Name: Marvin Nguyen

Table of Content

1. Executive Summary

The purpose of this investigation was to assist Airbnb in better understanding the opinions of its Albany, New York, listings. Identifying the most active reviewers, investigating review trends over time, using sentiment analysis to highlight important negative features of listings, and using topic modelling to find the most often cited themes were our objectives. We give business owners useful insights by utilising artificial intelligence techniques, such as topic modelling with LDA and sentiment analysis with VADER on extracted word tokens. For instance, location and amenities were emphasised as key selling aspects, while preserving cleanliness and enhancing communication were noted as crucial elements. Strategic decisions to improve visitor pleasure and maximise corporate performance are directly supported by these insights.

2. Data Exploration

```
!pip install Cython
!pip install gensim
```

```
Requirement already satisfied: Cython in /usr/local/lib/python3.11/d
Requirement already satisfied: gensim in /usr/local/lib/python3.11/d
Requirement already satisfied: numpy<2.0,>=1.18.5 in /usr/local/lib/
Requirement already satisfied: scipy<1.14.0,>=1.7.0 in /usr/local/li
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/p
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/di
```

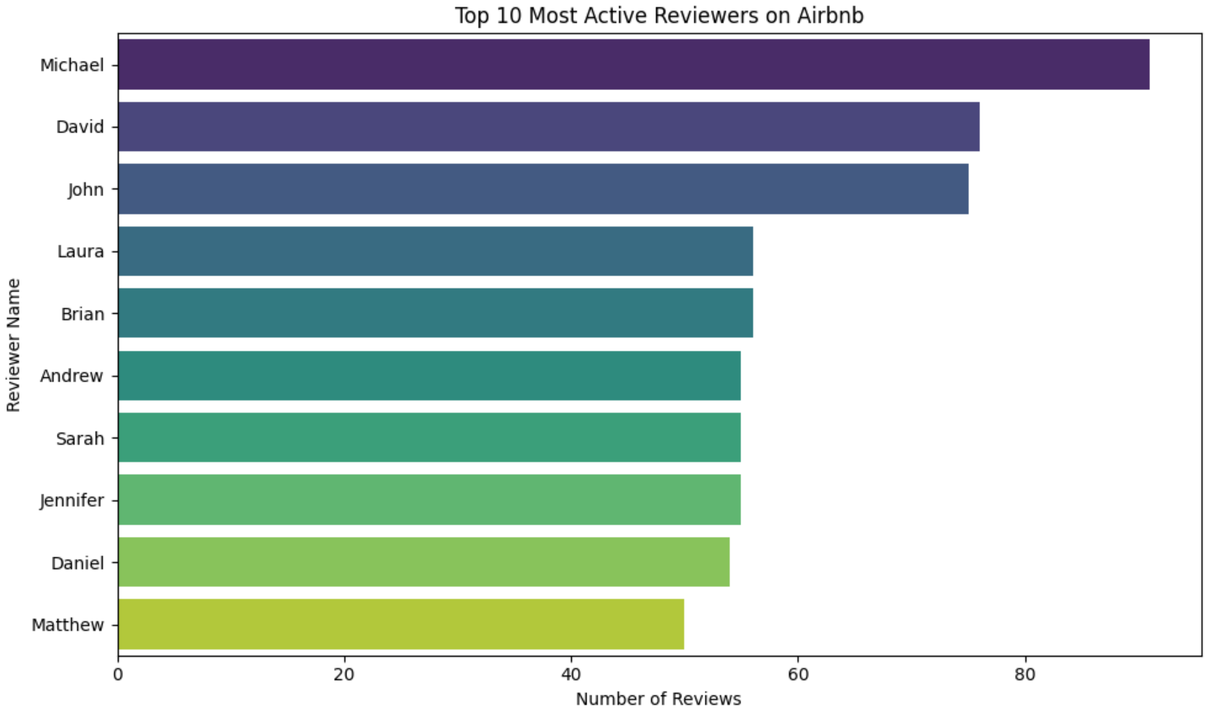
```
import pandas as pd

df = pd.read_excel('AirBnb_reviews.csv')
df.head()
```

