BEMM466 Business Project

Title: Sentiment Analysis of guest reviews for apartment-type Airbnb accommodation in London

Submitted by:
Chonchaya Yuthanarong
Student ID: 730068096

Supervisor:
Dr Stuart So

MSc Business Analytics, Year 2023/2024
University of Exeter

Executive Summary

The rise of the sharing economy, particularly through platforms like Airbnb has profoundly disrupted traditional business models which present both new challenges and opportunities (Guttentag, 2015). This research focuses on understanding evolving consumer perceptions of apartment-type Airbnb accommodations in London, especially global uncertainties such as Brexit and the COVID-19 pandemic. By analysing guest reviews, this study explores how external shocks have influenced market dynamics and consumer sentiments over time. The findings provide a deeper understanding of consumer behaviours and perceptions that offer practical applications for Airbnb hosts, policymakers, and industry leaders. For example, hosts can use these insights to adjust pricing models based on predicted demand fluctuations or enhance guest satisfaction by prioritising cleanliness and convenience. Policymakers might use these insights to develop regulations that balance market growth with consumer rights ensuring the sharing economy evolves beneficially for all parties. By translating sentiment analysis results into actionable data, this research supports the development of more informed and adaptable strategies in response to a rapidly changing environment.

The research primarily aims to offer actionable insights tailored for key stakeholders. By analysing the evolution of consumer sentiments and market dynamics, it aims to provide strategic guidance crucial for understanding broader economic and societal impacts (Li, Li, & Zhang, 2021). The findings could significantly contribute to the economic stability of the Airbnb market and the sustainable development of the hospitality industry.

Given the increasing importance of data-driven decision-making, understanding guest sentiments is vital for shaping effective strategies within the sharing economy. The study highlights the potential for boosting the tourism industry's growth through enhanced guest experiences which could increase bookings, retention, and reputation in a competitive sector. This research utilises advanced sentiment analysis tools and topic modelling to interpret guest reviews, uncovering underlying themes and correlations between apartment features and sentiments. By leveraging machine learning technologies, it identifies the extent to which these factors influence guest perceptions. Furthermore, the study improves the precision of predicting pricing and demand trends by integrating advanced statistical methods offering stakeholders a basis for strategic guidance as they navigate the evolving sharing economy landscape.

Notably, this study is among the first to apply advanced sentiment analysis tools and topic modelling specifically to Airbnb guest reviews in the context of significant external shocks. It provides a nuanced understanding of how these shocks influence market dynamics and consumer sentiments, contributing uniquely to the existing body of knowledge on the sharing economy.

