Nanyang Polytechnic
Post-Diploma Certificate in Applied Data Science
ITD214 Applied Data Science Project
Final Project Presentation
26 Feb 2025

Group 4
Hazizul Humayun S/O Rajaa Mohaamed (Admission No.1077787V)
Ong Teng Teng (Admission No. 6239822P)
Wong Shao Mun (Admission No. 1038987U)

Outline

1. Business Problem, Dataset and Data Cleaning (Group)
2. Model Design (Individual)
3. Model Assessment (Individual)
4. Evaluation and Recommendations (Group)

Business Understanding (Group)

Scenario Background

1. Hotels in the USA have collected quantitative data (reviews.rating) and qualitative data (reviews.text) over a period of 17 years
2. Would like to see what actionable insights can be derived from the collected data

Business Understanding (Group)

Business Goal

- To help USA hotels to improve their service

Business Objectives

1. Identify what key topics consumers typically reveal in their reviews (topic modelling by Hazizul)
2. Predict whether a review is a positive or negative sentiment (sentiment analysis by Teng Teng)
3. Identify time period that drives positive or negative sentiment (time series analysis by Shao Mun)

Data Understanding & Selection (Group)

1. Data Collection Sources
2. Acquire/Select Data
3. Data Fields Description
4. Data Exploration
5. Data Quality

Data Collection Sources

- Kaggle

Acquire/Select Data

1. Searched for reviews and found hotel reviews dataset at

   https://www.kaggle.com/datasets/datafiniti/hotel-reviews

2. Three datasets after download, chose dataset where reviews.rating were integers

	B	C	D	E	F	G	H	I
1	Data title = 7282_1	total row = 34 292		Data title = Datafiniti_Hotel_Reviews_Jun19	total row = 10 000		Data title = Datafiniti_Hotel_Reviews	total row = 10 000
2	Variables	Findings		Variables	Findings		SourceURLs	Findings
3	address	No Null / Empty Data		id	No Null / Empty Data		id	No Null / Empty Data
4	categories	No Null / Empty Data		dateAdded	No Null / Empty Data		dateAdded	No Null / Empty Data
5	city	No Null / Empty Data		dateUpdated	No Null / Empty Data		dateUpdated	No Null / Empty Data
6	country	No Null / Empty Data		address	No Null / Empty Data		address	No Null / Empty Data
7	latitude	Empty data = 86		categories	No Null / Empty Data		categories	No Null / Empty Data
8	longitude	Empty data = 86		primaryCategories	No Null / Empty Data		primaryCategories	No Null / Empty Data
9	name	No Null / Empty Data		city	No Null / Empty Data		city	Only US
10	postalCode	Empty data = 55		country	Only US		country	No Null / Empty Data
11	province	No Null / Empty Data		keys	No Null / Empty Data		keys	No Null / Empty Data
12	reviews.date	Empty cell = 259		latitude	No Null / Empty Data		latitude	No Null / Empty Data
13	reviews.dateAdded	No Null / Empty Data		longitude	No Null / Empty Data		longitude	No Null / Empty Data
14	reviews.doRecommend	Blank		name	No Null / Empty Data		name	No Null / Empty Data
15	reviews.id	Blank		postalCode	No Null / Empty Data		postalCode	No Null / Empty Data
16	reviews.rating	from 1 to 10 with decimal numbers		province	No Null / Empty Data		province	No Null / Empty Data
17	reviews.text	Empty data = 20		reviews.date	No Null / Empty Data		reviews.date	No Null / Empty Data
18	reviews.title	Empty data = 1620		reviews.dateAdded	Blank		reviews.dateSeen	No Null / Empty Data
19	reviews.userCity	Blank		reviews.dateSeen	No Null / Empty Data		reviews.rating	1 to 5 with decimals dig
20	reviews.username	Empty data = 42		reviews.rating	1 to 5		reviews.sourceURLs	No Null / Empty Data

Acquire/Select Data

Name	Date modified	Type	Size
7282_1	3/2/2025 8:33 pm	Microsoft Excel Com...	16,161 KB
Datafiniti_Hotel_Reviews	3/2/2025 8:33 pm	Microsoft Excel Com...	48,404 KB
Datafiniti_Hotel_Reviews_Jun19	3/2/2025 8:34 pm	Microsoft Excel Com...	121,536 KB

Data Fields Description

- Shape (10000, 26)
- Number of rows: 10000
- Number of columns: 26

Data Fields Description

```
df.dtypes:
id                          object
dateAdded                   object
dateUpdated                 object
address                     object
categories                  object
primaryCategories           object
city                        object
country                     object
keys                        object
latitude                   float64
longitude                  float64
name                        object
postalCode                  object
province                    object
reviews.date                object
reviews.dateAdded          float64
reviews.dateSeen            object
reviews.rating               int64
reviews.sourceURLs          object
reviews.text                object
reviews.title               object
reviews.userCity            object
reviews.userProvince        object
reviews.username            object
sourceURLs                  object
websites                    object
dtype: object
```

Data Fields Description

- Three fields with lesser categories shortlisted to explore further for modelling: 'primaryCategories', 'province' and 'reviews.rating'

```
Number of unique values in columns of
df:
id                         1433
dateAdded                  1341
dateUpdated                1397
address                    1432
categories                  631
primaryCategories             4
city                        842
country                       1
keys                       1433
latitude                   1430
longitude                  1431
name                       1311
postalCode                 1149
province                     46
```

```
reviews.date               3370
reviews.dateAdded             0
reviews.dateSeen            701
reviews.rating                5
reviews.sourceURLs         8228
reviews.text               9770
reviews.title              8470
reviews.userCity           3101
reviews.userProvince        244
reviews.username           9222
sourceURLs                 1433
websites                   1327
dtype: int64
```

Data Exploration

```
# Print unique values in columns of df where unique values are <= 50
for col in df.columns:
  unique_values = df[col].nunique()
  if unique_values <= 50:
    print(f"Unique values in column '{col}': {df[col].unique()}")
```

```
Unique values in column 'primaryCategories': ['Accommodation & Food Services'
 'Accommodation & Food Services,Arts Entertainment & Recreation'
 'Accommodation & Food Services,Administrative & Support & Waste Management & Remediation'
 'Accommodation & Food Services,Agriculture']
Unique values in column 'country': ['US']
```

Data Exploration

1. 'reviews.rating' most promising categorical variable to use as five categories can be easily grouped
2. 'province' not attractive because need spend effort to group 46 categories further

```
Unique values in column 'country': ['US']

Unique values in column 'province': ['CA' 'KY' 'LA' 'CO' 'IL' 'IN' 'FL' 'AK'
'GA' 'AL' 'AZ' 'AR' 'OR' 'WA'

 'UT' 'TX' 'TN' 'SC' 'PA' 'OH' 'NY' 'NM' 'MD' 'MI' 'MS' 'MO' 'IA' 'VA'

 'WI' 'HI' 'ID' 'NV' 'WV' 'WY' 'KS' 'MN' 'NE' 'ND' 'DE' 'OK' 'NC' 'MT'

 'SD' 'RI' 'NJ' 'MA']

Unique values in column 'reviews.dateAdded': [nan]

Unique values in column 'reviews.rating': [3 4 5 2 1]
```

Data Quality

- Majority of fields have all rows filled, especially those with potential for modelling: 'reviews.date', 'reviews.rating' and 'reviews.text' (each with 10k rows)

```
Count number of rows with non-empty
values:
id                      10000
dateAdded               10000
dateUpdated             10000
address                 10000
categories              10000
primaryCategories       10000
city                    10000
country                 10000
keys                    10000
latitude                10000
longitude               10000
name                    10000
postalCode              10000
province                10000
reviews.date            10000
reviews.dateAdded           0
reviews.dateSeen        10000
reviews.rating          10000
reviews.sourceURLs      10000
reviews.text            10000
reviews.title            9999
reviews.userCity        10000
reviews.userProvince     9998
reviews.username        10000
sourceURLs              10000
websites                10000
dtype: int64
```

Model Design and Model Assessment (Individual)

Cutover to individual member's presentation slides:

1. Identify what key topics consumers typically reveal in their reviews (topic modelling by Hazizul)
2. Predict whether a review is a positive or negative sentiment (sentiment analysis by Teng Teng)
3. Identify time period that drives positive or negative sentiment (time series analysis by Shao Mun)

Nanyang Polytechnic
Post-Diploma Certificate in Applied Data Science
ITD214 Applied Data Science Project
Final Project Presentation
26 Feb 2025

Group 4
Individual Presentation
Hazizul Humayun S/O Rajaa Mohaamed (Admission No.1077787V)

Outline

1. Clean Data
2. Construct Data
3. Exploratory Data Analysis (EDA)

Data Cleaning for text analysis

Data Cleaning Column to do the analysis on is reviews.text

1. Removed duplicate reviews to ensure unbiased analysis

```
Duplicates in 'reviews.text' column: 230
```

```
# 2. Data Cleaning
df.drop_duplicates(subset=['reviews.text'], keep='first', inplace=True)  # Remove duplicate reviews
```

2. Text Preprocessing

- Converted Text to Lowercase
- Removed punctuation
- Tokenization (splitting into words)
- Stopword removal (e.g., "the", "and")
- Lemmatization (converting words to base form)

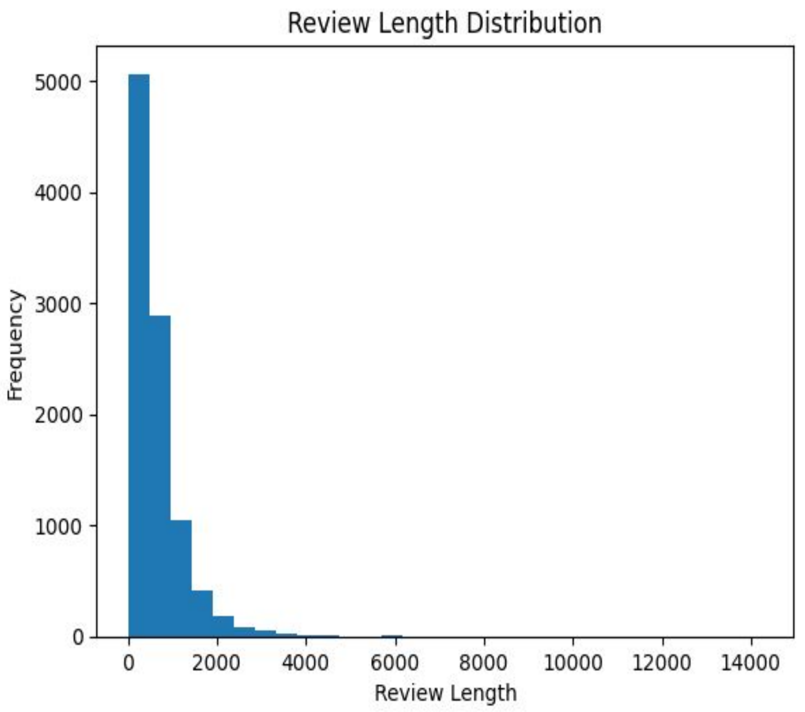
Example:

- ❌ "The rooms were amazing!!!"
- ✅ "room amazing"

```python
# 4. Text Preprocessing
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    text = text.lower()  # Lowercasing
    text = re.sub(r'[^\w\s]', '', text)  # Remove punctuation
    tokens = word_tokenize(text)
    tokens = [lemmatizer.lemmatize(token) for token in tokens if token not in stop_words and len(token) > 2]
    return " ".join(tokens)

df['processed_review'] = df['reviews.text'].apply(preprocess_text)
```

Construction of review length column

```
# 3. Review Length Distribution
df['review_length'] = df['reviews.text'].str.len()  # Character count
plt.hist(df['review_length'], bins=30)
plt.title('Review Length Distribution')
plt.xlabel('Review Length')
plt.ylabel('Frequency')
plt.show()
```



- Counting characters of the review length and frequency of words

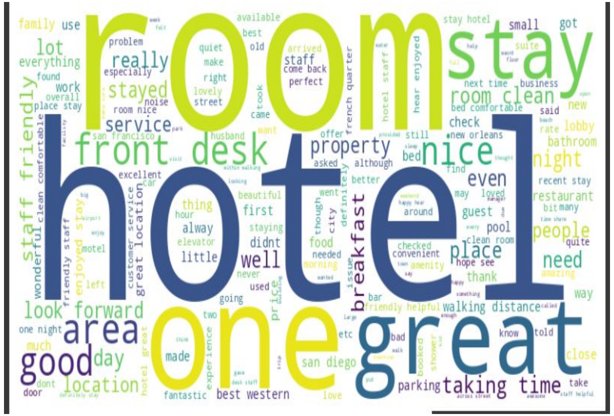
Purpose of studying the review lengths:

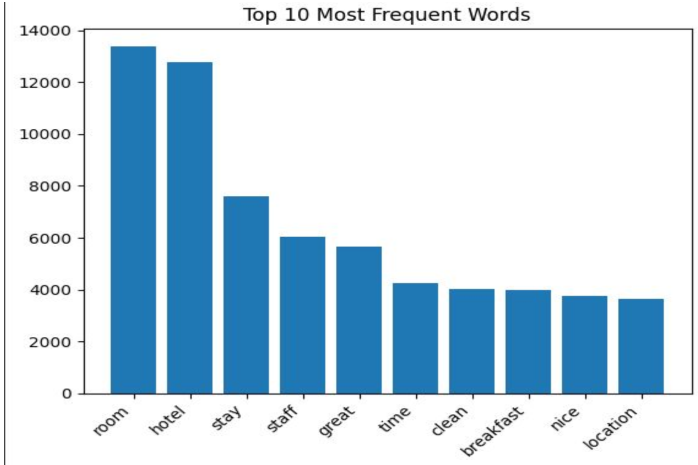
1. Helps to Understand Review Lengths:
   a. Helps analyse the distribution of short vs long reviews.
   b. Identifies whether customers leave detailed feedback or just brief comments.
2. Detecting Fake or Spam Reviews:
   a. Extremely short reviews (e.g., "Good" or "Bad") might indicate low-effort or spam content.
   b. Very long reviews could be fake or overly promotional.
3. Assessing Sentiment vs. Length:
   a. Comparing sentiment scores with character count can reveal if longer reviews tend to be more positive or negative.
4. Improving Text Processing:
   a. Helps determine whether to filter out excessively short reviews for better text analysis (e.g., topic modeling).
5. Optimizing User Experience:
   a. Businesses can encourage detailed feedback if most reviews are too short.

Exploratory Data Analysis: Word Count Frequency

📌 Purpose:

- Identifies key themes in customer reviews.
- Highlights frequently mentioned topics (e.g., "room", "clean", "staff").
- Helps detect positive vs. negative sentiment trends.
- Aids in refining stopword lists for topic modeling.