	listing_id	id	date	reviewer_id	reviewer_name	comments
0	2992450	35913434	2015-06-23	15772025	Jennifer	We were pleased to see how 2nd Street and the ...
1	3820211	19110185	2014-09-06	19139729	Harriett Jordan	Terra was a great host, super informative

```
import matplotlib.pyplot as plt
import seaborn as sns

# Identify the top 10 most active reviewers
top_reviewers = df['reviewer_name'].value_counts().nlargest(10)
print("Top 10 Most Active Reviewers:")
print(top_reviewers)

# Bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x=top_reviewers.values, y=top_reviewers.index, palette=
plt.title('Top 10 Most Active Reviewers on Airbnb')
plt.xlabel('Number of Reviews')
plt.ylabel('Reviewer Name')
plt.tight_layout()
plt.show()
```

```
Top 10 Most Active Reviewers:
reviewer_name
Michael     91
David       76
```

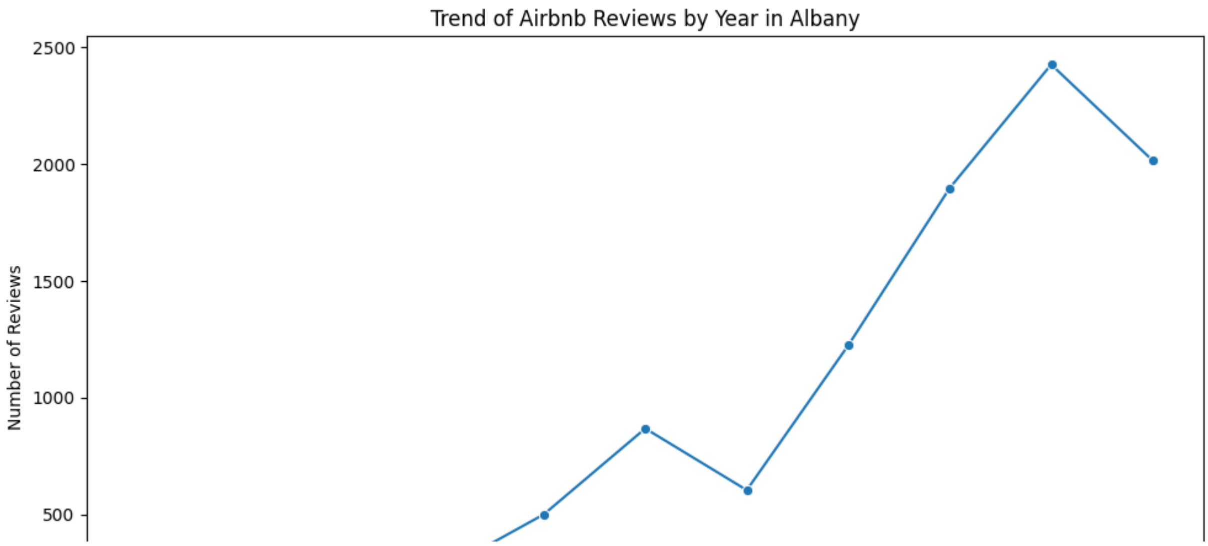
```
John         75
Laura        56
Brian        56
Andrew       55
Sarah        55
Jennifer     55
Daniel       54
Matthew      50
Name: count, dtype: int64
<ipython-input-3-c0c1eb931bba>:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be

  sns.barplot(x=top_reviewers.values, y=top_reviewers.index, palette
```