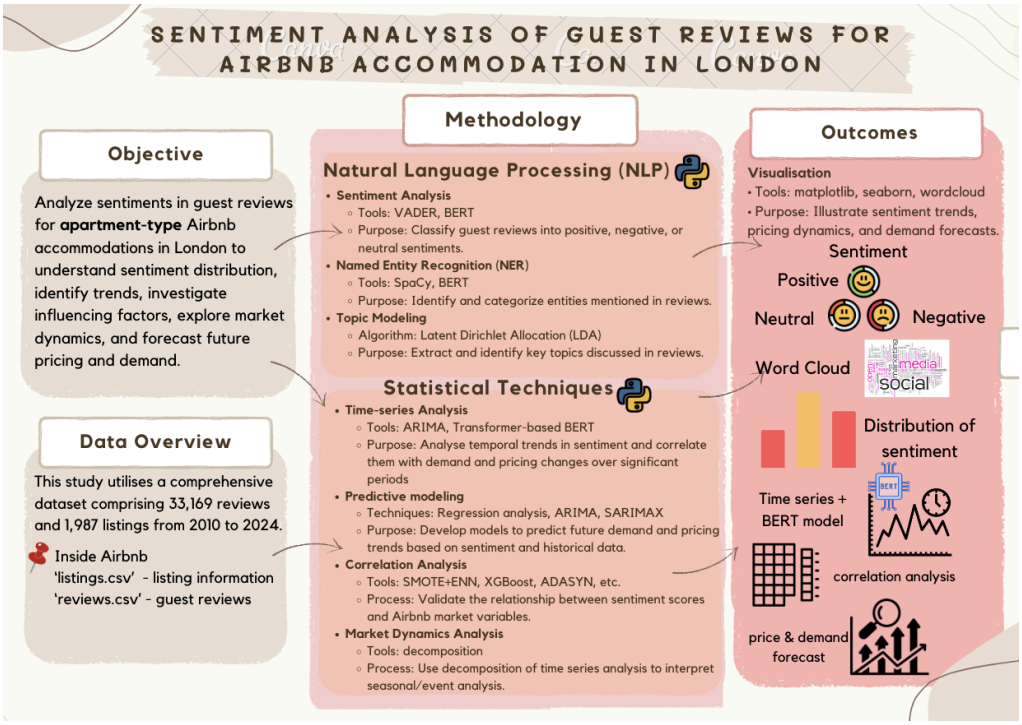


Figure 1. Graphical Abstract for sentiment analysis

The analysis was conducted on a dataset comprising 1,987 listings and 33,169 reviews from apartment- type Airbnb accommodations in London from 2010 to 2024. Positive sentiments overwhelmingly dominate guest reviews with 94.63%, with cleanliness, location, and convenience being the most frequently cited factors of guest satisfaction. These findings confirm the hypothesis that service quality is a significant driver of positive consumer perceptions in the Airbnb market (Li, Li, & Zhang, 2021). 2.64% of reviews were neutral, and 2.73% were negative, with the latter often related to issues like uncomfortable stays and small spaces. However, significant shifts in sentiment were observed during periods of socio-political instability, such as Brexit and COVID-19 pandemic as there was a 38% increase in neutral and some negative sentiments reflecting heightened uncertainty and disruptions in the travel industry.

The study further identified strong correlations between guest sentiments and specific apartment features. Properties with higher ratings for amenities, location convenience, and room quality consistently received more positive reviews, with the machine learning model demonstrating an accuracy rate of approximately 82.39% in predicting these outcomes. Negative sentiments were often associated with issues like uncomfortable stays and small spaces. Interestingly, the analysis revealed that pricing, while important, did not significantly influence sentiment scores suggesting that guests prioritise quality and experience over cost.

Through time-series models, the research captured fluctuations in pricing and demand. The models achieved a mean absolute error (MAE) of 18.38 and an R-squared (R^2) value of 0.71, indicating a high level of accuracy in predicting future trends. These predictive models suggest practical steps, such as dynamically adjusting prices during high-demand periods or in response to external events that shift consumer sentiment. This approach provides Airbnb hosts and policymakers with valuable foresight enabling them to proactively adjust strategies in response to anticipated market changes with greater confidence and effectiveness.

Based on the research findings, strategic recommendations are proposed to enhance the sustainability and effectiveness of Airbnb operations, particularly in the London market. For instance, Airbnb hosts could implement targeted cleaning protocols to address specific guest concerns highlighted in reviews or adjust service offerings to better meet consumer expectations during socio-political disruptions. Maintaining high standards in cleanliness, location accuracy, and convenience is crucial not only for addressing guest concerns but also for driving higher satisfaction and retention rates, thereby enhancing competitiveness. Implementing dynamic pricing strategies that adjust rates according to market conditions, seasonal trends, and significant events will help hosts remain competitive and maximise revenue. Developing contingency plans to mitigate disruptions and maintain market stability in response to external factors is essential.

The study also emphasises the importance of leveraging predictive analytics to anticipate changes in guest behaviour and optimise decision-making processes. Collaboration with policymakers is crucial to ensuring the long-term sustainability of the Airbnb market, promoting fair competition, protecting consumer rights, and fostering community well-being. Additionally, hosts should encourage ongoing guest feedback to drive continuous improvement, thereby helping them maintain a competitive edge and foster long-term success.

While the study provides robust findings, it also identifies areas for further research, such as exploring the long-term effects of regulatory changes and integrating emerging technologies in sentiment analysis. Future studies could refine methodologies by improving sentiment analysis tools with culturally aware models and addressing current machine learning techniques' limitations in understanding sarcasm and context-dependent expressions. These recommendations aim to equip stakeholders with the tools and strategies necessary to navigate the complexities of the sharing economy effectively and sustainably.

