

neg_freqdist = FreqDist(neg_tokens)

neg_freqdist.tabulate(10)
```

```
    food  italian    pizza  restaur   servic    order     went    pasta disappoint     slow
      18       10        9        7        7        6        6        6          5        5
```

20. We can also use the frequency distribution graph to show word frequency in both positive and negative reviews using the following codes below:

(a) Positive reviews

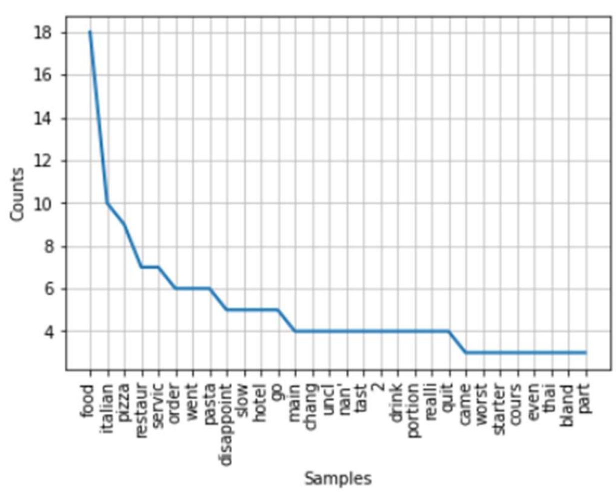
In [27]:
```
#Frequency distribution graph to show word frequency for postive reviews

pos_freqdist.plot(30)
```



Out[27]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f22c7fb9fd0>`

(b) Negative reviews

In [28]:
```
#Frequency distribution graph to show word frequency for negative reviews

neg_freqdist.plot(30)
```



Out[28]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f22c795cee0>`

Results analysis and discussion

Our dataset did not require any pre-processing as we used the implementation of VADER, which takes care of the process. We utilised tokenization, stopword removal, and stemming to generate a wordcloud that visualises the distribution of words and word count. This helped to identify the main words that contribute to positive and negative reviews at Uncle Nan's Italian Restaurant. The management can use this information to investigate and improve specific areas.

Conclusions

For the implementation of this analysis, we chose VADER due to its efficiency, ease of use, and clear, compelling results. This report aims to assist restaurants and hotels in Thailand in identifying their strengths and weaknesses, allowing them to improve their quality of service in the right areas and ultimately leading to higher positive scores.