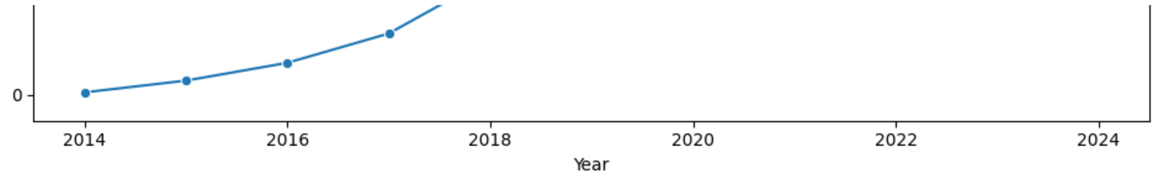


With 91 reviews, Michael is the most active reviewer according to the dataset. Brian (55), Laura (56), John (65), and David (70 reviews) are other very active reviewers. This suggests that a sizable portion of the dataset's total evaluations are contributed by a limited number of reviewers.

```
df['date'] = pd.to_datetime(df['date'], errors='coerce')
df['year'] = df['date'].dt.year

reviews_by_year = df.groupby('year').size().reset_index(name='revie
print("Review Counts by Year:")
print(reviews_by_year)

# Line plot
plt.figure(figsize=(10, 6))
sns.lineplot(data=reviews_by_year, x='year', y='review_count', mark
plt.title('Trend of Airbnb Reviews by Year in Albany')
plt.xlabel('Year')
plt.ylabel('Number of Reviews')
plt.tight_layout()
plt.show()
```

```
Review Counts by Year:
    year  review_count
0   2014             9
1   2015            59
2   2016           135
3   2017           260
4   2018           500
5   2019           868
6   2020           604
7   2021          1224
8   2022          1898
9   2023          2427
10  2024          2016
```





From just 9 reviews in 2014 to over 2,400 in 2023, the review counts clearly demonstrate an increased trend, suggesting that Albany has grown in popularity as an Airbnb destination. COVID-19 limits are probably to blame for the 2020 decline, which was followed by a robust recovery. Airbnb's growing popularity, efficient marketing, Albany's status as a capital hosting government events, and a post-pandemic spike in demand for travel are some of the factors propelling this trend.