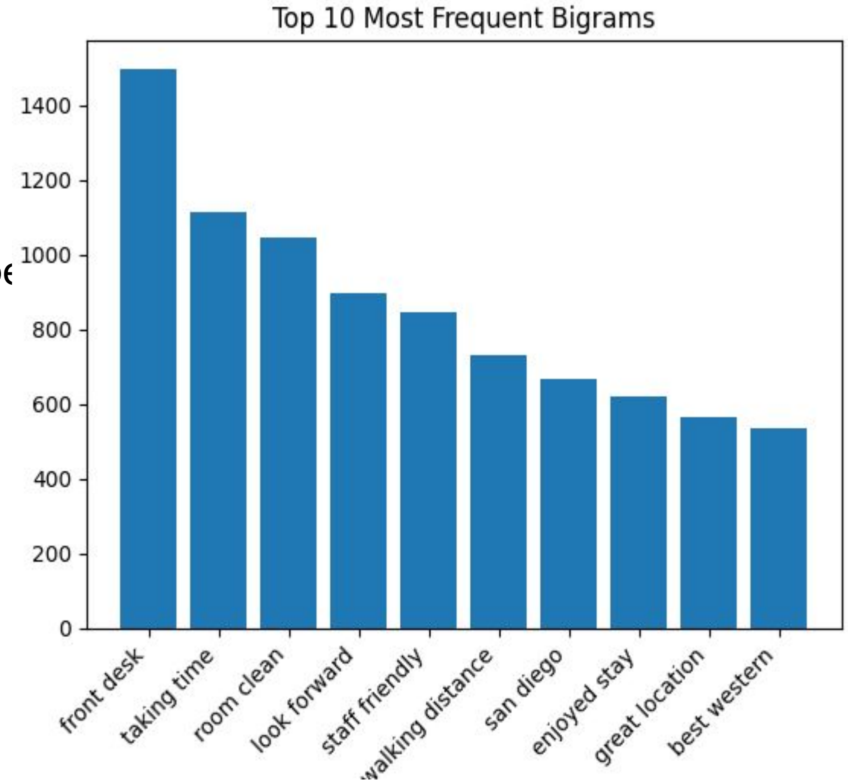


Exploratory Data Analysis: Bigram Analysis

N-Gram (Bigram) Analysis

Identifies common word pairs (e.g., "great service")

- Helps in understanding review themes and deepe



Exploratory Data Analysis: Sentiment Analysis

1. Sentiment Score by Rating

Higher Ratings → Higher Sentiment Scores

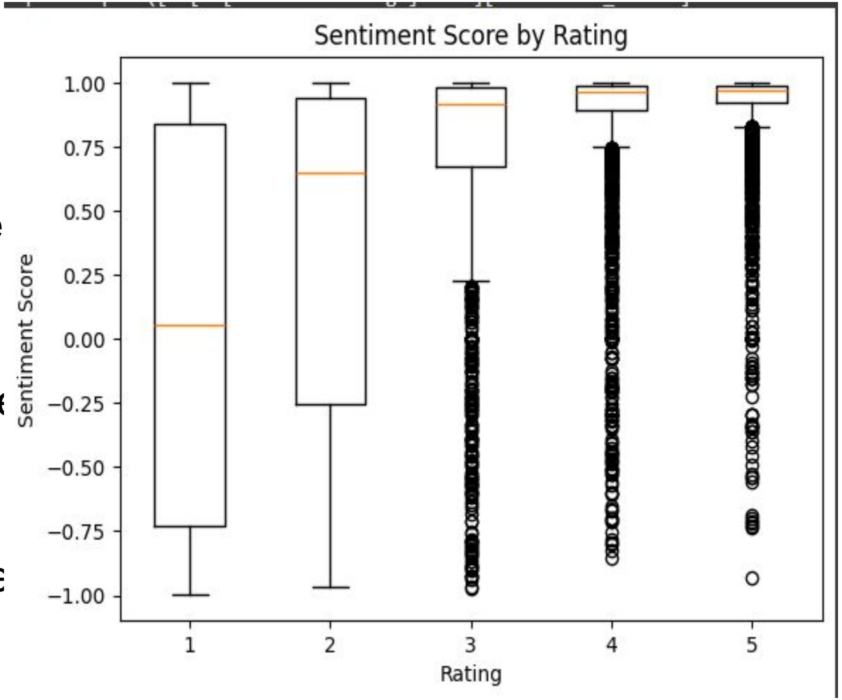
- Ratings 4 & 5 have strong positive sentime

Low Ratings (1 & 2) Show High Variability

- Wide sentiment range indicates mixed expe

Outliers in High Ratings

- Some 4 & 5-star reviews contain neutral/ne feedback or sarcasm.



Exploratory Data Analysis: Sentiment Analysis

2. Sentiment Score by Length of Review

Most Reviews Are Short

- Majority fall below 2000 characters, with a few very long reviews (~14,000 characters)

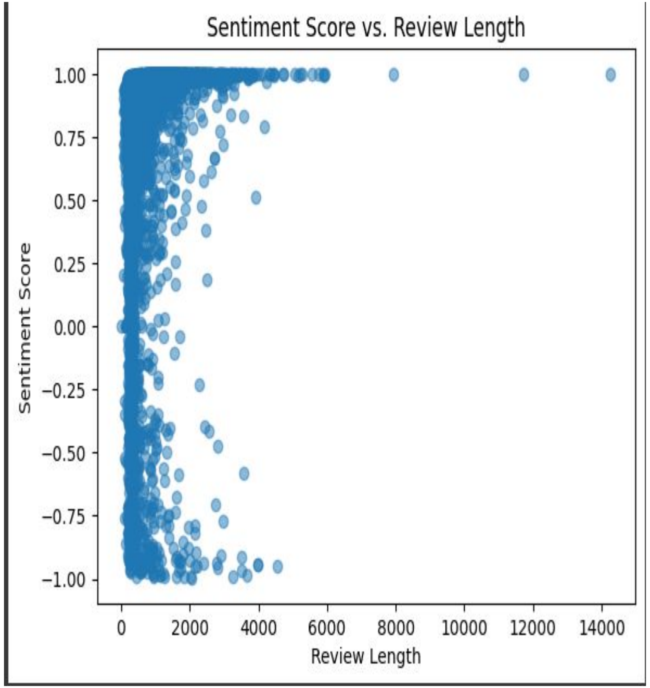
Positive Sentiment Dominates

- Many short reviews have high sentiment scores (~1.0).

Negative Sentiment Appears in Short Reviews

- Short reviews also show low sentiment (~-1.0), possibly indicating strong emotions (either praise or complaints).

Long Reviews Show Mixed Sentiment



Project Plan (Group)

- Hazizul: Actionable Insights for Hotel Management to look based on analysis
  - Identify top concerns (e.g., "bad WiFi", "dirty rooms").
  - Highlight strengths (e.g., "friendly staff", "good location").
  - Recommend areas for improvement (e.g., "upgrade breakfast options").

Topic Modelling with LDA

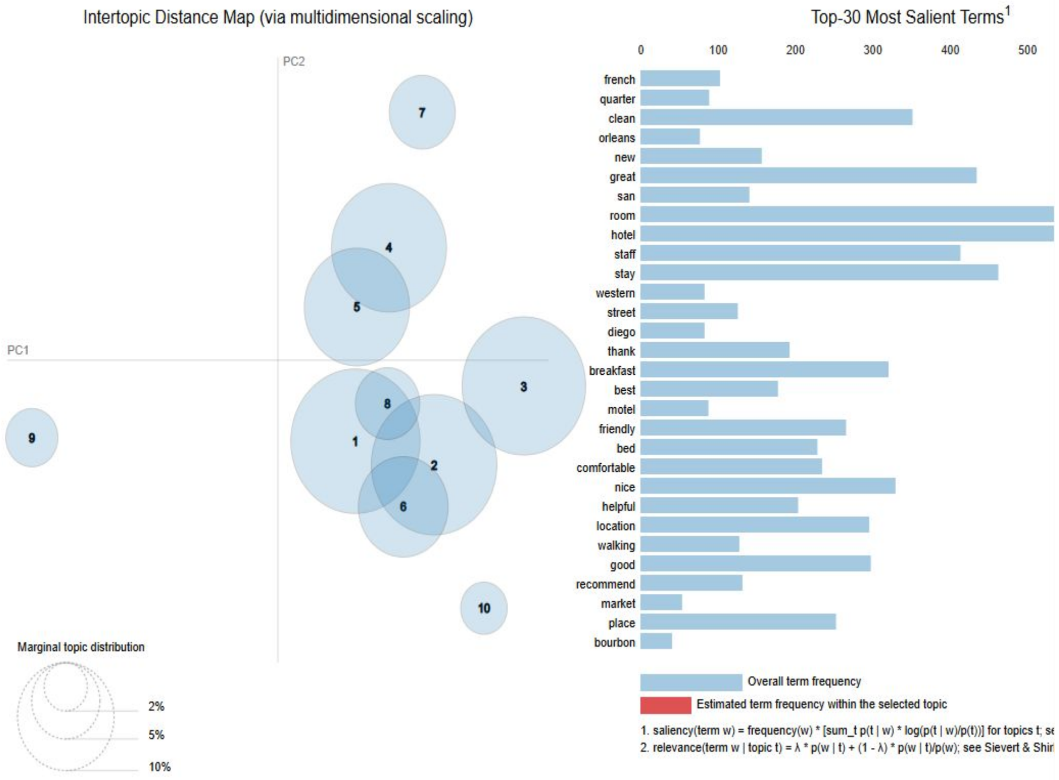
Topic Modelling was invoked with LDA vectorizer to deduce the topics and for a start 10 topics was set (set at 95%) .

```
tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2, max_features=1000)
```

```
lda_model = LatentDirichletAllocation(n_components=10, random_state=42)
```

```
Topic #1:
stay hotel great staff thank time guest forward service hope
Topic #2:
room clean motel older bed old good spring hotel stayed
Topic #3:
room breakfast nice hotel clean bed great coffee comfortable area
Topic #4:
clean staff great hotel stay nice breakfast friendly recommend room
Topic #5:
french quarter orleans new bourbon street historic hotel market river
Topic #6:
room hotel stay desk night would front time one guest
Topic #7:
hotel room great good breakfast clean restaurant nice location staff
Topic #8:
room hotel bed floor night bathroom nice good stay one
Topic #9:
hotel great stay staff beach room time wonderful location enjoyed
Topic #10:
san western diego best stay thank hotel time hope staff
```

Visualising Topics with plyDavis



Topic prevalence:
Based on the circle sizes, Topics 1-3 are the most common topics consisting of positive reviews of the Hotel.

Topic #1:
stay hotel great staff thank time guest forward service hope
Topic #2:
room clean motel older bed old good spring hotel stayed
Topic #3:
room breakfast nice hotel clean bed great coffee comfortable area

Topic Similarity:
Majority of the topics were closely positioned which indicate a similar topic with the only exception of topics 7 ,9 & 10.

Visualising Topics with plyDavis

```
Topic #7:
hotel room great good breakfast clean restaurant nice location staff
Topic #9:
hotel great stay staff beach room time wonderful location enjoyed
Topic #10:
san western diego best stay thank hotel time hope staff
```

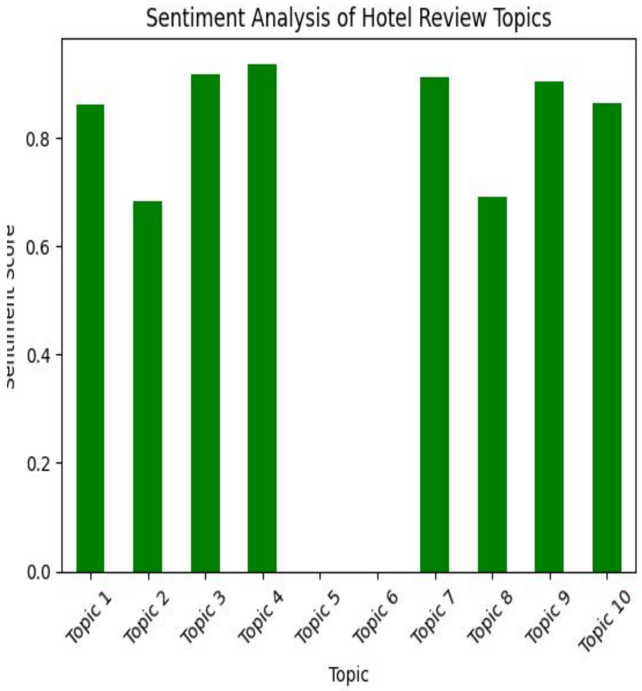
Still Common topics describing hotel amenities and what customers enjoyed the most. No insightful information to gather for the business objective. This is evident in the Vader sentiment analyser done in the next step.

Strategies done to Improve Analysis to meet the other 2 objectives

Adjusting TfidfVectorizer Parameters:

```
Topic #1:
hotel room breakfast good great clean restaurant nice location staff
Topic #2:
room motel dirty smell old carpet smelled bed door bathroom
Topic #3:
room hotel great nice view area staff clean stayed pool
Topic #4:
clean breakfast room nice staff great friendly bed hotel good
Topic #5:
stay hotel thank staff guest time experience great hope feedback
Topic #6:
french quarter hotel orleans new location great street room staff
Topic #7:
room hotel night stay would one time desk front get
Topic #8:
room clean hotel hampton stayed nice great breakfast good comfortable
Topic #9:
hotel stay great staff time room enjoyed thank wonderful review
Topic #10:
airport orlando hilton garden inn hampton shuttle philadelphia flight dallas
```

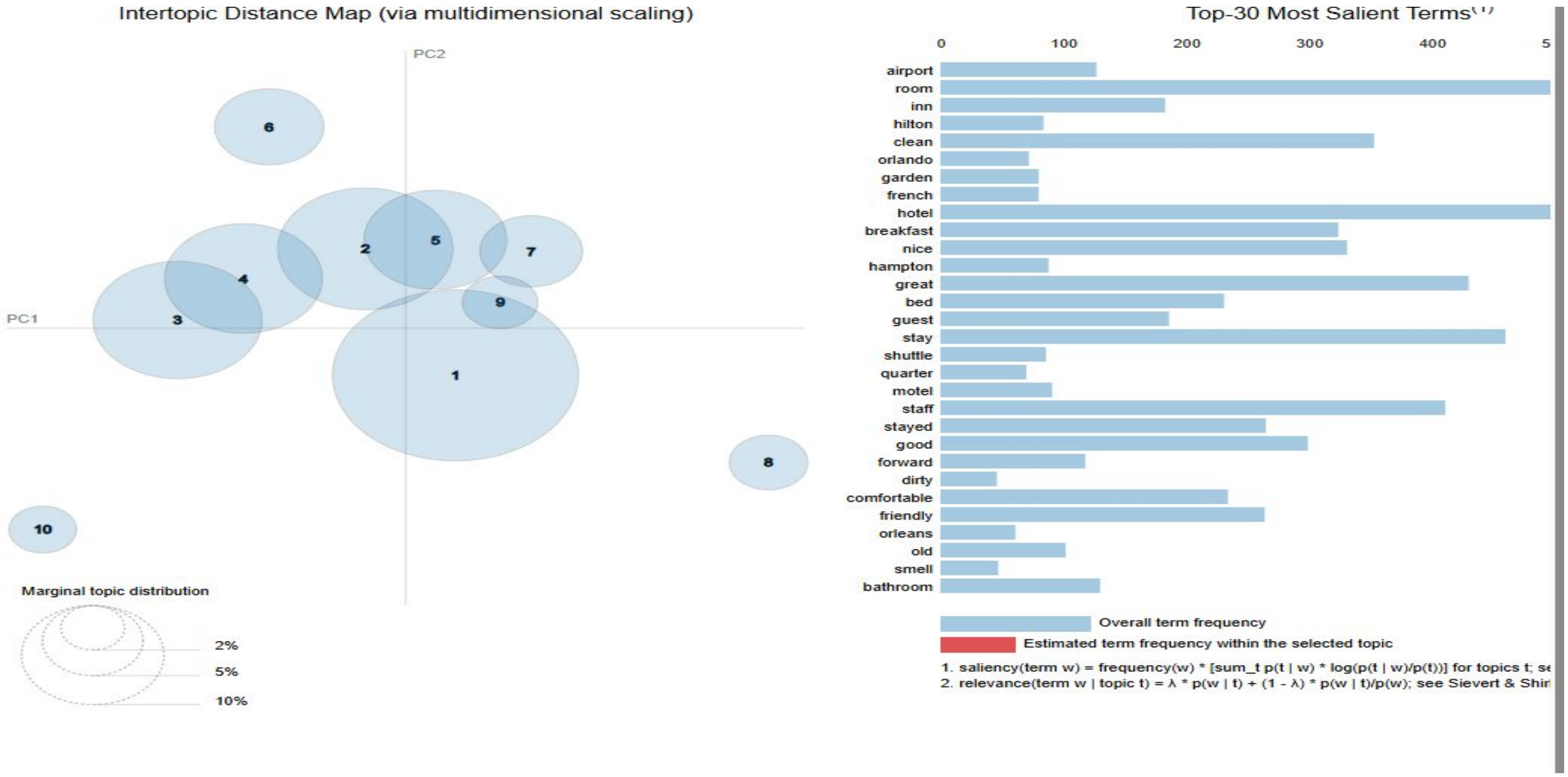
Improving Analysis



Still derived at an overall positive sentiment with the exception of 5 and 6 which were neutral

```
Topic #5:
stay hotel thank staff guest time experience great hope feedback
Topic #6:
french quarter hotel orleans new location great street room staff
```

plyDavis post adjustment



Post analysis Review

Dataset mainly consisted of general words describing of positive sentiment of the hotel and not much insight was gained.