The study highlights the importance of ethical data handling, particularly regarding the use of AI and machine learning tools. The anonymized data is securely stored with encryption and access controls, restricting access exclusively to authorised personnel within the University of Exeter's OneDrive system. The data will be retained for a maximum period of 4 months to facilitate any necessary verification or follow-up studies and will be securely destroyed in November 2024, following the awarding of the degree. The research addresses potential biases in these tools and highlights the need for transparency and accountability, aligning with both the EU AI Act’s principles (European Parliament, 2023) and the UK’s pro-innovation regulatory approach (Gov.UK, 2024).

In conclusion, the strategies proposed provide Airbnb hosts, policymakers, and industry stakeholders with practical solutions that can substantially improve the quality and competitiveness of Airbnb operations in London. Hosts can improve profitability and market position by focusing on key drivers of guest satisfaction to achieve higher ratings and increased bookings. Dynamic pricing and contingency planning will enable hosts to adapt to market fluctuations and external shocks. For policymakers, the study provides insights to develop balanced regulations that promote sustainable growth, fair competition, and consumer protection. Collaboration among stakeholders is essential to maintaining a thriving and well-regulated sharing economy that benefits both the local community and the broader tourism industry.

Table of Contents

Introduction

Background

Airbnb has significantly transformed the accommodation sector by providing unique and flexible lodging options through its sharing economy model (Guttentag, 2015). As the platform continues to expand and diversify its offerings, understanding consumer experiences and sentiments becomes critical for ensuring sustainable growth and maintaining Airbnb's competitive edge. While user-generated content like guest reviews offers valuable insights, the subjective and unstructured nature of this data presents challenges for systematic analysis. This necessitates the application of advanced analytical tools and methodologies (Garcia & Wang, 2020).

Hosts, who list their properties for short-term rental, play a key role in shaping the guest experience through decisions on pricing, communication, and service quality (Tussyadiah & Pesonen, 2016). To remain competitive, hosts must effectively leverage insights from guest feedback to refine their offerings and anticipate guest needs. These insights can also be beneficial to policymakers and regulators. The rapid expansion of apartment-type listings in a global city like London, as a leading global city and major tourist destination with high demand for unique accommodations, makes London an ideal setting for studying consumer engagement with Airbnb (Guttentag, 2015; Li, Li, & Zhang, 2021; Cocola-Gant, 2016). The city's experience with Airbnb can provide valuable lessons on balancing innovation and regulation ensuring that growth in the sharing economy does not compromise consumer protection or urban quality of life (Li, Li, & Zhang, 2021).

This study seeks to fill a gap in existing research by providing a comprehensive evaluation of guest sentiments through advanced Natural Language Processing (NLP) techniques. By thoroughly analysing the distribution of sentiments in Airbnb reviews for apartment-type accommodations in London, the research aims to identify key factors that influence guest satisfaction and dissatisfaction. The findings are intended to assist Airbnb hosts in enhancing service quality and support policymakers in developing regulations that ensure a sustainable short-term rental market.

Problem statement

Airbnb has emerged as a significant player in the global hospitality industry, especially in urban environments like London (Guttentag, 2015). The city's diverse tourist population and the high demand for unique accommodations have resulted in a substantial increase in apartment-type Airbnb listings (Li, Li, & Zhang, 2021). Despite the growth, while Airbnb's growth is undeniable, the subjective and often unstructured nature of guest reviews poses a significant challenge for stakeholders aiming to extract actionable insights to enhance service quality and inform strategic decisions (Garcia & Wang, 2020).

This study seeks to address the difficulty of systematically analysing guest feedback to derive meaningful insights, particularly in the context of evolving consumer preferences and external events. By focusing on sentiment analysis, this research aims to bridge the gap between raw data and valuable business intelligence. The goal is to aid stakeholders with a deeper understanding of market dynamics which enable them to forecast future trends and make informed decisions that enhance the competitiveness and sustainability of Airbnb operations in London.