3. Sentiment Analysis

```
from nltk.stem import PorterStemmer
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

porter = PorterStemmer()
stop_words = stopwords.words('english')

documents = df['comments']

cleaned_docs = []

for review in documents:
```

```
    try:
        cleaned = re.sub('[^A-Za-z]', ' ', review)
        cleaned = cleaned.lower()

        tokens = cleaned.split()

        tokens = [porter.stem(word) for word in tokens]

        tokens = [word for word in tokens if len(word) > 3]

        cleaned_review = ' '.join(tokens)
    except Exception as e:
        print(f"Error processing review: {e}")
        cleaned_docs.append('')
        continue

    cleaned_docs.append(cleaned_review)
    print('-[Review Text]: ', cleaned_review)

final_docs = []
for doc in cleaned_docs:
    filtered_tokens = [word for word in doc.split() if word not in
    final_doc = ' '.join(filtered_tokens)
    final_docs.append(final_doc)
    print('-[Cleaned Text]: ', final_doc)

df['cleaned_comments'] = final_docs
```

```
-[Cleaned Text]:  cute littl spot veri clean well equip full spice
-[Cleaned Text]:  great place exactli describ veri commun perfectl
-[Cleaned Text]:  nice place simpl clean
-[Cleaned Text]:  veri good place quiet area close place respond ev
-[Cleaned Text]:  great hous around neighborhood quiet peac
-[Cleaned Text]:  great commun check good place famili stay three
-[Cleaned Text]:  pleasant stay line clean comfort bathroom veri c
-[Cleaned Text]:  like pictur great locat
-[Cleaned Text]:  veri help start street park ampl
-[Cleaned Text]:  home like advertis veri clean conveni locat stay
-[Cleaned Text]:  great place excel host
-[Cleaned Text]:  beauti hous locat traffic could time definit aff
-[Cleaned Text]:  short stay downtown area close major shop
-[Cleaned Text]:  amaz stay veri help respons place veri clean ver
-[Cleaned Text]:  love stay cozi hous great three separ bedroom ki
-[Cleaned Text]:  nice hous accommod great
-[Cleaned Text]:  jason great host great commun veri accommod loca
-[Cleaned Text]:  amaz locat great clean privat space super friend
-[Cleaned Text]:  great place clean well furnish good locat want c
-[Cleaned Text]:  spot great locat walk distanc everi point intere
-[Cleaned Text]:  veri comfort wonder linen pillow bath towel plus
-[Cleaned Text]:  stay albani weekend place great locat jason real
-[Cleaned Text]:  nice place valu
```

```
    -[Cleaned Text]:  super cozi great locat
    -[Cleaned Text]:  great studio apart kitchen bathroom essenti need
    -[Cleaned Text]:  wonder place stay well locat charm area albani c
    -[Cleaned Text]:  jason great commun cute littl place exactli need
    -[Cleaned Text]:  great apart quick work trip close capitol restau
    -[Cleaned Text]:  jason awesom host place super clean comfi perfec
    -[Cleaned Text]:  cozi littl spot town everyth describ clean check
    -[Cleaned Text]:  beauti apart histor part albani
    -[Cleaned Text]:  clean nice place stay
    -[Cleaned Text]:  realli enjoy night jason airbnb apart veri clean
    -[Cleaned Text]:  great place locat veri clean veri comfort nois n
    -[Cleaned Text]:  jason place excel peopl veri clean extrem conven
    -[Cleaned Text]:  solid littl place cook sleep walk coffe could
    -[Cleaned Text]:  wonder appoint littl apart close park jason exce
    -[Cleaned Text]:  cute place fantast neighborhood found good food
    -[Cleaned Text]:  veri clean garden studio thought bright airi
    -[Cleaned Text]:  jason famili best basement accomod everyth need
    -[Cleaned Text]:  realli great place close everi restaur want befo
    -[Cleaned Text]:  avon server amoureux trajet tait long voulion co
    -[Cleaned Text]:  appart uniqu pour quipement avec document albani
    -[Cleaned Text]:  jason amaz host veri accommod meet need definit
    -[Cleaned Text]:  great apart everyth could need
    -[Cleaned Text]:  love room heart albani bunch cute shop restaur s
    -[Cleaned Text]:  jason great host veri commun help littl snag che
    -[Cleaned Text]:  jason super attent veri help apart cute clean co
    -[Cleaned Text]:  great place solo trip central locat near restaur
    -[Cleaned Text]:  jason place excel veri clean everyth need jason
    -[Cleaned Text]:  place middl histor albani easi walk site
    -[Cleaned Text]:  place neighborhood cute hous came attend cimafun
    -[Cleaned Text]:  veri nice place exactli expect veri well locat g
    -[Cleaned Text]:  stop albani jason apart perfect locat close empi
    -[Cleaned Text]:  nice night stay good locat albani
    -[Cleaned Text]:  space veri clean well kept great locat everyth v
    -[Cleaned Text]:  locat perfect jason place veri clean instruct cl
    -[Cleaned Text]:  great spot recommend host locat enough
    -[Cleaned Text]:  awesom locat veri clean perfect littl town alban
```