Things done to further improve analysis:

1. Custom stop words to remove words like "hotel", "room" and etc
2. Segregating Topics based on sentiment to perform modelling

Topics post-adjustment

Positive Topics:

Topic 1: staff great clean hampton walking friendly helpful historic stayed French

Topic 1 (suggested): Exceptional Staff & Historic Charm at Hampton

Topic 2: clean motel price disneyland nice good breakfast great del friendly

Topic 2 (suggested): Affordable & Clean Stay Near Disneyland with Great Breakfast

Topic 3: great view nice pool room staff stayed clean location perfect

Topic 3 (suggested): Scenic Views, Pool, and Perfect Location for a Relaxing Stay

Topic 4: guest like staff experience desk service front good feedback nice

Topic 4 (suggested): Positive Guest Experience & Friendly Front Desk Service

Topic 5: staff clean check desk great front nice stayed always friendly

Topic 5 (suggested): Consistently Great Staff and Smooth Check-in Experience

Topic 6: hampton owner ritz motel philadelphia inn stayed friendly atlanta clean

Topic 6 (suggested): Comfortable Stay at Hampton and Ritz, with Friendly Hospitality

Topic 7: good great breakfast clean parking location restaurant nice staff free

Topic 7 (suggested): Great Value Stay with Free Parking, Breakfast & Restaurant Access

Topics post-adjustment

Negative Topics:

Topic 1: half returned fruit nicer saturday eating cozy restaurant meal lake

Topic 1 (suggested): Disappointing Dining Experience & Food Quality Issues

Topic 2: per mind con relax entrance conveniently seaworld tip budget priced

Topic 2 (suggested): Budget Stay with Mixed Convenience & Relaxation Factors

Topic 3: month third unless compared visiting dog still decent pretty twice

Topic 3 (suggested): Inconsistent Experience Across Multiple Visits

Topic 4: dirty smell like bed floor loud water door old someone

Topic 4 (suggested): Cleanliness Concerns: Dirty Rooms & Unpleasant Smells

Topic 5: hall spa one touch couch completely question bit mattress loud

Topic 5 (suggested): Room Comfort Issues: Noise, Mattress, and Spa Discomfort

Topic 6: decided shower luggage large picture fix airport microwave dirty always

Topic 6 (suggested): Maintenance Issues: Broken Showers & Dirty Appliances

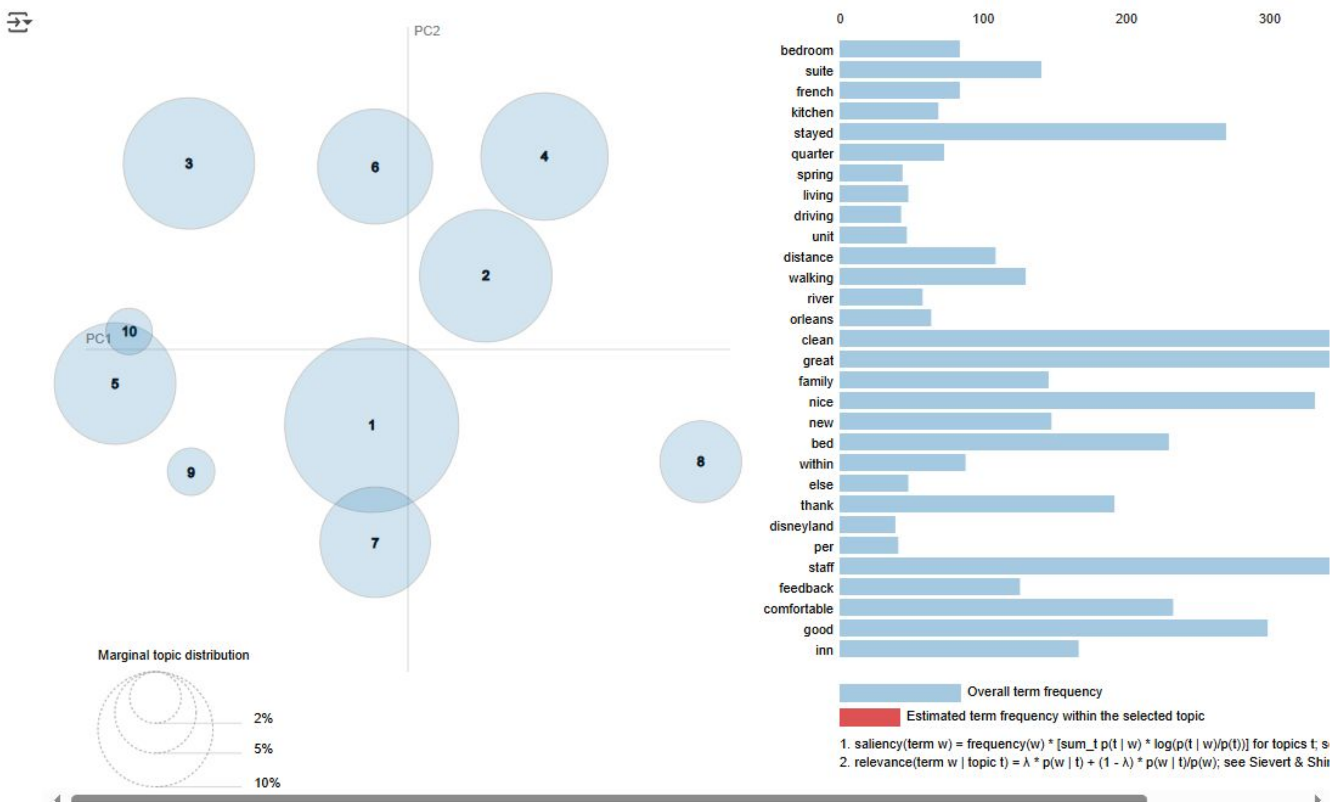
Topic 7: ice machine del smoke personal weather deal general con wedding

Topic 7 (suggested): Facility Problems: Ice Machine, Smoke Issues & Event Stays

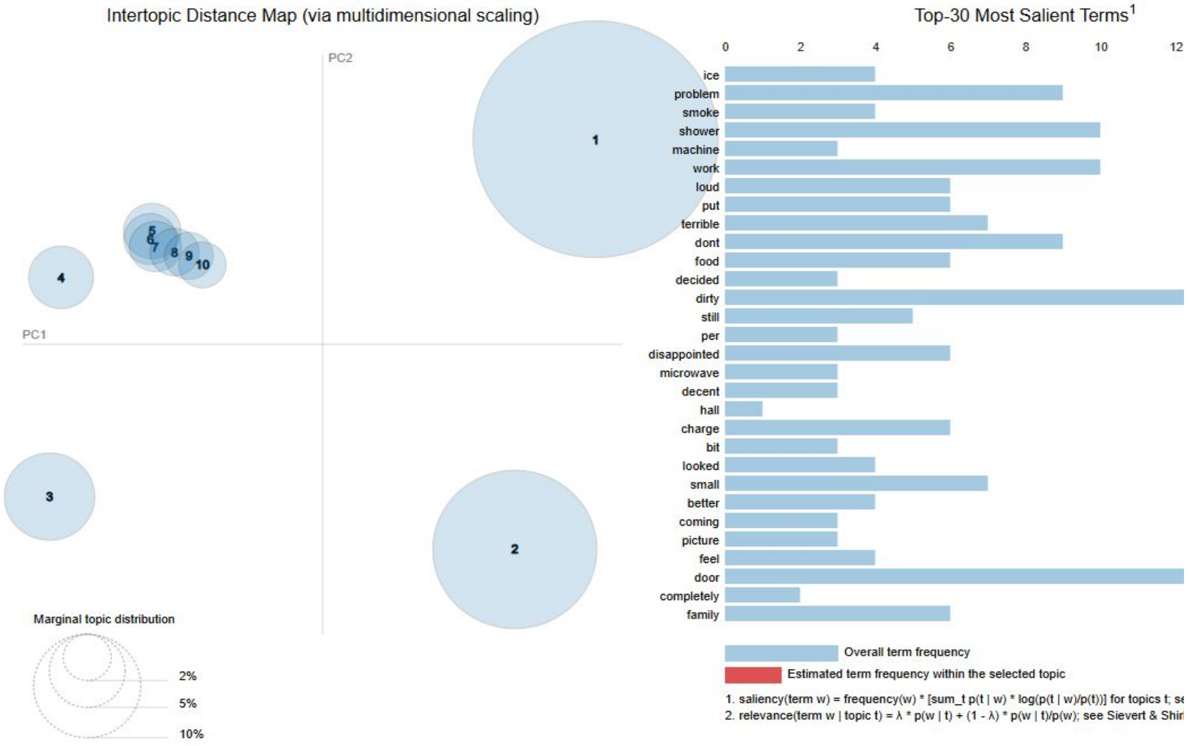
Topic 8: desk room bad didnt front bathroom stayed booked bed told

Topic 8 (suggested): Front Desk & Booking Problems Leading to Guest Dissatisfaction

plyDavis visualisation (Positive Reviews)



plyDavis visualisation (Negative Reviews)



Evaluating the Topics (Positive Reviews)

Topic 1: Exceptional Staff & Historic Charm at Hampton

Topic 2: Affordable & Clean Stay Near Disneyland with Great Breakfast

Topic 3: Scenic Views, Pool, and Perfect Location for a Relaxing Stay

Topic 4: Positive Guest Experience & Friendly Front Desk Service

Topic 5: Consistently Great Staff and Smooth Check-in Experience

Topic 6: Comfortable Stay at Hampton and Ritz, with Friendly Hospitality

Topic 7: Great Value Stay with Free Parking, Breakfast & Restaurant Access

Evaluating the Topics (Negative Reviews)

Topic 1: Disappointing Dining Experience & Food Quality Issues

Topic 2: Budget Stay with Mixed Convenience & Relaxation Factors

Topic 3: Inconsistent Experience Across Multiple Visits

Topic 4: Cleanliness Concerns: Dirty Rooms & Unpleasant Smells

Topic 5: Room Comfort Issues: Noise, Mattress, and Spa Discomfort

Topic 6: Maintenance Issues: Broken Showers & Dirty Appliances

Topic 7: Facility Problems: Ice Machine, Smoke Issues & Event Stays

Topic 8: Front Desk & Booking Problems Leading to Guest Dissatisfaction

Conclusion

To elevate the guest experience and foster greater satisfaction, the following key areas are recommended for enhancement:

- **Cleanliness and Hygiene:** Implement rigorous cleaning protocols, paying meticulous attention to guest rooms, bathrooms, and fitness facilities to ensure a spotless and hygienic environment.
- **Maintenance and Upkeep:** Address maintenance issues swiftly and efficiently, prioritizing prompt repairs for showers, microwaves, and other in-room amenities to guarantee a seamless and comfortable stay.
- **Front Desk Service:** Elevate front desk operations by providing comprehensive customer service training, empowering staff to deliver exceptional service, efficient communication, and a warm, welcoming atmosphere.
- **Breakfast Enhancement:** Expand and enrich breakfast offerings with a wider variety of choices and superior quality ingredients to cater to diverse preferences and elevate the dining experience.
- **Marketing and Promotion:** Capitalize on existing strengths by highlighting exceptional staff friendliness, prime locations, and complimentary amenities in marketing materials to attract and delight potential guests.

Nanyang Polytechnic
Post-Diploma Certificate in Applied Data Science
ITD214 Applied Data Science Project
Final Project Presentation
26 Feb 2025

Group 4
Individual Presentation
Ong Teng Teng (Admission No. 6239822P)

Outline

1. Clean Data
2. Conduct preprocessing steps
3. Prepare word representation
4. Accuracy of review rating

df.shape (10000, 26) and 2 variables to use to do prediction of the data = reviews.text and reviews.rating . df.shape (10000, 26)

	B	C	D	E	F	G	H	I
1	Data title = 7282_1	total row = 34 292		Data title = Datafiniti_Hotel_Reviews_Jun19	total row = 10 000		Data title = Datafiniti_Hotel_Reviews	total row = 10 000
2	Variables	Findings		Variables	Findings		SourceURLs	Findings
3	address	No Null / Empty Data		id	No Null / Empty Data		id	No Null / Empty Data
4	categories	No Null / Empty Data		dateAdded	No Null / Empty Data		dateAdded	No Null / Empty Data
5	city	No Null / Empty Data		dateUpdated	No Null / Empty Data		dateUpdated	No Null / Empty Data
6	country	No Null / Empty Data		address	No Null / Empty Data		address	No Null / Empty Data
7	latitude	Empty data = 86		categories	No Null / Empty Data		categories	No Null / Empty Data
8	longitude	Empty data = 86		primaryCategories	No Null / Empty Data		primaryCategories	No Null / Empty Data
9	name	No Null / Empty Data		city	No Null / Empty Data		city	Only US
10	postalCode	Empty data = 55		country	Only US		country	No Null / Empty Data
11	province	No Null / Empty Data		keys	No Null / Empty Data		keys	No Null / Empty Data
12	reviews.date	Empty cell = 259		latitude	No Null / Empty Data		latitude	No Null / Empty Data
13	reviews.dateAdded	No Null / Empty Data		longitude	No Null / Empty Data		longitude	No Null / Empty Data
14	reviews.doRecommend	Blank		name	No Null / Empty Data		name	No Null / Empty Data
15	reviews.id	Blank		postalCode	No Null / Empty Data		postalCode	No Null / Empty Data
16	reviews.rating	from 1 to 10 with decimal numbers		province	No Null / Empty Data		province	No Null / Empty Data
17	reviews.text	Empty data = 20		reviews.date	No Null / Empty Data		reviews.date	No Null / Empty Data
18	reviews.title	Empty data = 1620		reviews.dateAdded	Blank		reviews.dateSeen	No Null / Empty Data
19	reviews.userCity	Blank		reviews.dateSeen	No Null / Empty Data		reviews.rating	1 to 5 with decimals dig
20	reviews.username	Empty data = 42		reviews.rating	1 to 5		reviews.sourceURLs	No Null / Empty Data

“reviews.text” consists of 10,000 rows with no null and “reviews rating” =<4, 5 , our group have decide it will consider as positive reviews based on Exploratory Data Analysis: Sentiment Analysis . Sentiment Score by Rating, Higher Ratings → Higher Sentiment Scores. Results = Ratings 4 & 5 have strong positive sentimental

Tokenization is the first step in text analytics. The process of breaking down a text paragraph into smaller chunks such as words or sentence is called Tokenization. Token is a single entity that is building blocks for sentence or paragraph.

Sentence tokenizer breaks text paragraph into sentences.Tokenize the first review into sentence

Word tokenizer breaks text paragraph into words

Results as below:

```python
from nltk.tokenize import word_tokenize

tokenized_words = word_tokenize(df['reviews.text'][1])
print(tokenized_words)
print('number of words:' + str(len(tokenized_words)))
```

```
['We stayed in the king suite with the separation between the bedroom and the living space.', "The sofa bed wasn't very good I had back discomfort by the day
number of sentences:4
['We', 'stayed', 'in', 'the', 'king', 'suite', 'with', 'the', 'separation', 'between', 'the', 'bedroom', 'and', 'the', 'living', 'space', '.', 'The', 'sofa',
number of words:67
```

By preprocessing the data, we ensure that the sentiment analysis model receives clean, consistent, and meaningful input, which ultimately leads to more accurate and reliable results !!!