Project Objective

This research aims to conduct a comprehensive analysis of the sentiments expressed in guest reviews for apartment-type Airbnb accommodations in London. Utilising advanced sentiment analysis techniques, this study seeks to illuminate the distribution of sentiments, identify temporal trends, and explore the factors that significantly influence guest experiences. Additionally , the study aims to examine the contextual dynamics of the Airbnb market in London and forecast future trends in pricing and demand based on sentiment analysis.
To achieve these objectives, the research integrates both cross-sectional and time-series analyses. The cross-sectional analysis will provide a snapshot of guest sentiments at a specific point in time while the time-series analysis will track how these sentiment trends evolve over extended periods with particular emphasis on key events such as Brexit and the COVID-19 pandemic. This approach aims to uncover the evolution of sentiments over time and how they are influenced by significant external factors, providing a dynamic understanding of guest experiences.

By combining these methods, the research will provide a comprehensive view of both the static and dynamic aspects of guest sentiments offering deeper insights into what influences guest satisfaction. Beyond advancing academic understanding of sentiment dynamics in the sharing economy, the study seeks to offer actionable insights for Airbnb hosts, policymakers and the broader tourism industry, contributing to the sustainable growth and competitive advantage of the Airbnb market in London.

Significance and motivation of the research

Significant of the study

The significance of this research lies in its potential to provide valuable insights into the rapidly evolving landscape of short-term rentals, particularly within the context of Airbnb in London. As one of the leading platforms in the sharing economy, Airbnb's influence on the hospitality industry is profound and understanding the sentiments of its users is crucial for several reasons.

Firstly, the research will provide actionable insights for Airbnb hosts enabling them to enhance service quality by identifying key drivers of guest satisfaction and dissatisfaction. This can lead to improved ratings, increased bookings, and greater success on the platform. Secondly, the study will inform policymakers about the dynamics of the short-term rental market, particularly considering significant events like Brexit and the COVID-19 pandemic. These insights are crucial for developing evidence-based regulations that balance the economic benefits of short-term rentals with the need for sustainable urban development. Finally, this research will contribute to the academic discourse on sentiment analysis within the hospitality industry. By applying a robust methodological framework that integrates various NLP techniques, the study will advance the understanding of how to effectively analyse user-generated content. This framework can be adapted to other contexts, broadening the impact of the research beyond Airbnb in London.

Motivation for the research

The motivation behind this research is deeply rooted in the significant impact that the sharing economy, particularly Airbnb, has had on traditional hospitality models. As Airbnb continues to revolutionise how accommodations are booked and experienced, it becomes increasingly important to understand the sentiments and perceptions of guests. This study aims to equip Airbnb hosts and policymakers with actionable intelligence to enhance service quality and inform regulatory frameworks. For hosts, sentiment analysis provides valuable insights to tailor services, improve guest satisfaction and boost competitiveness, especially in a saturated market like London. For policymakers, understanding the sentiment landscape is essential for developing regulations that balance the benefits of the sharing economy with consumer protection and fair competition contributing to a more sustainable short-term

rental market. This study seeks to transform raw data into actionable policy recommendations while promoting a balanced and well-regulated market environment.

As of December 2023, London hosts over 91,000 active Airbnb listings marking a significant 32% year-over-year growth from 2022. This rapid expansion underscores the urgent need for data-driven insights to guide hosts and regulators alike in maintaining service quality and ensuring sustainable growth in a highly dynamic market (Emily, 2024). As the volume of user-generated content continues to rise, the urgency to effectively harness and interpret this data becomes increasingly critical. Guest reviews are inherently subjective and unstructured, posing significant challenges for analysis. Advanced Natural Language Processing (NLP) techniques such as sentiment analysis, offer the potential to decode these complex data sources which transform them into actionable insights that can drive strategic decisions.

Another driving factor is the unprecedented disruptions caused by significant global events, such as Brexit and the COVID-19 pandemic. These events have drastically altered travel patterns, consumer behaviour, and market dynamics which create a need to understand their impact on guest sentiments. Focusing on apartment-type accommodations in London, this study targets a popular segment of travellers who prioritise comfort, cost-efficiency and location advantages, particularly in a global city and major tourist destination like London. Additionally, the integration of advanced NLP methodologies, such as DistilBERT, Named Entity Recognition (NER), and Latent Dirichlet Allocation (LDA), highlights the innovative approach of this research. These tools will be crucial in uncovering both broad trends and specific aspects of guest sentiments enabling a deeper understanding of the factors that influence guest experiences and preferences and provide actionable insights for improving service offerings and market strategies.