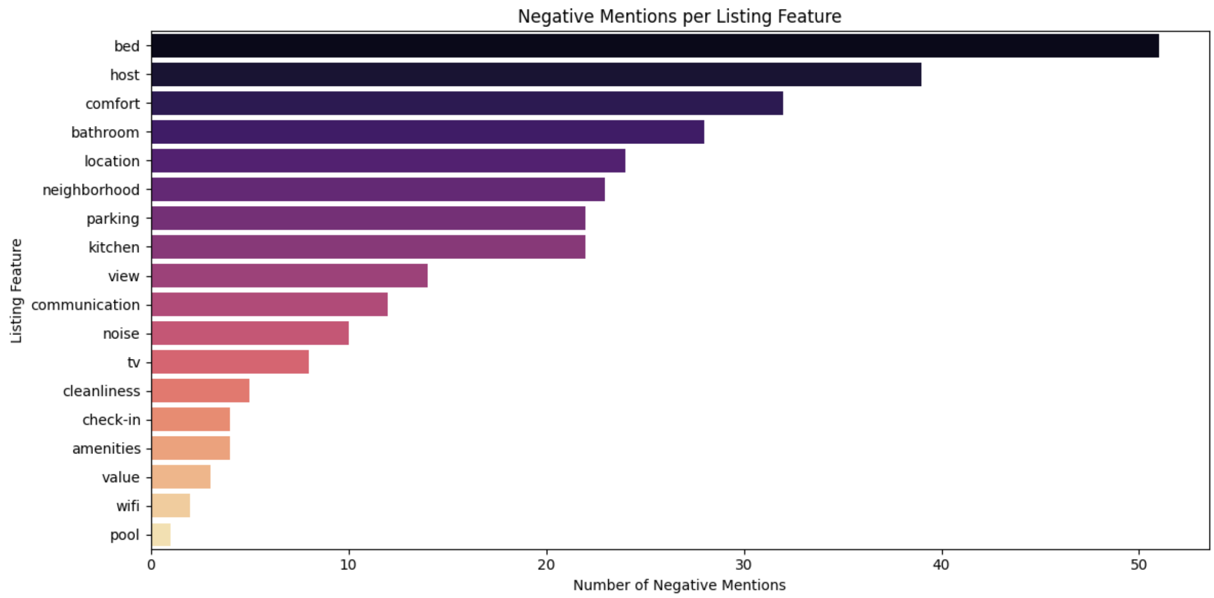
```
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
import matplotlib.pyplot as plt
import seaborn as sns

nltk.download('vader_lexicon')

df['comments'] = df['comments'].astype(str)

sia = SentimentIntensityAnalyzer()
df['sentiment_score'] = df['comments'].apply(lambda text: sia.polar

df['sentiment_label'] = df['sentiment_score'].apply(lambda score: '
                                                      else ('positiv

print(df[['comments', 'sentiment_score', 'sentiment_label']].head()
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
                                            comments  sentiment_scor
0  We were pleased to see how 2nd Street and the ...           0.988
1  Terra was a great host, super informative, and...           0.976
2  The apartment was clean, nice & convenient. I ...           0.915
3  Very comfortable apartment and super central t...           0.968
4  Our favourite place to stay in Albany! The pla...           0.981

  sentiment_label
0        positive
1        positive
2        positive
3        positive
4        positive
```

```
features = [
    "location", "cleanliness", "comfort", "host", "neighborhood",
    "noise", "value", "amenities", "check-in", "communication",
    "wifi", "parking", "view", "bed", "bathroom", "kitchen", "pool"
]

feature_negative_counts = {}
for feature in features:
    mask = df['comments'].str.contains(feature, case=False, na=Fals
    negative_count = df[mask & (df['sentiment_label'] == 'negative'
    feature_negative_counts[feature] = negative_count

import pandas as pd
feature_negative_df = pd.DataFrame.from_dict(feature_negative_count
feature_negative_df = feature_negative_df.sort_values(by='Negative
```

```
print("Negative Mentions by Feature:")
print(feature_negative_df)

plt.figure(figsize=(12, 6))
sns.barplot(x=feature_negative_df['Negative Mentions'], y=feature_n
plt.title("Negative Mentions per Listing Feature")
plt.xlabel("Number of Negative Mentions")
plt.ylabel("Listing Feature")
plt.tight_layout()
plt.show()
```

```
Negative Mentions by Feature:
               Negative Mentions
bed                           51
host                          39
comfort                       32
bathroom                      28
location                      24
neighborhood                  23
parking                       22
kitchen                       22
view                          14
communication                 12
noise                         10
tv                             8
cleanliness                    5
check-in                       4
amenities                      4
value                          3
wifi                           2
pool                           1
<ipython-input-7-f8e443876119>:20: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be

  sns.barplot(x=feature_negative_df['Negative Mentions'], y=feature_
```