Next, we will loop through all the reviews and create a word list for visualisation. At the same time we will do case normalization to convert all the words/terms into lower case. Loop through all reviews and tokenize into wordsall_words = [word.lower() for sent in df['Review'] for word in word_tokenize(sent)]#print the first 20 words

print(all_words[:20]) - it give us a broad pictures of how the data is on which topics and results shown keys words like “Hotel” / “train” / “near” / “by” and etc

```
#print the first 20 words
print(all_words[:20])
```

```
['this', 'hotel', 'was', 'nice', 'and', 'quiet', '.', 'did', 'not', 'know', ',', 'there', 'was', 'train', 'track', 'near', 'by', '.', 'but', 'it']
```

Frequency distribution will calculate the number of occurence of each word to view high and low frequent words in all the reviews

list out 10 most frequent words and 10 least frequent in the entire list of words

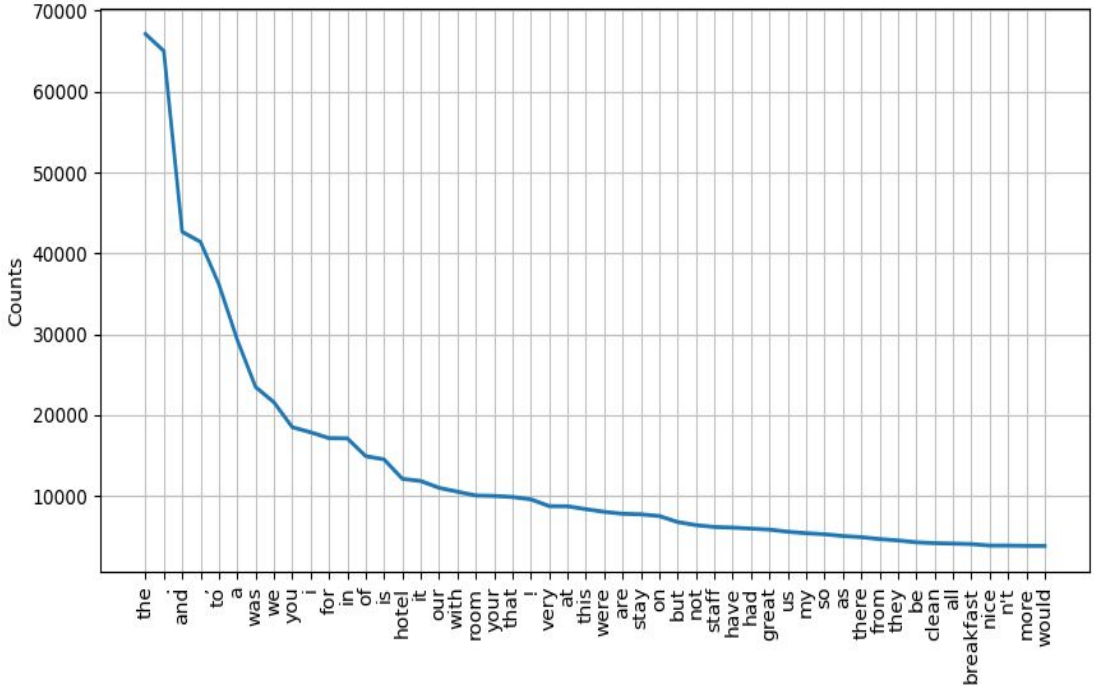
```
# print 10 most frequently occurring words
print ("\nTop 10 most frequently occurring words")
print (all_words_frequency.most_common(10))

# print 10 least frequently occurring words
print ("\nTop 10 least frequently occurring words")
print (all_words_frequency.most_common()[-10:])
```

```
<FreqDist with 30038 samples and 1323309 outcomes>

Top 10 most frequently occurring words
[('the', 67110), ('.', 65010), ('and', 42655), (',', 41408), ('to', 36175), ('a', 29310), ('was', 23466), ('we', 21578), ('you', 18497), ('i', 17862)]

Top 10 least frequently occurring words
[('basslights', 1), ('langley/ft', 1), ('eutis', 1), ('williamsburg', 1), ('parmesan', 1), ('lottery', 1), ('arcade', 1), ('cure', 1), ('boredom', 1), ('doct
```

Create a function to plot the frequency, make it a function as we will be re-using it later.

def plot_frequency freq

```
    def plot_frequency freq
  plt figure figsize= 10 5
  freq        50  cumulative=False
  plt show


plot_frequency all_words_frequency
```



From the above frequency distribution of words, we can see the most frequently occurring words are either punctuation marks or stopwords.

```python
# Prepare the data for modelling using different text features
porter_stemmer = PorterStemmer()
stopwords_english = set(stopwords.words('english'))

common_words =['hotel'] #add common words to stop words
stopwords_english.update(common_words)

def clean(doc):
    all_words_clean = []
    for word in doc:
        if word not in stopwords_english:
          # Using string.punctuation here
            punc_free = ''.join([ch for ch in word if ch not in string.punctuation])
            if len(punc_free)>=2 and not punc_free.isdigit():
                all_words_clean.append(porter_stemmer.stem(punc_free))

    return all_words_clean

df2['reviews.text'] = df2['reviews.text'].apply(lambda x: word_tokenize(x.lower()))
df2['reviews.text'] = df2['reviews.text'].apply(lambda x: clean(x))

df2['reviews.rating'] = np.where(df2['reviews.rating']>= 4, 'Good', 'Bad')
df2.head()
```

Data as shown below that after the text are cleaned with punctuation marks or stopwords and why this step is important? - Humans are able define which punctuation are meaningless but computers are unable to. It will be mislead thinking this words are very important if there are words in uppercase letter and exclamation mark.

...	reviews.dateSeen	reviews.rating	reviews.sourceURLs	reviews.text	reviews.title	reviews.userCity	reviews.userProvince	reviews.userr
...	2018-01-03T00:00:00Z	3	https://www.tripadvisor.com/Hotel_Review-g3243...	This hotel was nice and quiet. Did not know, t...	Best Western Plus Hotel	San Jose	UnitedStates	tatsurok2
...	2016-10-09T00:00:00Z	4	https://www.tripadvisor.com/Hotel_Review-g3217...	We stayed in the king suite with the separatio...	Clean rooms at solid rates in the heart of Carmel	San Francisco	CA	STEPHE
...	2016-10-09T00:00:00Z	3	https://www.tripadvisor.com/Hotel_Review-g3217...	Parking was horrible, somebody ran into my ren...	Business	Prescott Valley	AZ	15Deb
...	2016-10-31T00:00:00Z	5	https://www.tripadvisor.com/Hotel_Review-g3217...	Not cheap but excellent location. Price is som...	Very good	Guaynabo	PR	Wilfred

Removing of Stop words are those frequently words which do not carry any significant meaning in text analysis - For example, I, me, my, the, a, and, is, are, he, she, we, etc.
Using reviews username “STEPHEN N” as our checkpoint - review.text are left with important keys words in our results

reviews.rating	reviews.sourceURLs	reviews.text	reviews.title	reviews.userCity	reviews.userProvince	reviews.username	sour
Bad	https://www.tripadvisor.com/Hotel_Review-g3243...	[nice, quiet, know, train, track, near, train,...	Best Western Plus Hotel	San Jose	UnitedStates	tatsurok2018	https://www.tripadvisor.com/Hotel_
Good	https://www.tripadvisor.com/Hotel_Review-g3217...	[stay, king, suit, separ, bedroom, live, space...	Clean rooms at solid rates in the heart of Carmel	San Francisco	CA	STEPHEN N	http://www.tripadvisor.com/Hotel_ g
Bad	https://www.tripadvisor.com/Hotel_Review-g3217...	[park, horribl, somebodi, ran, rental, car, st...	Business	Prescott Valley	AZ	15Deborah	http://www.tripadvisor.com/Hotel_ g
Good	https://www.tripadvisor.com/Hotel_Review-g3217...	[cheap, excel, locat, price, somewhat, standar...	Very good	Guaynabo	PR	Wilfredo M	http://www.tripadvisor.com/Hotel_ g

By using the selected model to predict new review, we could test the accuracy of the text rating been provided earlier in the data we have chosen for this project

```
# Use the selected model to predict new review
data = {'custom_review': ['I hated the room. The service is bad.',
                          'It was a wonderful stay. I loved it. Good room service.']}

df_test = pd.DataFrame (data, columns = ['custom_review'])
df_test['custom_review'] = df_test['custom_review'].apply(lambda x: word_tokenize(x.lower()))
df_test['custom_review'] = df_test['custom_review'].apply(lambda x: clean(x))

# Apply the same transformations used during training (including PCA)
test_features_tfidf = pd.DataFrame(get_tfidf_features(df_test, 'custom_review'),
                            columns=header.split(','), index = df_test.index)
test_features_pca = pca.transform(test_features_tfidf) # Apply PCA transformation

# Create bow representation for the test data
test_features_bow = pd.DataFrame(get_bow_features(df_test, 'custom_review'),
                            columns=header.split(','), index = df_test.index)
test_bow_pca = pca.transform(test_features_bow)  # Apply PCA transformation to bow features
```

Results shown for bow_features vs tfidf_features

This output shows the original class distribution in the training data for the TF-IDF model before SMOTE is applied. It reveals that the 'Good' reviews are the majority class (around 77%), while 'Bad' reviews are the minority (around 23%).

```
5 rows × 22044 columns
reviews.rating
Bad     0.5
Good    0.5
Name: proportion, dtype: float64
reviews.rating
Good    0.768875
Bad     0.231125
Name: proportion, dtype: float64
classification report and accuracy for bow_features:
              precision    recall  f1-score   support

         Bad       0.45      0.59      0.51       462
        Good       0.87      0.79      0.82      1538

    accuracy                           0.74      2000
   macro avg       0.66      0.69      0.67      2000
weighted avg       0.77      0.74      0.75      2000

Accuracy Score: 0.7415
classification report and accuracy for tfidf_features:
              precision    recall  f1-score   support

         Bad       0.59      0.67      0.63       462
        Good       0.90      0.86      0.88      1538

    accuracy                           0.82      2000
   macro avg       0.74      0.76      0.75      2000
weighted avg       0.83      0.82      0.82      2000
```

What is the deciding facts to choose TF-IDF model over BoW model ?

In summary, the results indicate that the TF-IDF model outperforms the BoW model in predicting hotel review sentiment. Its higher accuracy, precision, recall, and F1-scores suggest better performance. The predictions on new reviews further demonstrate the differences between the two models, and the feature importance analysis helps understand the factors driving the TF-IDF model's predictions.

```
Accuracy Score: 0.816
TF-IDF classifier prediction of test data
['Bad' 'Bad']
TF classifier prediction of test data
['Good' 'Good']

Top 10 Important Featoures of TF-IDF classifier
Variable: categori   Importance: 0.3733085519
Variable: pass       Importance: 0.0691510642
Variable: best       Importance: 0.0384961287
Variable: nice       Importance: 0.0346435611
Variable: left       Importance: 0.018177888
Variable: circul     Importance: 0.0162522587
Variable: chang      Importance: 0.0151940292
Variable: western    Importance: 0.0151002967
Variable: plu        Importance: 0.0148722641
Variable: item       Importance: 0.0142765293
```

Project Plan (Group)

Teng Teng

- **Improved Performance:** Hyperparameter tuning helps find the optimal settings for the decision tree, potentially improving its accuracy and generalization ability. By exploring different hyperparameter combinations, discover a model that better captures the underlying patterns in the data.
- **Different Algorithms:** experiment with other classification algorithms like Random Forest, Support Vector Machines (SVM), Tensorflow to see if they yield better results.

Why we use Tensor flow and no others as our deep learning exercise?
TensorFlow can handle both small and large-scale data, making it suitable for sentiment analysis tasks of any size.

tf.data.AUTOTUNE is a parameter in TensorFlow's tf.data API that automates the tuning of the dataset performance. When you set AUTOTUNE as the value for parameters like num_parallel_calls or prefetch, TensorFlow dynamically determines the optimal number of parallel calls and the prefetch buffer size, respectively, to improve the efficiency and performance of your data pipeline. This means you don't have to manually figure out the best settings, as TensorFlow does it for you!

We start by split the data into train and validation sets using tf.data.Dataset.take and tf.data.Dataset.skip

```
# Assuming you want 80% for training and 20% for validation
train_size = int(0.8 * len(train_ds))
val_size = int(0.2 * len(train_ds))

train_ds = train_ds.take(train_size) # This is your training dataset
val_ds = train_ds.skip(train_size).take(val_size) # This is your validation dataset


# Apply cache and prefetch
train_ds = train_ds.cache().prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

Each word in the vocabulary will be represented as a vector of 128 numbers. This allows the model to capture semantic relationships between words.This is the **embedding layer**. It's the heart of how the model understands words. It takes the numerical representations of words from the vectorize_layer and transforms them into dense vectors of dimension. Bidirectional means the LSTM processes the text in both forward and backward directions, which helps it capture context more effectively.uses a sigmoid activation function, which makes it suitable for binary classification problems (e.g., positive or negative sentiment). The output of this layer is a value between 0 and 1, representing the probability of the input text belonging to the positive class.

```
EMBEDDING_DIM=128

model = tf.keras.Sequential([
    vectorize_layer,
    tf.keras.layers.Embedding(input_dim=VOCAB_SIZE,
              output_dim=EMBEDDING_DIM,
              mask_zero=True,
              name='embedding'),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64)),
    tf.keras.layers.Dense(64),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Let start with 5 epo for our training set
Adam is a popular optimization algorithm that is generally efficient and effective for many types of neural networks.

- **Loss Function**: The loss function measures the difference between the model's predictions and the actual target values. The goal of training is to minimize this loss.
- This is a loss function specifically designed for **binary classification** problems (where the output is either 0 or 1). It calculates the cross-entropy between the predicted probabilities and the true labels.

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.BinaryCrossentropy(from_logits=False),
              metrics=['accuracy'])

model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=5)
```

```
Epoch 1/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 75s 283ms/step - accuracy: 0.7826 - loss: 0.4867
Epoch 2/5
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/epoch_iterator.py:151: UserWarning: Your input ra
  self._interrupted_warning()
250/250 ━━━━━━━━━━━━━━━━━━━━ 69s 276ms/step - accuracy: 0.9043 - loss: 0.2398
Epoch 3/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 69s 278ms/step - accuracy: 0.9455 - loss: 0.1422
Epoch 4/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 69s 275ms/step - accuracy: 0.9732 - loss: 0.0752
Epoch 5/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 69s 278ms/step - accuracy: 0.9738 - loss: 0.0639
<keras.src.callbacks.history.History at 0x7d2904a60e10>
```

Let put 3 results together - Tensor flow results shown the data is overfitting!!!

Final conclusion - the results in TF-IDF model outperforms 2 others models in predicting hotel review sentiment. Its higher accuracy, precision, recall, and F1-scores suggest better performance.