Research Question

These insights and motivations have led to the development of research questions aimed at addressing the key challenges identified:

RQ 1: What is the distribution of positive, negative, and neutral sentiments among reviews for apartment-type Airbnb accommodations in London, and what specific words or phrases are most frequently mentioned in these reviews?

RQ 2: How do sentiment trends vary over time, particularly during significant events such as economic crisis (e.g., Brexit in UK) and COVID-19 pandemic, and are there notable differences in sentiment expressions during those specific phases?

RQ 3: How do various factors influence the predominance of positive or negative evaluations in guest reviews for apartment-type Airbnb accommodations?

RQ 4: To what extent specific features of apartment-type accommodations correlate with the sentiments expressed in guest reviews?

RQ 5: How do the growth and dynamics of the Airbnb market in London influence guest sentiments in apartment-type accommodations?

RQ 6: How can sentiment analysis of guest reviews for apartment-type Airbnb accommodations in London be used to forecast future demand and pricing trends?

Literature Review

I. Overview of Relevant Literature

The rapid growth of Airbnb has undeniably transformed the hospitality landscape, particularly in cosmopolitan hubs like London. Guttentag (2015) explores the emergence of Airbnb through the lens of disruptive innovation theory and highlighting its rapid growth by offering cost savings, household amenities, and authentic local experiences. This rise has significantly impacted traditional accommodation sectors and brought forth regulatory and tax challenges. Reports by organisations like the OECD (2020) emphasise the need for new regulatory frameworks to manage the sharing economy's growth and its impact on local markets, underscoring the importance of insights derived from consumer data to inform policy decisions.

As a leading platform in the sharing economy, Airbnb offers flexible accommodation options catering to a diverse array of travellers (Li, Li, & Zhang, 2021). The platform's exponential growth of user-generated content, primarily in the form of online reviews, presents both opportunities and challenges. These reviews offer a wealth of information on guest experiences, preferences, and sentiments. However, extracting meaningful insights from this unstructured data requires sophisticated analytical tools and methodologies. Effective sentiment analysis can yield critical insights into guest satisfaction and the broader economic impact of short-term rentals which are essential for both market participants and policymakers (Garcia & Wang, 2020).

Sentiment analysis, an advancing tool in Natural Language Processing (NLP), has become a powerful method for deriving valuable insights from unstructured textual data. Studies such as those by Santos et al. (2022) have demonstrated how sentiment analysis can uncover distinctions in guest feedback, influenced by factors such as host type, location, and available amenities. This detailed understanding allows for a targeted approach to improving the Airbnb experience by addressing specific pain points while highlighting strengths. The World Tourism Organization (UNWTO) has also acknowledged the potential of sentiment analysis to enhance the competitiveness of tourism markets by aligning service offerings with guest expectations (UNWTO, 2021).

Medhat et al. (2014) present a comprehensive overview of sentiment analysis techniques, dividing them into two primary categories: lexicon-based approaches (rely on predefined dictionaries of words and their associated sentiments) and machine learning methods (learn patterns from labelled data to classify sentiment). Lexicon-based approaches, such as VADER (Hutto & Gilbert, 2014), are widely used for their simplicity and effectiveness in handling social media text. However, researchers have also integrated machine learning techniques leveraging labelled datasets to train models that can accurately classify sentiment (Stieglitz & Dang-Xuan, 2018). These machine learning models often outperform lexicon-based approaches in terms of accuracy but require larger amounts of labelled data for training. By generating actionable insights, these techniques can significantly enhance business strategies and support informed policy-making aligning with broader trends in digital transformation within the tourism sector as discussed in the literature (Buhalis & Sinarta, 2019).