I utilized VADER to assess sentiment scores in my analysis. The overarching positive sentiment score indicates that visitors generally feel satisfied with their Airbnb experiences. However, despite this generally optimistic outlook, there is a notable presence of negative feedback pertaining to specific aspects, such as comfort (32), host communication (39), bed quality (51), and toilet conditions (28). This suggests that although the overall impressions are positive, enhancing these critical areas could further elevate visitor satisfaction.

4. Topic Modeling

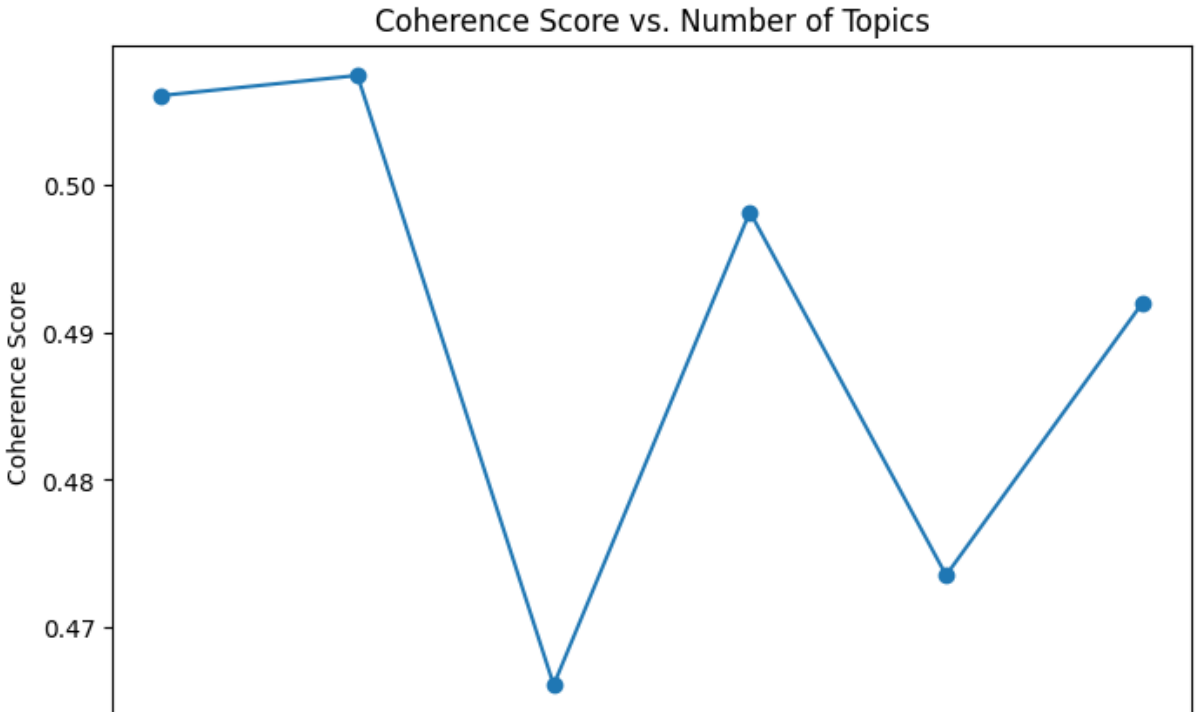
```
import re
import nltk
from nltk import pos_tag, word_tokenize
from nltk.corpus import stopwords

nltk.download('punkt_tab')
nltk.download('averaged_perceptron_tagger_eng')
nltk.download('stopwords')

stop_words = set(stopwords.words('english'))

def extract_nouns(text):
    """
    Cleans text, tokenizes, performs POS tagging, and extracts only
    Filters out stop words and words shorter than 4 characters.
    """

    text = re.sub('[^A-Za-z]', ' ', text.lower())
    tokens = word_tokenize(text)

    nouns = [word for word, pos in pos_tag(tokens) if pos.startswit

    nouns = [word for word in nouns if word not in stop_words and l
    return nouns


df['noun_tokens'] = df['comments'].astype(str).apply(extract_nouns)

df = df[df['noun_tokens'].map(len) > 0]
```