```
[21] from sklearn.metrics import classification_report

     print(classification_report(y_labels, y_preds))
```

```
              precision    recall  f1-score   support

         0.0       0.99      0.98      0.98      1847
         1.0       0.99      1.00      0.99      6153

    accuracy                           0.99      8000
   macro avg       0.99      0.99      0.99      8000
weighted avg       0.99      0.99      0.99      8000
```

5 rows × 22044 columns

```
reviews.rating
Bad     0.5
Good    0.5
Name: proportion, dtype: float64
reviews.rating
Good    0.768875
Bad     0.231125
Name: proportion, dtype: float64
classification report and accuracy for bow_features:
              precision    recall  f1-score   support

         Bad       0.45      0.59      0.51       462
        Good       0.87      0.79      0.82      1538

    accuracy                           0.74      2000
   macro avg       0.66      0.69      0.67      2000
weighted avg       0.77      0.74      0.75      2000

Accuracy Score: 0.7415
classification report and accuracy for tfidf_features:
              precision    recall  f1-score   support

         Bad       0.59      0.67      0.63       462
        Good       0.90      0.86      0.88      1538

    accuracy                           0.82      2000
   macro avg       0.74      0.76      0.75      2000
weighted avg       0.83      0.82      0.82      2000
```

Nanyang Polytechnic
Post-Diploma Certificate in Applied Data Science
ITD214 Applied Data Science Project
Final Project Presentation
26 Feb 2025

Group 4
Individual Presentation
Wong Shao Mun (Admission No. 1038987U)

Outline

1. Business Problem, Dataset and Data Cleaning (Group)
2. Model Design (Individual)
3. Model Assessment (Individual)
4. Evaluation and Recommendations (Group)

Clean Data

- Shape (10000, 26)
- Number of rows: 10000
- Number of columns: 26

Clean Data

- Data rather clean, fields of interest need not be dropped or imputed: 'reviews.date', 'reviews.rating' and 'reviews.text' (each with 10k rows)

```
Count number of rows with non-empty
values:
id                          10000
dateAdded                   10000
dateUpdated                 10000
address                     10000
categories                  10000
primaryCategories           10000
city                        10000
country                     10000
keys                        10000
latitude                    10000
longitude                   10000
name                        10000
postalCode                  10000
province                    10000
```

```
reviews.date                10000
reviews.dateAdded               0
reviews.dateSeen            10000
reviews.rating              10000
reviews.sourceURLs          10000
reviews.text                10000
reviews.title                9999
reviews.userCity            10000
reviews.userProvince         9998
reviews.username            10000
sourceURLs                  10000
websites                    10000
dtype: int64
```

Construct Data

- Extract year, month, day, weekofyear and day_of_week for time series analysis
- Apply one-hot encoding for day_of_week

```
# Extract year, month, day and weekofyear from 'reviews.date'.
df['year'] = df['reviews.date'].dt.year
df['month'] = df['reviews.date'].dt.month
df['day'] = df['reviews.date'].dt.day
df['weekofyear'] = df['reviews.date'].dt.isocalendar().week

# Extract day of the week (e.g., Mon, Tue, etc.)
df['day_of_week'] = df['reviews.date'].dt.strftime('%a')  # Short format (Mon, Tue)

# Apply one-hot encoding for df['day_of_week'] column.
df = pd.get_dummies(df, columns=['day_of_week'], dtype=int)

print(df.head())
```

Construct Data

```
                                           keys  latitude  ...  month  \
0           us/ca/goleta/5620callereal/-1127060008  34.44178  ...      1
1  us/ca/carmelbythesea/5thandsancarlospobox3574/...  36.55722  ...      4
2  us/ca/carmelbythesea/5thandsancarlospobox3574/...  36.55722  ...      1
3  us/ca/carmelbythesea/5thandsancarlospobox3574/...  36.55722  ...      8
4  us/ca/carmelbythesea/5thandsancarlospobox3574/...  36.55722  ...      3

   day  weekofyear  day_of_week_Fri  day_of_week_Mon  day_of_week_Sat  \
0    1           1                0                1                0
1    2          13                0                0                1
2    6           1                0                0                0
3   22          34                0                1                0
4   21          12                0                1                0

   day_of_week_Sun  day_of_week_Thu  day_of_week_Tue  day_of_week_Wed
0                0                0                0                0
1                0                0                0                0
2                0                0                0                1
3                0                0                0                0
4                0                0                0                0

[5 rows x 37 columns]
```

Construct Data

- 'reviews.rating' = 1, 2, 3, 4 or 5
- Negative sentiment: 1 or 2 and positive sentiment: 4 or 5
- Need check 'reviews.rating' = 3 is whether negative or positive sentiment

```
# Filter for rows where 'reviews.rating' is equal to 3
filtered_df = df[df['reviews.rating'] == 3]

# Select the desired columns and print the first 20 rows
print(filtered_df[['reviews.rating', 'reviews.text']].head(20))
```

Construct Data

- Output shows 'reviews.rating' = 3 is for negative sentiment

```
     reviews.rating                                       reviews.text
0                 3  This hotel was nice and quiet. Did not know, t...
2                 3  Parking was horrible, somebody ran into my ren...
11                3  I stayed here for three nights while I explore...
13                3  The water is very hot and there's no cold wate...
18                3  The Whitney Hotel is ideally located to see mo...
45                3  The bar closed at 10pm which was poorOnly 3 da...
86                3  We stayed here after traveling through Rocky M...
89                3                                           MoreMore
98                3  The issues started the first night. Do Not sta...
105               3  If you are driving then this is for you! A bit...
106               3  First impression, I see vehicles parked tightl...
113               3  The hotel staff was friendly and engaging. The...
115               3  I had a friend in town, so was looking for a S...
128               3  This hotel is in a great location and the room...
132               3  super friendly staff, average breakfast, free ...
144               3  Nice place, but.........21 to park!Wifi would ...
160               3  Average place , over priced and they charge yo...
175               3  I chose this hotel because it was close to the...
179               3  Although the Best Western is aesthetically ple...
183               3  we checked in around 8pm room air conditioner ...
```

Construct Data

- Therefore will map 'reviews.rating' = 1, 2 and 3 as negative sentiment and 'reviews.rating' = 4 and 5 as positive sentiment

```
# Add df['sentiment'] where df['sentiment']=1 where df['reviews.rating']==4, 5 and df['sentiment']=0 where df['reviews.rating']==1, 2, 3.

# Create the 'sentiment' column based on 'reviews.rating'
df.loc[df['reviews.rating'].isin([1, 2, 3]), 'sentiment'] = 0 # 0 for negative sentiments.
df.loc[df['reviews.rating'].isin([4, 5]), 'sentiment'] = 1 # 1 for positive sentiment.

# Convert to integer
df['sentiment'] = df['sentiment'].astype(int)
```

Construct Data

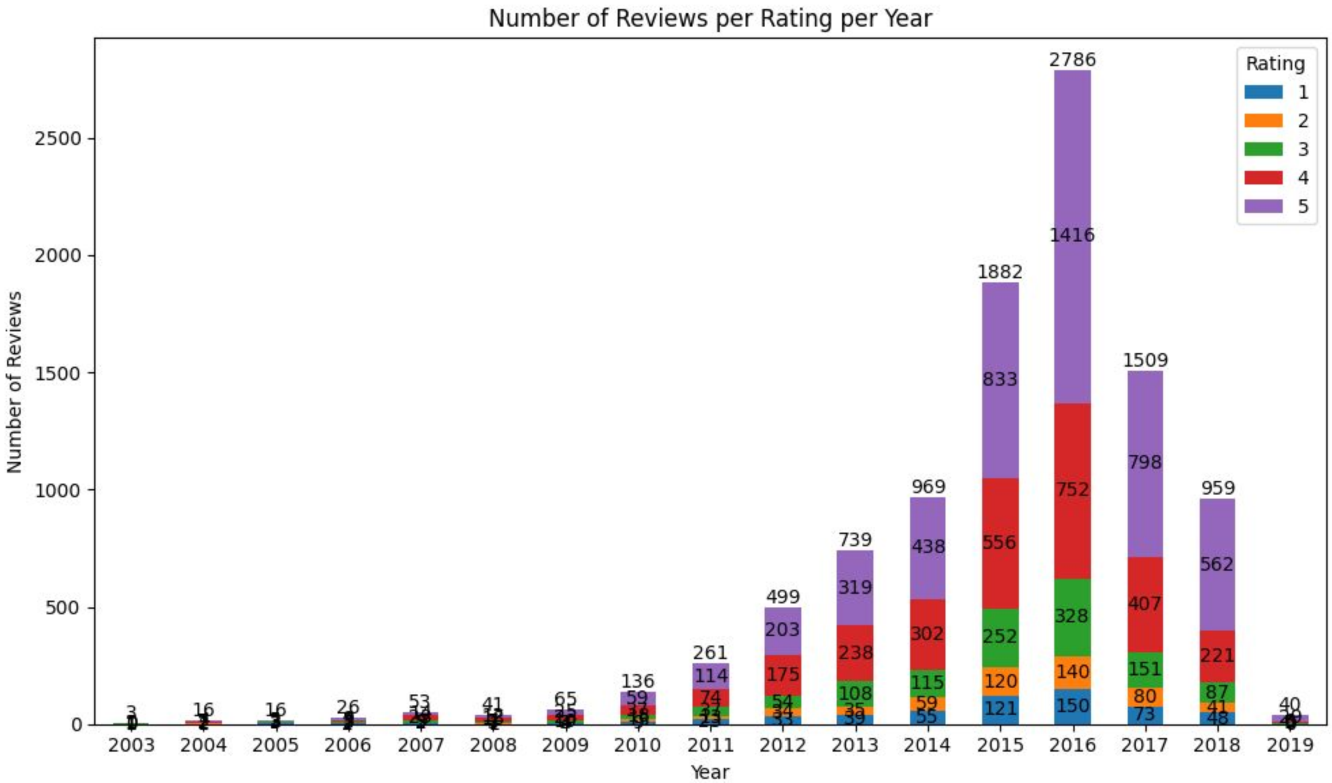
- Negative sentiment is around 23% while positive sentiment is around 77%

sentiment	reviews.rating	count
0	1	567
	2	554
	3	1190
1	4	2849
	5	4840

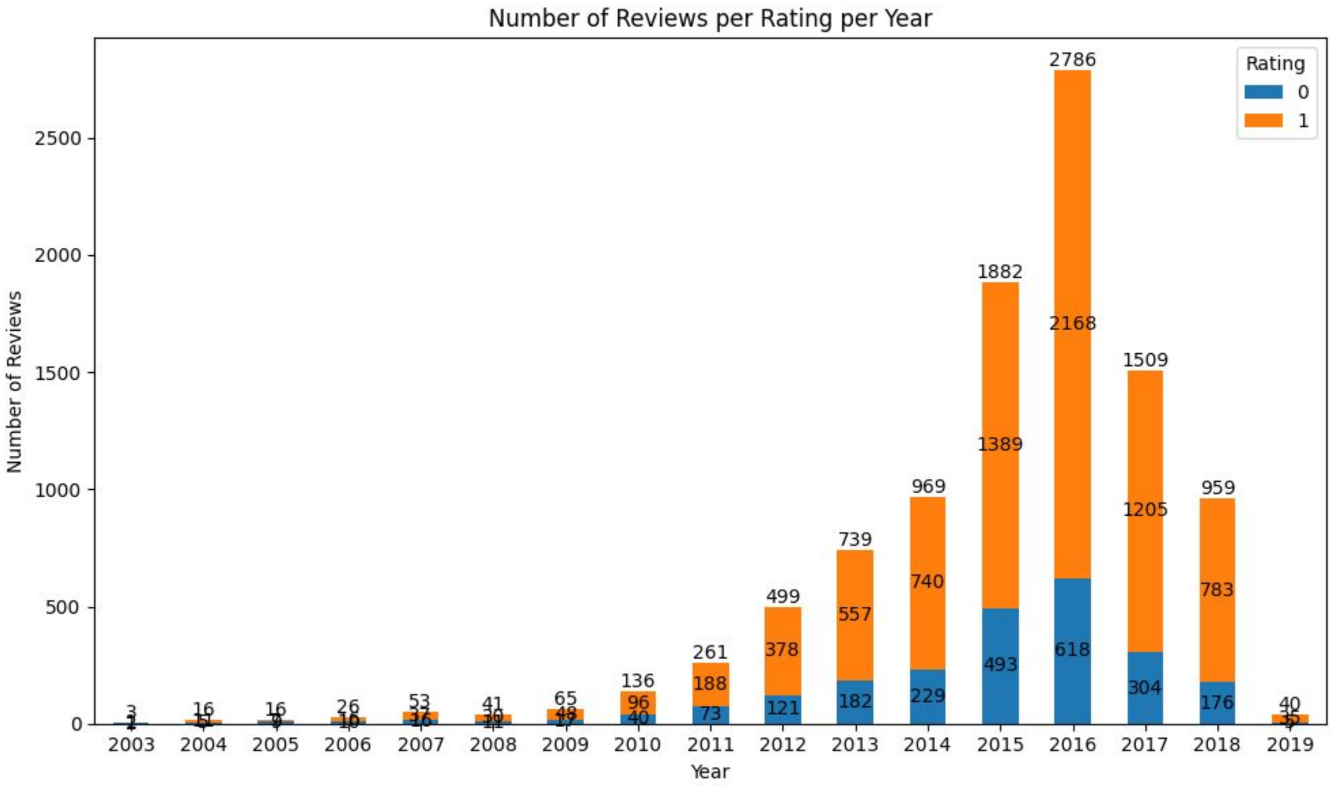
dtype: int64

	sentiment	count	percentage
1	0	2311	23.11
0	1	7689	76.89

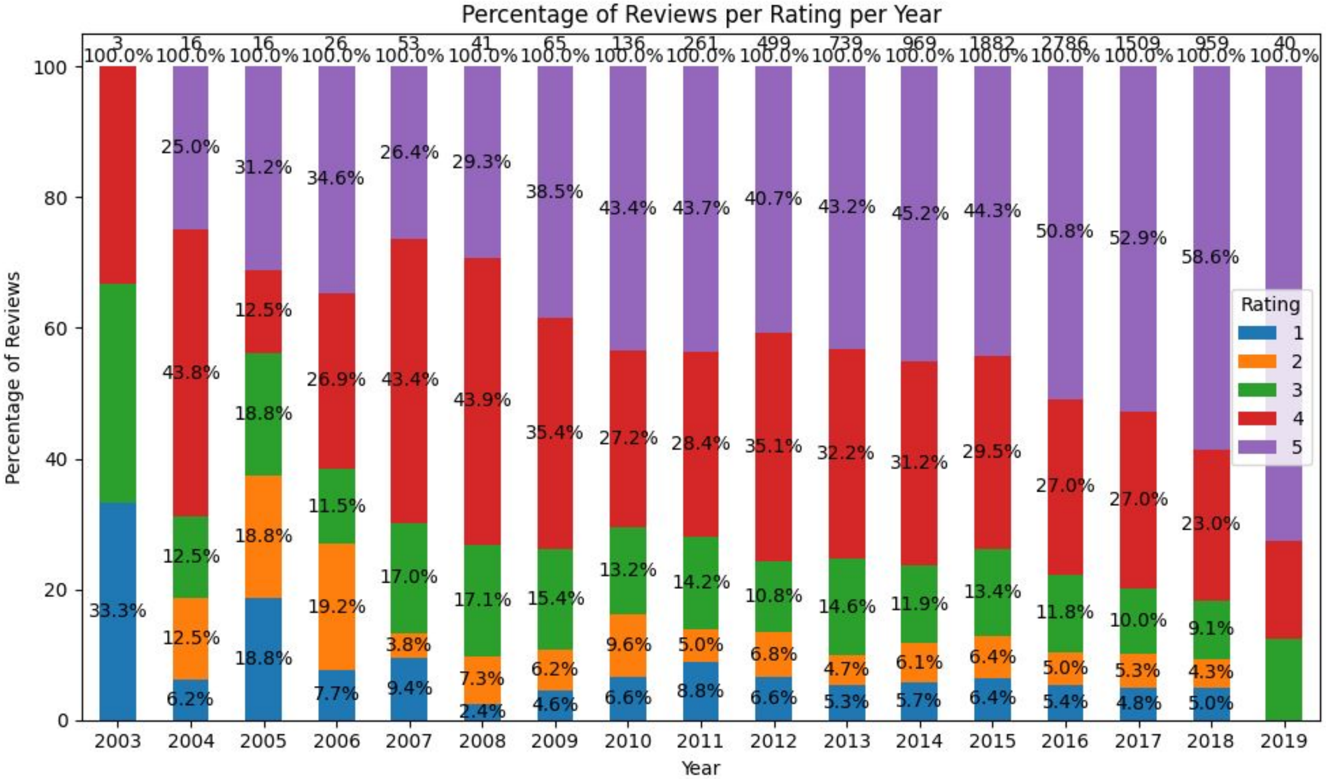
Construct Data (Before Merging Target Variable Categories)



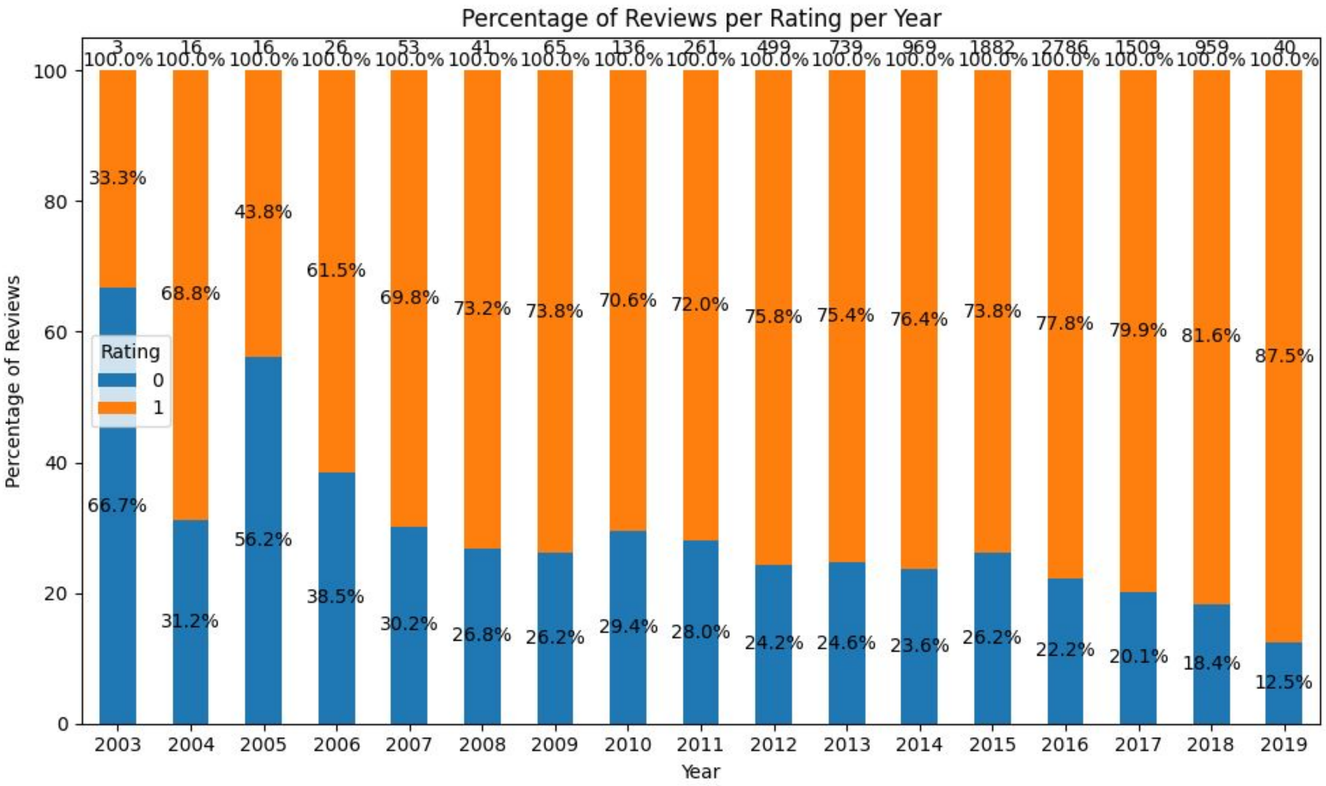
Construct Data (After Merging Target Variable Categories)



Construct Data (Before Merging Target Variable Categories)



Construct Data (After Merging Target Variable Categories)



Integrate Data

- Declare time-related variables as features and sentiment as target variable