Garcia & Wang (2020) also utilised sentiment analysis to explore the relationship between sentiment and price in the sharing economy. They showcased how sentiment analysis could offer insights into pricing dynamics and forecast trends. This capability is particularly relevant for policymakers who need to understand market behaviour to craft regulations that balance market growth with consumer protection. Moreover, researchers have used sentiment analysis to examine how external events affect guest experiences. Various studies have delved into the influence of occurrences like disasters (Qiang & Han 2021) and political turmoil (Jahanbakhsh & Mao 2020) on sentiments expressed in tourism reviews. The impact of major events such as Brexit and the COVID-19 pandemic on guest sentiment in Airbnb remains an underexplored area indicating potential for further research to guide both business and policy interventions.

This research enhances the body of knowledge by merging lexicon based and machine learning methods integrating NLP strategies such as Named Entity Recognition (NER) and topic modelling. It also explores how major events influence guest sentiment, in the London Airbnb market. Consequently, this research seeks to contribute to a more comprehensive understanding of the factors influencing guest satisfaction and the dynamics of the Airbnb market in a major urban centre.

II. Gaps or Controversies in the Literature

Despite advances in sentiment analysis techniques, significant gaps remain in the context of Airbnb research. One major gap is the lack of integrated approaches that combine multiple NLP techniques to effectively grasp the range of opinions, emotions, and core themes conveyed in reviews. While existing studies often focus on individual methods like lexicon-based sentiment analysis or topic modelling, a comprehensive understanding of guest sentiments requires a more holistic approach. This study addresses this by integrating diverse NLP tools, including lexicon-based and machine learning-based sentiment analysis, Named Entity Recognition (NER), and topic modelling. This method aims to reveal patterns and connections that might go unnoticed when using individual methods aligning with suggestions from industry reports on using AI to improve tourism services (WTTC, 2021).

Another gap in the literature is the underutilization of sentiment analysis in attempting to predict future market movements but specifically with Airbnb. While some research has used sentiment analysis to predict current guest experiences, its utility in forecasting future fluctuations in pricing and demand is currently underused. This study intends to fill this void by using time-series analysis methods to investigate whether changes in guest sentiment is associated with market dynamics. While 'market dynamic analysis' traditionally refers to assessing market conditions and consumer behaviour changes, it is used to describe the application of sentiment analysis in inferring these changes. Prior studies have demonstrated that sentiment analysis is an effective tool for uncovering market trends and consumer behaviour patterns in the sharing economy (Garcia & Wang, 2020).

Using ARIMA models to forecast on historical data trends and implementing SARIMAX models by inputting sentiment scores as exogenous variables will provide more accurate predictions of future prices and demand. The work is of high importance for hosts, property managers and policymakers as it provides a non-binary picture of how sentiment drives market trends over time (Sanh et al., 2019; Devlin et al., 2019; He et al.,2020).

Additionally, The COVID-19 pandemic and Brexit have undeniably reshaped the tourism and hospitality industry, particularly impacting the short-term market. However, limited research exists on how these events have specifically affected guest sentiment on Airbnb in London. This study aims to fill this gap by conducting a longitudinal analysis of guest reviews, which will examine how sentiment has evolved over time in response to these significant events. This analysis will provide valuable insights into the resilience and adaptability of the Airbnb market in the face of external shocks, as well as the changing needs and expectations of guests in a post-pandemic landscape.

III. Debates and Future Directions

Despite its potential, sentiment analysis is not without its ongoing debates and obstacles. The accuracy and reliability of plenty of tools and techniques, such as the widely used VADER and TextBlob, are still being debated. One primary challenge is determining the extent to which these computational methods can comprehend and interpret those complex and truly expressed personal emotions and opinions in textual data.

This study acknowledges these limitations and seeks to contribute to the ongoing scholarly discourse by rigorously evaluating the performance of both TextBlob and VADER in the context of Airbnb reviews. Recent applications of sentiment analysis in similar contexts have shown mixed results. For instance, Zhu et al. (2020) used sentiment analysis to examine Airbnb reviews in New York City, finding that while broad sentiment trends were captured, subtle social implications were regularly missed. Similarly, Abeysinghe and Bandara (2022) applied VADER to TripAdvisor reviews which demonstrate its strength in detecting obvious sentiments but limitations in understanding sarcasm and context-dependent expressions.