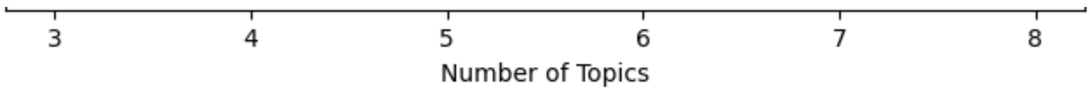
```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-t
[nltk_data]       date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
from gensim import corpora

dictionary = corpora.Dictionary(df['noun_tokens'])

dictionary.filter_extremes(no_below=5, no_above=0.5)

corpus = [dictionary.doc2bow(text) for text in df['noun_tokens']]
```

```
from gensim.models import LdaModel
from gensim.models.coherencemodel import CoherenceModel
import matplotlib.pyplot as plt

topic_range = range(3, 9)
coherence_values = []
model_list = []

for num_topics in topic_range:
    lda_model = LdaModel(corpus=corpus, id2word=dictionary, num_top
                         passes=10, random_state=42)
    model_list.append(lda_model)
    coherence_model = CoherenceModel(model=lda_model, texts=df['nou
                                     dictionary=dictionary, coheren
    coherence = coherence_model.get_coherence()
    coherence_values.append(coherence)
    print(f"Num Topics = {num_topics} --> Coherence Score = {cohere

plt.figure(figsize=(8, 5))
plt.plot(topic_range, coherence_values, marker='o')
plt.xlabel("Number of Topics")
plt.ylabel("Coherence Score")
plt.title("Coherence Score vs. Number of Topics")
plt.show()
```

```
Num Topics = 3 --> Coherence Score = 0.5060
Num Topics = 4 --> Coherence Score = 0.5074
Num Topics = 5 --> Coherence Score = 0.4661
Num Topics = 6 --> Coherence Score = 0.4981
Num Topics = 7 --> Coherence Score = 0.4735
Num Topics = 8 --> Coherence Score = 0.4920
```






```
optimal_topics = topic_range[coherence_values.index(max(coherence_v
print(f"\nOptimal number of topics determined: {optimal_topics}")

optimal_model = model_list[coherence_values.index(max(coherence_val

print("\nDiscovered Topics:")
topics = optimal_model.print_topics(num_words=10)
for idx, topic in topics:
    print(f"Topic {idx+1}: {topic}")
```

```
Optimal number of topics determined: 4

Discovered Topics:
Topic 1: 0.042*"room" + 0.028*"kitchen" + 0.019*"bathroom" + 0.016*"
Topic 2: 0.104*"stay" + 0.095*"apartment" + 0.060*"location" + 0.031
Topic 3: 0.066*"home" + 0.050*"host" + 0.044*"stay" + 0.036*"house"
Topic 4: 0.266*"place" + 0.043*"host" + 0.035*"location" + 0.033*"ev
```

In the analysis, the four-topic model came out on top with the highest coherence score—about 0.5074. This means it struck the best balance between having distinct themes and being easy to understand. While the three-topic model also did a good job, the four-topic approach gave us a little more detail without breaking the content into too many pieces.