```
# Declare features, X with columns: year, month, day, weekofyear,
day_of_week_Mon, day_of_week_Tue, day_of_week_Wed, day_of_week_Thu,
day_of_week_Fri, day_of_week_Sat, day_of_week_Sun.
x = df[['year', 'month', 'day', 'weekofyear', 'day_of_week_Mon',
'day_of_week_Tue', 'day_of_week_Wed',
        'day_of_week_Thu', 'day_of_week_Fri', 'day_of_week_Sat',
'day_of_week_Sun']]

# Declare target variable, Y with column: sentiment.
y = df['sentiment']
```

Format Data

- Convert 'reviews.date' to datetime data type

```
# Convert 'reviews.date' column to datetime objects, specifying the correct format
df['reviews.date'] = pd.to_datetime(df['reviews.date'], format='ISO8601') # alternative for ISO8601 format
```

Format Data

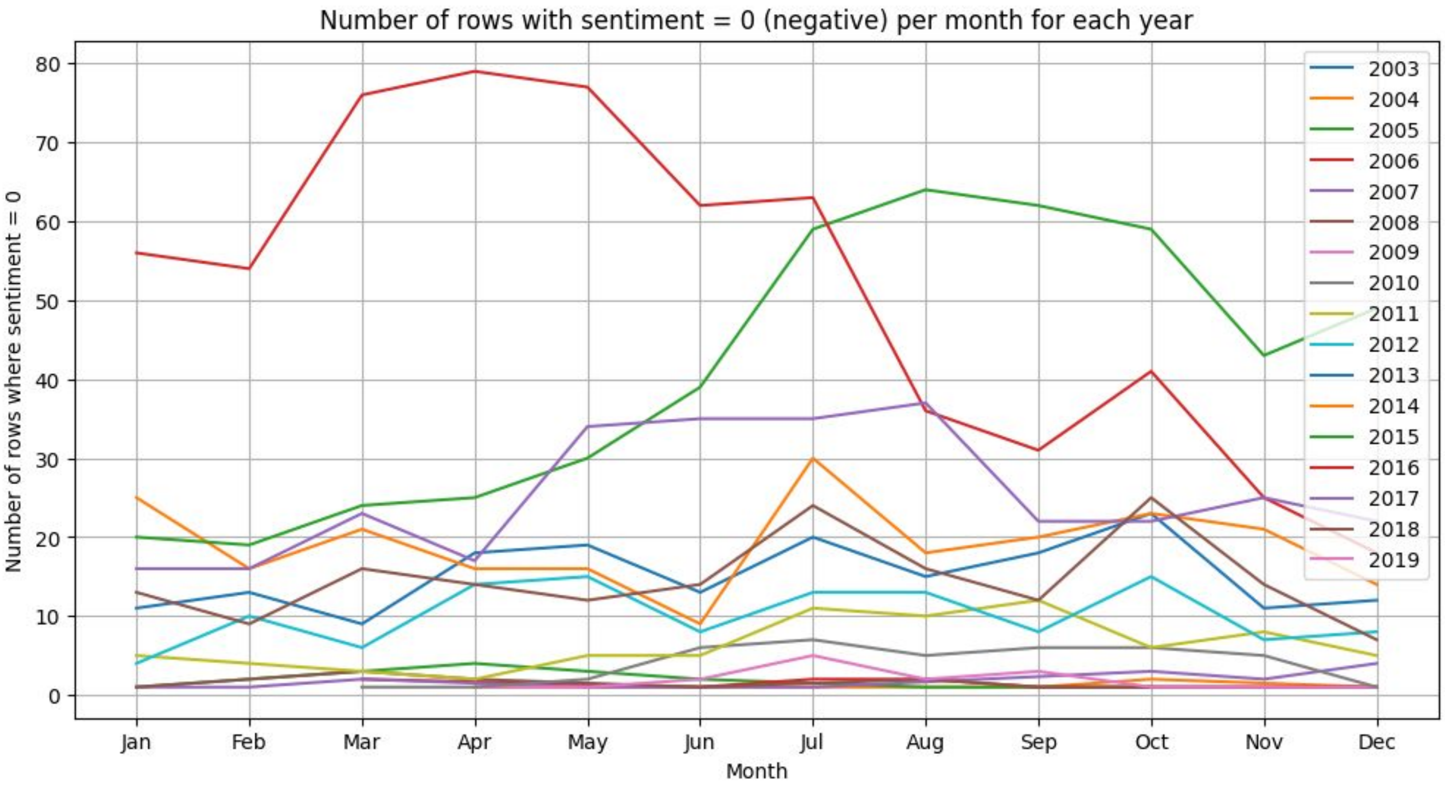
- Data type of 'reviews.date' changed from object to datetime

```
df.dtypes:
id                            object
dateAdded                     object
dateUpdated                   object
address                       object
categories                    object
primaryCategories             object
city                          object
country                       object
keys                          object
latitude                     float64
longitude                    float64
name                          object
postalCode                    object
province                      object
```

```
reviews.date      datetime64[ns, UTC]
reviews.dateAdded             float64
reviews.dateSeen               object
reviews.rating                  int64
reviews.sourceURLs             object
reviews.text                   object
reviews.title                  object
reviews.userCity               object
reviews.userProvince           object
reviews.username               object
sourceURLs                     object
websites                       object
dtype: object
```

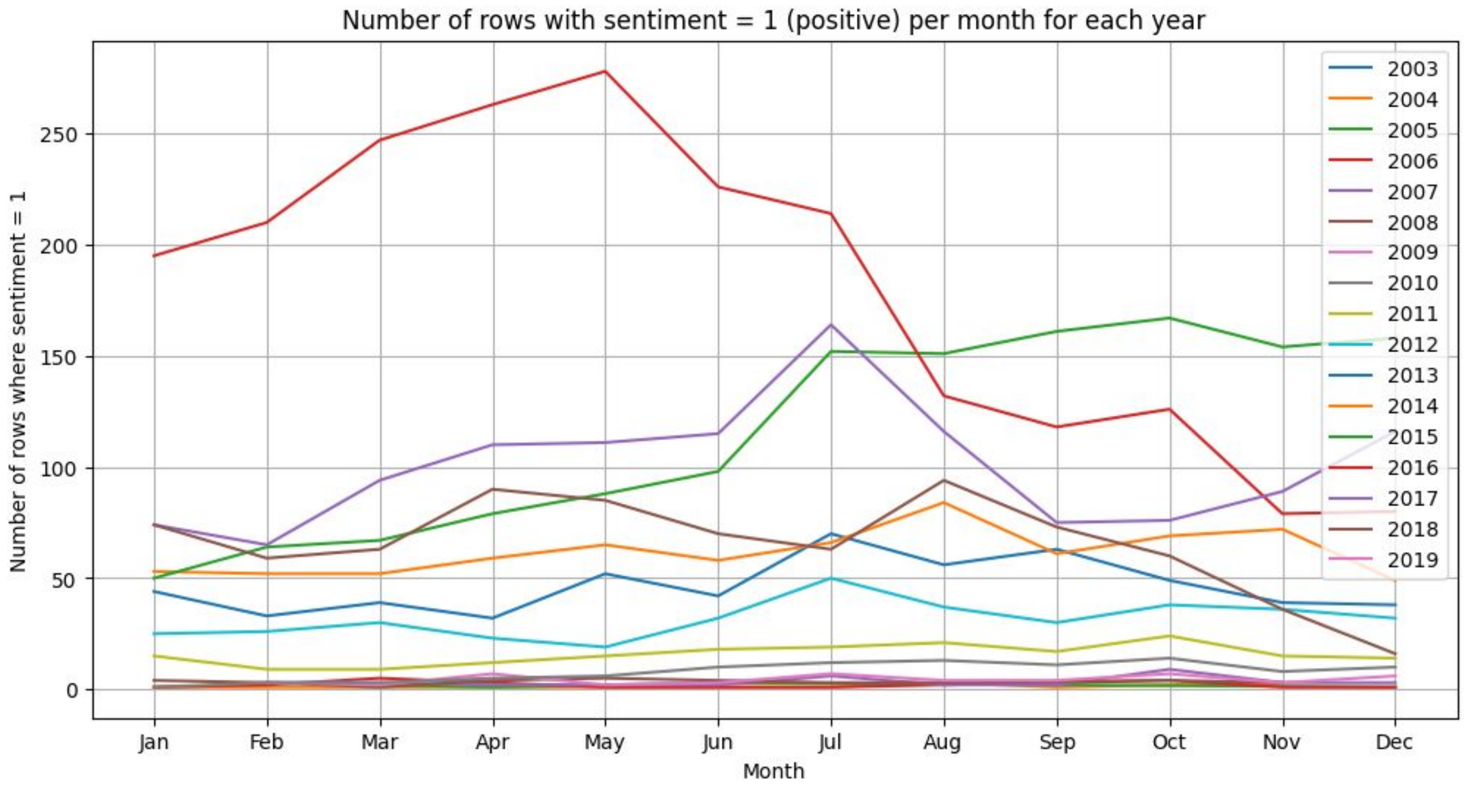
Project Plan (Group)

- 2015 and 2016 data seems out of place



Project Plan (Group)

- 2015 and 2016 data seems out of place

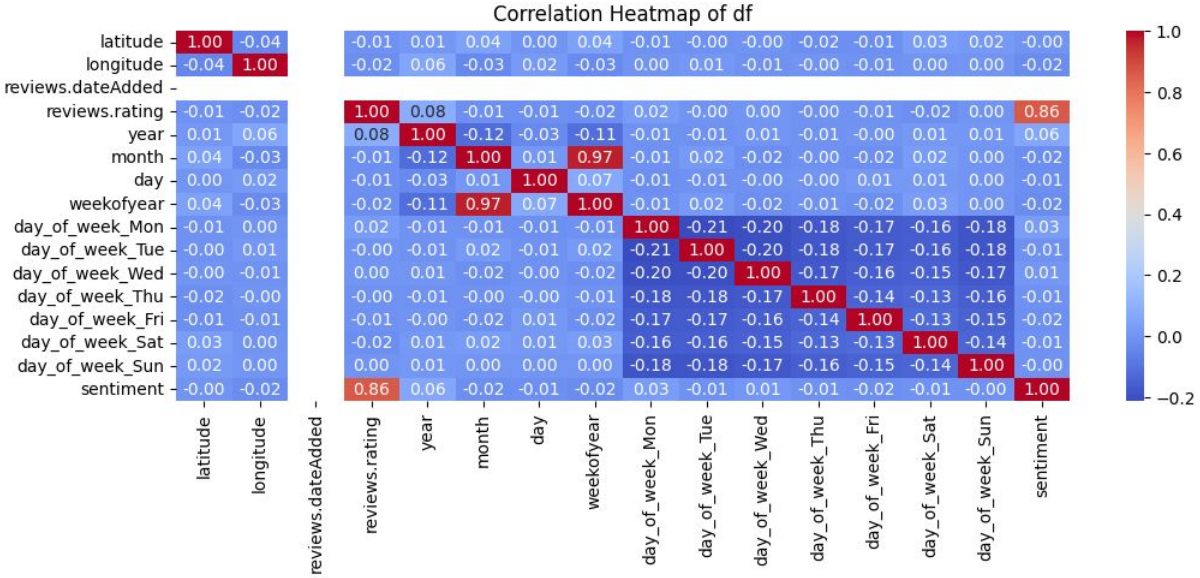


Project Plan (Group)

- Shao Mun
  - Fine-tune hyperparameters of models [KNN, SVM, logistic regression, decision tree, Gaussian Naive Bayes (NB), random forest and gradient boosting] used
  - Upon consultation with tutor, Ms Joanna Foo, <u>do not drop years</u> for time series analysis

Model Design

- Feature Selection
  - Correlation analysis
    - Heatmap shows sentiment to be highly positively correlated to reviews.rating which is expected as sentiment is a derived variable of reviews.rating



Model Design

- Feature Selection
  - Any feature that does not require any further grouping
  - Potential candidate: ‘primaryCategories’ (4 categories)
  - Not selected because highly imbalanced data

primaryCategories	count
Accommodation & Food Services	9762
Accommodation & Food Services,Arts Entertainment & Recreation	7
Accommodation & Food Services,Administrative & Support & Waste Management & Remediation	1
Accommodation & Food Services,Agriculture	1

dtype: int64

Model Design

- Feature Selection
  - Features: Time-related variables
  - Target variable: ‘sentiment’ where 0 is negative sentiment and 1 is positive sentiment