By combining NER and topic modelling methods, the research aims to enhance the clarity and depth of sentiment analysis, potentially uncovering subtle patterns and relationships that might be overlooked by individual tools alone. This approach is guided by cross-disciplinary research combining linguistics and computer science, particularly drawing on studies like Chen and Skiena's (2022) work on culturally-aware sentiment analysis.

The study will evaluate the strengths and limitations of each approach addressing potential biases and areas that need to be improved. By tackling these challenges directly, the research aims to advance sentiment analysis and its use in the sharing economy. The outcomes are expected to deepen the understanding of Airbnb guest experiences and guide the creation of more reliable sentiment analysis tools. Ultimately, this study strives to enhance the application of sentiment analysis in consumer behaviour, business strategies, and policymaking within the evolving sharing economy.

Research Design

The research design for this study is structured to systematically analyse and forecast the sentiments expressed in guest reviews for apartment-type Airbnb accommodations in London as shown in Figure 1. The design incorporates both quantitative and qualitative methods to ensure a comprehensive understanding of the data and provide a holistic view of guest sentiments. The quantitative aspect involves the use of sentiment analysis tools to classify and quantify sentiments expressed in the reviews. The qualitative aspect includes the application of Named Entity Recognition (NER) and topic modelling to uncover deeper insights into the themes and entities discussed in the reviews. Correlation analysis examines relationships between market factors such as pricing and occupancy rates, while seasonal decomposition assesses the influence of market dynamics on sentiment trends. Predictive modelling, including ARIMA and SARIMAX, forecasts pricing and demand trends.

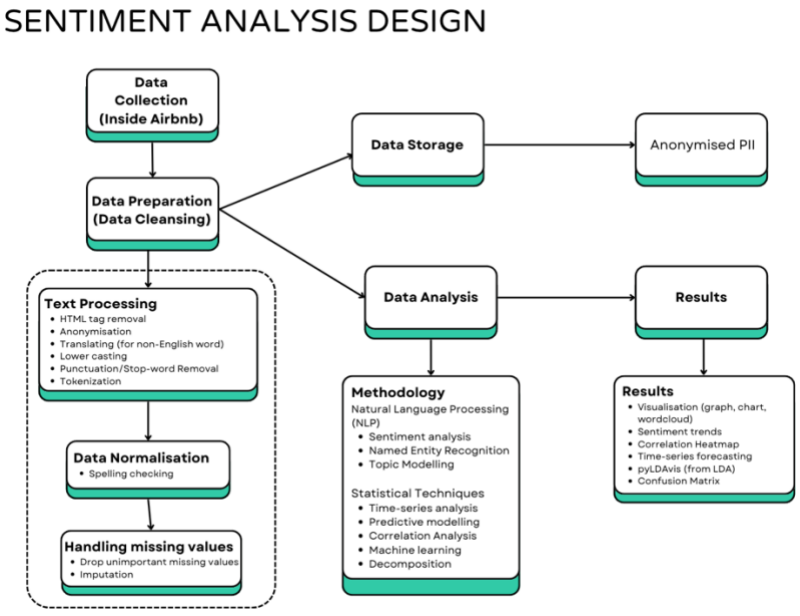


Figure 1. Sentiment Analysis research design

Methodology

Key Theories and Concepts

The selection of methods is a crucial decision in research design, as there are numerous methodologies available for conducting sentiment analysis, each with its own strengths and weaknesses. TextBlob, a versatile Python library known for its user-friendly interface, offers a comprehensive suite of NLP capabilities, including sentiment analysis, part-of-speech tagging, and noun phrase extraction (Loria, 2018). Its sentiment analysis module operates on a lexicon-based approach assigning polarity scores to words and phrases based on predefined dictionaries. This approach is computationally efficient and provides a general assessment of sentiment making it a popular choice for exploratory analysis and tasks where a broad overview of sentiment is sufficient. However, TextBlob's lexicon-based approach may struggle with the nuances of language, such as sarcasm, irony and domain-specific jargon (Kiritchenko et al., 2014), potentially leading to less accurate results in specialised contexts like Airbnb reviews, where guests often employ colloquial language and figurative expressions.