The four topics we uncovered tell an interesting story. One topic zeroes in on amenities and comfort, focusing on the little in-room features that make a stay cozy. Another topic looks at location and practical details like how close things are and the availability of parking. A third topic highlights the role of the host and local experience, showcasing how hospitality and local recommendations can enrich a stay. Lastly, the fourth topic centers on host communication and the check-in process, which are crucial for a smooth experience. Overall, these insights offer clear, actionable ideas for boosting guest satisfaction and improving business performance.

```
!pip install wordcloud
from wordcloud import WordCloud
import math

rows_wc = math.ceil(optimal_topics / 4)
fig, ax = plt.subplots(rows_wc, 4, figsize=(15, 2.5 * rows_wc))

if rows_wc == 1:
    ax = ax.reshape(1, -1)

[axi.set_axis_off() for axi in ax.ravel()]

topics_wc = optimal_model.show_topics(num_topics=optimal_topics, nu
for topic_idx, topic in enumerate(topics_wc):

    word_freq = dict(topic[1][:10])
    wordcloud = WordCloud(background_color="white").generate_from_f
    row_idx = topic_idx // 5
    col_idx = topic_idx % 5
    ax[row_idx, col_idx].imshow(wordcloud)
    ax[row_idx, col_idx].set_title(f"Topic {topic_idx+1}")
plt.tight_layout()
plt.show()
```

```
Requirement already satisfied: wordcloud in /usr/local/lib/python3.1
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python
Requirement already satisfied: pillow in /usr/local/lib/python3.11/d
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/py
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/p
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/p
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/pyt
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/py
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/li
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11
```

```
pinterpretation = """
Topic 1 – Amenities & Comfort was characterized by words such as "r
For this reason, business owners must improve the quality and the d

Topic 2 – Location & Stay Factors has high counts of "stay," "apart
where the property is and how accessible it is. Hosts could emphasi

Topic 3 – Host & Local Experience has high-frequency terms such as
host personality are very significant to guests. By way of response

Topic 4 – Host Communication & Check-in is governed by words such a
smooth check-in are the requirements. Therefore, business owners sh
"""

print(pinterpretation)
```

```
Topic 1 – Amenities & Comfort was characterized by words such as "ro
For this reason, business owners must improve the quality and the de

Topic 2 – Location & Stay Factors has high counts of "stay," "apartm
where the property is and how accessible it is. Hosts could emphasiz

Topic 3 – Host & Local Experience has high-frequency terms such as "
host personality are very significant to guests. By way of response,

Topic 4 – Host Communication & Check-in is governed by words such as
smooth check-in are the requirements. Therefore, business owners sho
```

I found a number of important themes based on our topic modelling research, which involved extracting just nouns from the English reviews using text processing. The subjects that are most often brought up are:

- Location & Neighbourhood: Visitors often talk about things like the property's location, how close it is to attractions, and the general atmosphere of the neighbourhood.
- Cleaning & upkeep: A lot of evaluations stress how crucial cleaning and regular upkeep are.
- Amenities & Comfort: Remarks frequently highlight the standard of amenities such as kitchenware, wifi, and general comfort when visiting.
- Host Interaction & Communication: Evaluations stress the importance of responsive hosting, easy check-in procedures, and efficient communication.

Although pricing did not surface as a separate topic in the model, the prevalence of negative mentions and recurring review patterns indicate that it remains a notable concern Price is a sensitive topic for guests, who often discuss whether they felt they got a good deal.

Suggestions:

In order to enhance their business efficiency, Airbnb hosts ought to:

- Improve Local Listings: To capitalise on the interest in location, include comprehensive information about the neighbourhood and neighbouring attractions.
- Invest in Cleaning Protocols: To address concerns about cleanliness, implement strict cleaning and maintenance routines.
- Enhance Amenities: Consistently update the property's features and make sure that its top-notch amenities are prominently displayed.
- Enhance Host Training: Put an emphasis on efficient check-in processes and good communication.
- Optimise Pricing Strategies: Take into account dynamic pricing models and make sure that the value being supplied is communicated effectively in the pricing specifics.