```
# Declare features, X with columns: year, month, day, weekofyear, day_of_week_Mon, day_of_week_Tue,
# day_of_week_Wed, day_of_week_Thu, day_of_week_Fri, day_of_week_Sat, day_of_week_Sun.
x = df[['year', 'month', 'day', 'weekofyear', 'day_of_week_Mon', 'day_of_week_Tue', 'day_of_week_Wed',
        'day_of_week_Thu', 'day_of_week_Fri', 'day_of_week_Sat', 'day_of_week_Sun']]

print("x.shape:", x.shape)
print("x:")
print(x)
print()

# Declare target variable, Y with column: sentiment.
y = df['sentiment']

print("y.shape:", y.shape)
print("y:")
print(y)
print()
```

Model Design

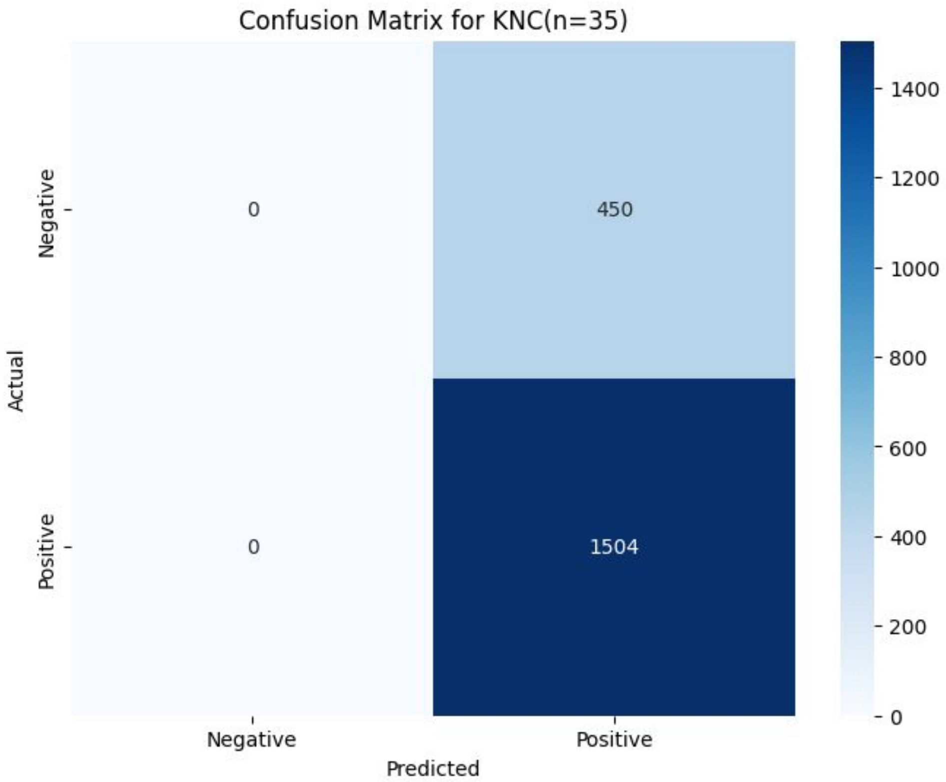
- Models Ran
  - K-Nearest Neighbors: n_neighbors tuned from 5 to 50 in stepsize of 5
  - Support Vector Machine (SVM)
  - Logistic Regression
  - Decision Trees: min_samples_split tuned as 2, 10, 20, 30
  - Gaussian Naive Bayes
  - Random Forest: n_estimators tuned from 5 to 30 in stepsize of 5
  - Gradient Boosting
- Clarification for Ms Lim Ai Huey's question on Wed, 26 Feb 2025
  - Joint examination by Ms Lim Ai Huey and Mr Kee Li-ren
  - Ms Lim to Shao Mun: Model using linear regression, neural network or generalised linear model (GLM) as SVM not suitable
  - Reply to Ms Lim: The target variable was coded categorical variable, 0 for negative sentiment and 1 for positive sentiment. That is why models like SVM were used

Model Assessment

- Performance
  - Metrics: Mean CV Accuracy, Train Accuracy, Test Accuracy, Train Precision, Test Precision
  - Highest accuracies came from KNC models
  - But model cannot predict negative sentiment

	Model	Cross-Validation Accuracy Scores	Mean CV Accuracy	Train Accuracy	Test Accuracy	Train Precision	Test Precision
6	KNC(n=35)	[0.7693, 0.7692, 0.7692, 0.7697, 0.7697]	0.7694	0.7693	0.7697	0.5919	0.5924
7	KNC(n=40)	[0.7693, 0.7692, 0.7692, 0.7697, 0.7697]	0.7694	0.7693	0.7697	0.5919	0.5924
8	KNC(n=45)	[0.7693, 0.7692, 0.7692, 0.7697, 0.7697]	0.7694	0.7693	0.7697	0.5919	0.5924
9	KNC(n=50)	[0.7693, 0.7692, 0.7692, 0.7697, 0.7697]	0.7694	0.7693	0.7697	0.5919	0.5924
11	LR	[0.7693, 0.7692, 0.7692, 0.7697, 0.7697]	0.7694	0.7693	0.7697	0.5919	0.5924
5	KNC(n=30)	[0.7693, 0.7677, 0.7692, 0.7697, 0.7692]	0.7690	0.7696	0.7692	0.7459	0.6693
23	GB	[0.7683, 0.7687, 0.7697, 0.7692, 0.7677]	0.7687	0.7720	0.7677	0.8241	0.6307
10	SVM	[0.7693, 0.7687, 0.7666, 0.7687, 0.7677]	0.7682	0.7731	0.7677	0.8174	0.5921
4	KNC(n=25)	[0.7688, 0.7682, 0.7692, 0.7692, 0.7671]	0.7685	0.7699	0.7671	0.7462	0.6251
16	GNB	[0.7708, 0.761, 0.7666, 0.7636, 0.7631]	0.7650	0.7660	0.7631	0.6793	0.6632
2	KNC(n=15)	[0.7606, 0.761, 0.7584, 0.7646, 0.761]	0.7611	0.7723	0.7610	0.7400	0.6444
3	KNC(n=20)	[0.7662, 0.762, 0.7651, 0.7656, 0.761]	0.7640	0.7701	0.7610	0.7176	0.6286
1	KNC(n=10)	[0.7335, 0.7462, 0.7349, 0.739, 0.7303]	0.7368	0.7717	0.7303	0.7259	0.6373
15	DT(min_samples_samples split=30)	[0.7212, 0.7124, 0.7134, 0.7323, 0.7149]	0.7189	0.7812	0.7149	0.7473	0.6345
0	KNC(n=5)	[0.7182, 0.7329, 0.7247, 0.718, 0.7134]	0.7214	0.7818	0.7134	0.7487	0.6383
19	RF(n=15)	[0.7074, 0.7032, 0.7134, 0.6929, 0.7042]	0.7042	0.8178	0.7042	0.8055	0.6402
21	RF(n=25)	[0.7033, 0.7108, 0.7057, 0.7021, 0.7011]	0.7046	0.8205	0.7011	0.8084	0.6407
14	DT(min_samples_samples split=20)	[0.6997, 0.6965, 0.7032, 0.7011, 0.7006]	0.7002	0.7899	0.7006	0.7628	0.6363
20	RF(n=20)	[0.7084, 0.7088, 0.7073, 0.697, 0.6996]	0.7042	0.8203	0.6996	0.8082	0.6382
22	RF(n=30)	[0.7049, 0.7062, 0.7062, 0.7042, 0.6981]	0.7039	0.8208	0.6981	0.8094	0.6364
18	RF(n=10)	[0.7049, 0.7052, 0.7057, 0.6909, 0.6899]	0.6993	0.8157	0.6899	0.8009	0.6357
17	RF(n=5)	[0.7003, 0.7001, 0.6868, 0.6868, 0.6791]	0.6906	0.8072	0.6791	0.7884	0.6315
13	DT(min_samples_samples split=10)	[0.6706, 0.6694, 0.6847, 0.674, 0.6709]	0.6739	0.8052	0.6709	0.7858	0.6370
12	DT(min_samples_split=2)	[0.6645, 0.651, 0.6699, 0.6597, 0.6535]	0.6597	0.8214	0.6535	0.8076	0.6368

Model Assessment



Model Design

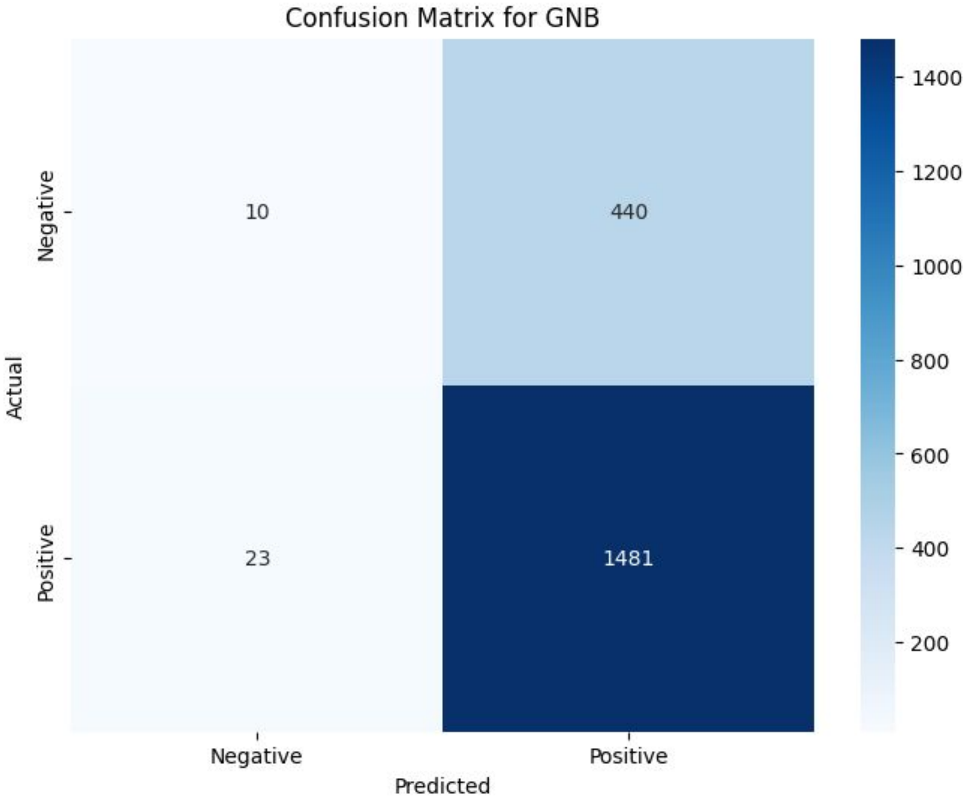
- Models Ran
  - Tried undersampling of the majority class (sentiment=1) in the training set
    - K-Nearest Neighbors: n_neighbors tuned from 5 to 50 in stepsize of 5
    - Support Vector Machine (SVM)
    - Logistic Regression
    - Decision Trees: min_samples_split tuned as 2, 10, 20, 30
    - Gaussian Naive Bayes
    - Random Forest: n_estimators tuned from 5 to 30 in stepsize of 5
    - Gradient Boosting

Model Assessment

- Performance
  - Metrics: Mean CV Accuracy, Train Accuracy, Test Accuracy, Train Precision, Test Precision
  - Highest accuracy came from GNB model
  - Accuracies worsened from no sampling to undersampling of the majority class (sentiment=1) in the training set
  - Model can predict negative sentiment but not very well

	Model	Cross-Validation Accuracy Scores	Mean CV Accuracy	Train Accuracy	Test Accuracy	Train Precision	Test Precision
16	GNB	[0.5038, 0.4964, 0.5287, 0.5604, 0.5952]	0.5369	0.5300	0.5952	0.5340	0.6554
11	LR	[0.4987, 0.5113, 0.5031, 0.5353, 0.5921]	0.5281	0.5158	0.5921	0.5178	0.6556
23	GB	[0.5402, 0.5138, 0.5092, 0.5281, 0.5302]	0.5243	0.6154	0.5302	0.6155	0.6748
10	SVM	[0.4844, 0.5133, 0.5271, 0.5087, 0.5123]	0.5092	0.6692	0.5123	0.6694	0.6489
6	KNC(n=35)	[0.4844, 0.5184, 0.5092, 0.5123, 0.5046]	0.5058	0.5799	0.5046	0.5799	0.6543
19	RF(n=15)	[0.5079, 0.4995, 0.4964, 0.4933, 0.5041]	0.5003	0.7879	0.5041	0.7881	0.6529
8	KNC(n=45)	[0.4997, 0.5082, 0.5159, 0.5138, 0.5041]	0.5083	0.5588	0.5041	0.5588	0.6569
4	KNC(n=25)	[0.5013, 0.501, 0.4872, 0.5061, 0.5031]	0.4997	0.5874	0.5031	0.5874	0.6547
17	RF(n=5)	[0.5095, 0.5087, 0.4826, 0.4882, 0.501]	0.4980	0.7673	0.5010	0.7674	0.6532
21	RF(n=25)	[0.5028, 0.4939, 0.4923, 0.4949, 0.4995]	0.4967	0.7915	0.4995	0.7916	0.6508
18	RF(n=10)	[0.5095, 0.4908, 0.4765, 0.5005, 0.4995]	0.4953	0.7823	0.4995	0.7824	0.6525
20	RF(n=20)	[0.5054, 0.499, 0.4903, 0.4893, 0.4974]	0.4963	0.7901	0.4974	0.7902	0.6482
22	RF(n=30)	[0.5084, 0.4995, 0.4898, 0.5015, 0.4949]	0.4988	0.7920	0.4949	0.7921	0.6453
13	DT(min_samples_samples split=10)	[0.4721, 0.4754, 0.4601, 0.4729, 0.4913]	0.4744	0.7460	0.4913	0.7474	0.6545
2	KNC(n=15)	[0.4813, 0.4903, 0.499, 0.5041, 0.4893]	0.4928	0.6129	0.4893	0.6133	0.6444
14	DT(min_samples_samples split=20)	[0.5033, 0.4708, 0.498, 0.4887, 0.4841]	0.4890	0.7019	0.4841	0.7030	0.6500
9	KNC(n=50)	[0.4701, 0.4821, 0.4882, 0.4877, 0.4836]	0.4823	0.5696	0.4836	0.5708	0.6624
0	KNC(n=5)	[0.4813, 0.4785, 0.477, 0.4667, 0.4688]	0.4745	0.6728	0.4688	0.6759	0.6478
12	DT(min_samples_split=2)	[0.4706, 0.4591, 0.4396, 0.4493, 0.4667]	0.4571	0.7926	0.4667	0.7998	0.6523
7	KNC(n=40)	[0.4757, 0.4693, 0.4857, 0.4944, 0.4632]	0.4776	0.5666	0.4632	0.5685	0.6556
15	DT(min_samples_samples split=30)	[0.4936, 0.4678, 0.4995, 0.4708, 0.4621]	0.4788	0.6778	0.4621	0.6799	0.6392
5	KNC(n=30)	[0.4476, 0.4754, 0.4458, 0.4662, 0.458]	0.4586	0.5793	0.4580	0.5832	0.6582
3	KNC(n=20)	[0.4373, 0.4591, 0.4406, 0.4514, 0.4529]	0.4483	0.5968	0.4529	0.6006	0.6491
1	KNC(n=10)	[0.4143, 0.4324, 0.3951, 0.4335, 0.4268]	0.4204	0.6278	0.4268	0.6434	0.6524

Model Assessment



Model Design

- Previous models ran could not predict negative sentiment well
- Proceed to try AutoRegressive Integrated Moving Average (ARIMA) time series forecasting model
    - df3_numeric = df3.select_dtypes(include=['number']).groupby(df3['date_yyyy_mm_dd']).mean()
    - sentiment values of 0 and 1 were grouped by time period (date/week/month) and mean taken for aggregated time period
    - Implication: Prediction from 0 to < 0.5: negative sentiment while prediction from 0.5 to 1: positive sentiment

Model Design

- Need check whether time series data is stationary
- Check reveals time series data is stationary

```
[82] # Augmented Dickey-Fuller (ADF) test. The null hypothesis of the ADF test is that the series is non-stationary.
     from statsmodels.tsa.stattools import adfuller

     # Perform ADF test on the 'sentiment' column (you can replace with your target variable)
     result = adfuller(df3['sentiment'])

     # Print the results
     print("ADF Statistic:", result[0])
     print("p-value:", result[1])
     print("Critical Values:", result[4])
```

```
ADF Statistic: -12.111143448908674
p-value: 1.9181150644839028e-22
Critical Values: {'1%': -3.4310214251582605, '5%': -2.8618367291146485, '10%': -2.56692794378353}
```

Augmented Dickey-Fuller (ADF) Test Result Analysis

1. The null hypothesis of the ADF test is that the series is non-stationary.
2. Since the p-value is significantly less than 0.05 (in fact, it's very close to 0), you can reject the null hypothesis and conclude that the data is stationary.
3. The ADF Statistic (-12.11) is much smaller than the critical values at the 1%, 5%, and 10% levels (e.g., -3.43 at the 1% level). This further confirms that the data does not have a unit root and is indeed stationary.
4. ARIMA modelling needs time series to be stationary. Since the ADF test shows that the series is stationary, therefore can proceed to do ARIMA modelling.

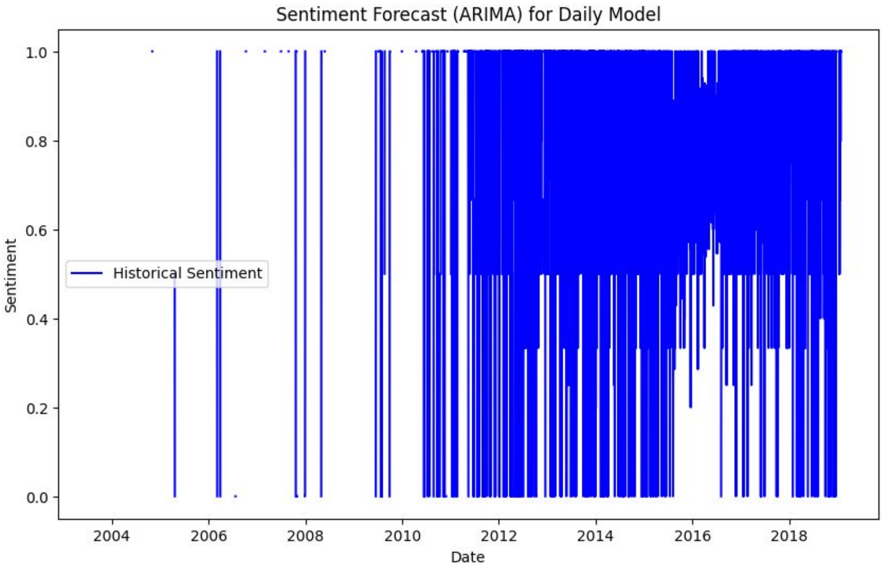
Model Assessment

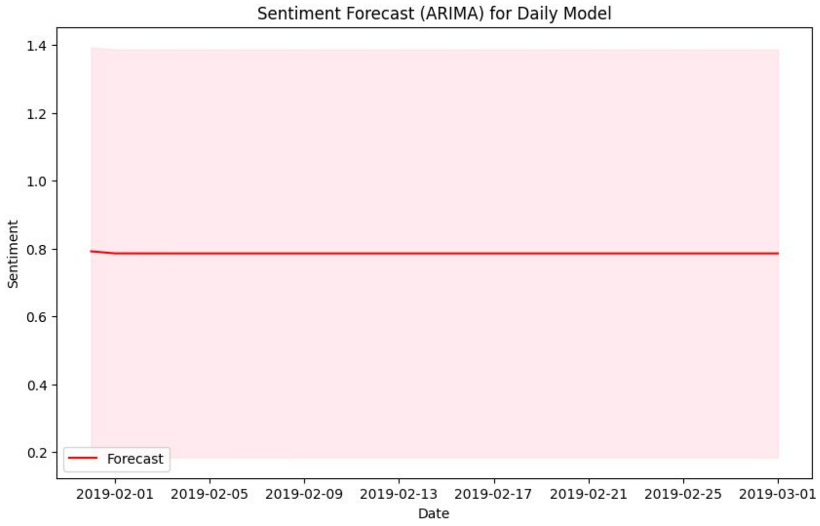
	model	mae	mse	rmse
daily	<statsmodels.tsa.arima.model.ARIMAResultsWrapp...	0.287963	0.119532	0.345734
weekly	<statsmodels.tsa.arima.model.ARIMAResultsWrapp...	0.195103	0.073571	0.271240
monthly	<statsmodels.tsa.arima.model.ARIMAResultsWrapp...	0.163335	0.064178	0.253334

- Performance
    - Metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE)
    - Results were close to one another
        - Monthly model most accurate
        - Daily model the worst
            - More sensitive to short-term variations so less stable and harder to use for long-term forecasting
    - Rather say which model is the best, better to view each model serves its own purpose
        - Daily model helps with daily analysis of sentiment spike
        - Weekly model can help to smooth out the daily fluctuations
        - Monthly model gives a broader picture of sentiment trends which is useful for business or marketing strategies

Model Assessment

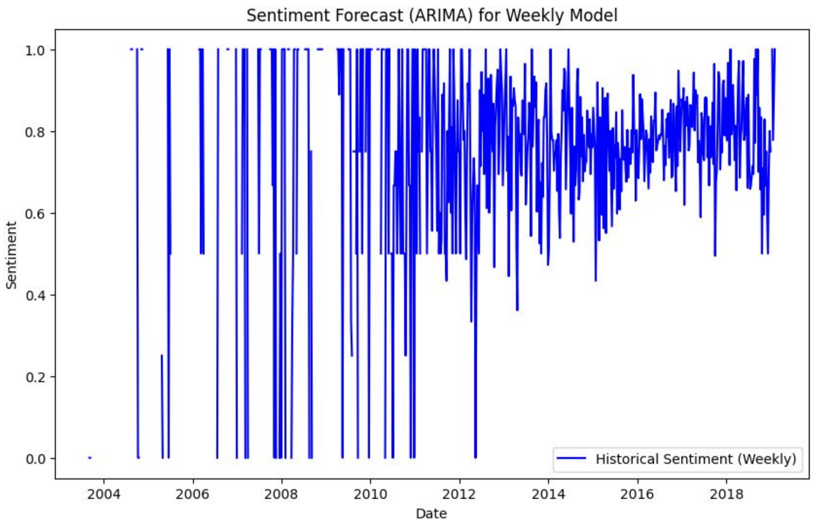
Daily model shows 30-day forecast of sentiment = 0.8 (80% likely positive)

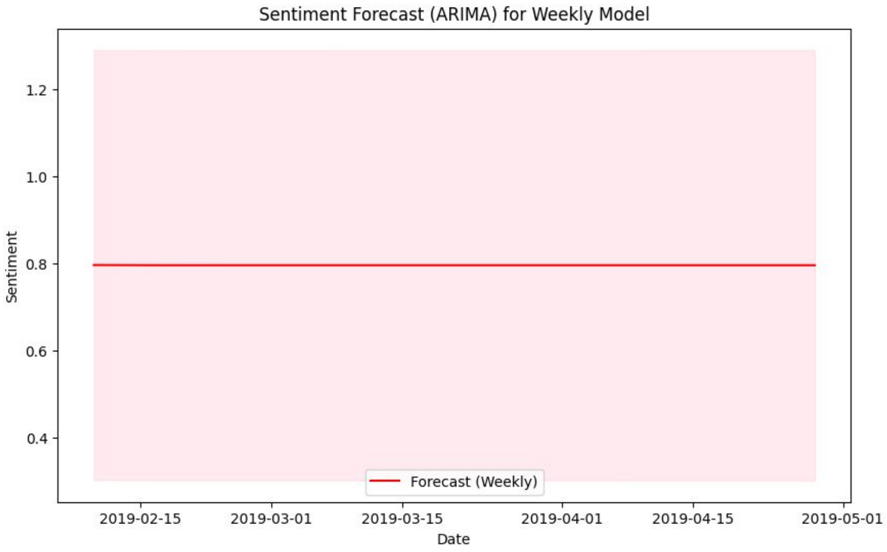




Model Assessment

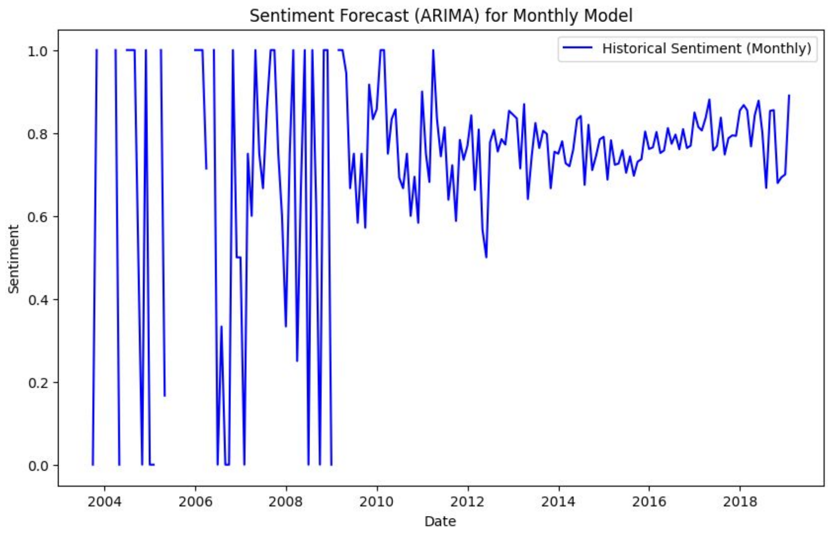
Weekly model shows 12-week forecast of sentiment = 0.8 (80% likely positive)

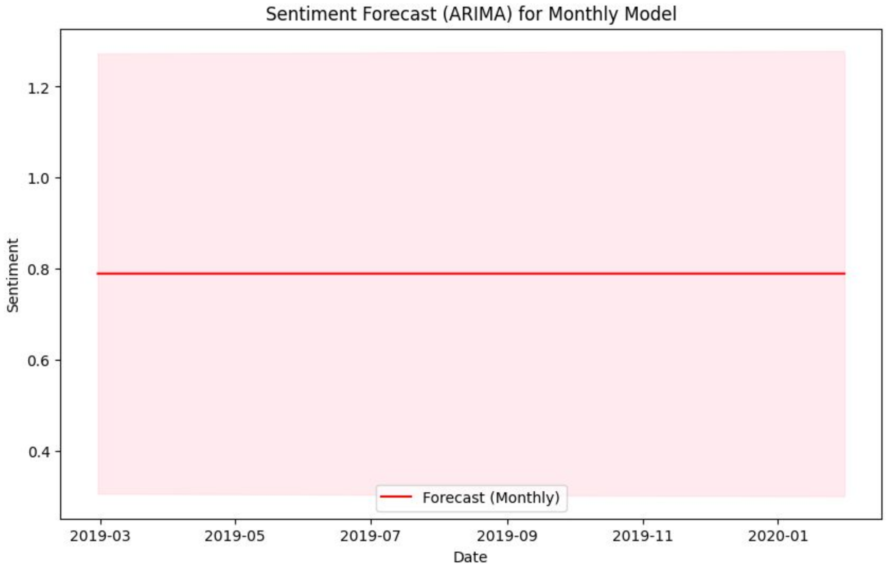




Model Assessment

Monthly model shows 12-month forecast of sentiment = 0.8 (80% likely positive)





Evaluation and Recommendations

- All 3 models (daily, weekly, monthly) shows forecast of sentiment = 0.8 (80% likely positive)
- Means a likely 20% chance of getting negative sentiment
- Hotel can reschedule existing staff or hire more staff to have 20% more man hours to handle potential negative sentiment

Evaluation and Recommendations

Business Proposal: Enhancing Guest Experience and Revenue

Executive Summary:

This proposal outlines a data-driven strategy to enhance guest experiences and increase revenue for the hotel by leveraging insights gained from topic modeling, sentiment analysis, and time series analysis of hotel reviews and operational data. By understanding guest preferences, addressing concerns, and optimizing operations based on seasonal trends, we can achieve significant improvements in guest satisfaction and profitability.

Evaluation and Recommendations

1. Understanding Guest Preferences through Topic Modeling:

- **Analysis:** We employed topic modeling to identify recurring themes and topics within guest reviews. This analysis revealed key areas of interest for our guests, including frequently mentioned topics, e.g. "room", "clean", "staff".
- **Recommendations:**
  - **Targeted Marketing:** Develop marketing campaigns that highlight the specific aspects of the hotel that resonate most with guests, such as hotel room cleanliness
  - **Service Enhancement:** Focus on improving services and amenities that are frequently mentioned in positive reviews, such as hotel staff services
  - **Addressing Concerns:** Identify and address negative topics or concerns raised by guests, such as parking spaces

Evaluation and Recommendations

2. Enhancing Guest Satisfaction with Sentiment Analysis:

- **Analysis:** The TF-IDF model before SMOTE is applied. It reveals that the 'Good' reviews are the majority class (around 77%), while 'Bad' reviews are the minority (around 23%). This analysis helped us understand the positive and negative aspects of guest experiences.
- **Recommendations:**
    - **Proactive Service Recovery:** Implement a system to identify and address negative reviews in real-time, offering solutions and demonstrating a commitment to guest satisfaction.
    - **Staff Training:** Train staff to address common guest concerns example room cleanliness and hotel staff service.It provide exceptional service in areas identified as needing improvement.
    - **Personalized Experiences:** Leverage sentiment analysis to personalize guest interactions, offering tailored recommendations and amenities based on their preferences - Use a customer journey map template to help create each persona.

Evaluation and Recommendations

3. Optimizing Operations with Time Series Analysis:

- **Analysis:** All 3 models (daily, weekly, monthly) shows forecast of sentiment = 0.8 (80% likely positive)
- Means a likely 20% chance of getting negative sentiment
- **Recommendation:**
    - Staffing Optimization: Hotel can reschedule existing staff or hire more staff to have 20% more man hours to handle potential negative sentiment

Evaluation and Recommendations

Overall recommendations:

1. From topic modelling
    a. Positive topics: Customer service, charm of location, affordability, cleanliness, smooth check-in, comfort, breakfast, parking
    b. Negative topics: Dining experience, cleanliness, mattress/bed, noise, microwave, smoke, front desk, room
    c. Focus on these **common aspects** can help drive positive sentiment up and reduce negative sentiment **at the same time**:
        i. Customer service: Check-in, front desk experience
        ii. Room: Bed, mattress, comfort
        iii. Ambience: Cleanliness, smoke
        iv. Food: Breakfast, dining experience

Evaluation and Recommendations

4. Measuring Success:

- **Key Performance Indicators (KPIs):**
  - **Guest Satisfaction Scores:** Monitor online reviews and guest surveys to track improvements in overall satisfaction.
  - **Revenue Growth:** Measure the impact of implemented strategies on revenue generation and profitability.
  - **Occupancy Rates:** Track changes in occupancy rates to assess the effectiveness of pricing and marketing initiatives.
- **Reporting and Monitoring:** Regularly report on KPIs and adjust strategies as needed to ensure continuous improvement.

Conclusion:

By implementing the recommendations outlined in this proposal, the hotel can enhance guest experiences, optimize operations, and achieve significant improvements in overall performance and profitability. The data-driven insights gained from Python analysis provide a strong foundation for making informed decisions and driving positive change within the business.