In contrast, VADER (Valence Aware Dictionary and Sentiment Reasoner) is specifically designed for social media text, incorporating slang, abbreviations and emoticons in its lexicon (Hutto & Gilbert, 2014). Its grammatical rules account for linguistic nuances like negation and intensifiers make it adept at analysing the informal, expressive text typical of Airbnb reviews. However, while VADER excels at capturing general sentiment trends, it may not fully address complex syntactic structures found in longer review texts, as noted in studies comparing lexicon-based and deep learning approaches (Hutto & Gilbert, 2014; Devlin et al., 2019; Liu et al., 2021).

In recent years, advanced machine learning models such as BERT (Bidirectional Encoder Representations from Transformers) have revolutionised the field of Natural Language Processing (NLP) by enabling more context-aware sentiment analysis. BERT utilises a transformer architecture to capture the bidirectional context of words within a sentence, thus providing a deeper understanding of language compared to traditional models (Devlin et al., 2019). DistilBERT, a distilled and more efficient version of BERT, retains much of the original model's performance while being significantly lighter and faster. This computational efficiency makes DistilBERT particularly suitable for large-scale sentiment analysis tasks (Sanh et al., 2019). By leveraging the capabilities of DistilBERT, this study aims to achieve state-of-the-art results in sentiment analysis, capturing the contextual relationships between words and phrases with greater precision.

Named Entity Recognition (NER) is used to identify and classify entities within the text, such as locations, people, and organisations. This technique allows for a more targeted analysis of specific aspects of the accommodation or host that guests frequently mention, providing insights into the factors that most significantly influence guest satisfaction (Nadeau & Sekine, 2007). By identifying entities that are commonly referenced in guest reviews, NER facilitates a deeper understanding of the guest experience.

Latent Dirichlet Allocation (LDA) is employed for topic modelling to uncover latent themes and topics within the reviews revealing patterns and concerns that may not be immediately obvious through sentiment analysis alone which can be written in equation as Table 1 (Blei, Ng, & Jordan, 2003). This approach provides a deeper understanding of the broader themes that shape guest experiences, such as cleanliness, location, and amenities.

Table 1. LDA model equation (Blei et al., 2003)

LDA model equation	Description
$P(w\|\alpha,\beta)=\Sigma_{z=1}^{k} P(w\|z,\beta)\cdot P(z\|d,\alpha)$ Where: **K** is the number of topics **w** is the word from documents **P(w|z,β)** is the probability of word w given topic z and the word distribution β **P(z|d,α)** is the probability of topic z given document d and the topic distribution α	**Topic Distribution P(z|d,α):** Represents the proportion of topics within a document d. **Word Distribution P(w|z,β)**: Represents the distribution of words within a topic z.

Logistic Regression is a widely used statistical model that predicts the probability of a categorical outcome based on one or more predictor variables. This model's simplicity and interpretability make it an effective choice for analysing the relationships between sentiment scores and various features (Hosmer et al., 2013). Despite its advantages, logistic regression may struggle with imbalanced datasets, a common issue in sentiment analysis where positive reviews often outnumber negative ones. To address this, techniques such as SMOTEENN (Synthetic Minority Over-sampling Technique and Edited Nearest Neighbours) are utilised. SMOTEENN combines over-sampling of minority classes with under-sampling of majority classes to balance the dataset enhancing the model's ability to accurately predict sentiments across all classes (Shukla, 2020). An example of the effective use of SMOTEENN in combination with advanced machine learning models, such as XGBoost, is demonstrated by Gustiyani Zainal, who applied these techniques to bank marketing data, showcasing their potential to improve classification accuracy in imbalanced datasets (Zainal, 2020).

Table 2. Statistical model equation

Statistic Model Equation	Description
Logistic Regression (Hosmer et al., 2013) $P(Y=1 \mid X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}}$	where: **P(Y=1|X)** is the probability of the outcome being a positive class $\boldsymbol{\beta_0}$ is the intercept of the model $\boldsymbol{\beta_1\beta_2\ldots\beta_n}$ are the coefficients corresponding to each predictor $\boldsymbol{X_1X_2\ldots X_n}$ $\boldsymbol{X_1X_2\ldots X_n}$ are the predictor variables
XGBoost : Extreme Gradient Boosting (Chen & Guestrin, 2016) $\text{Obj}(\theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$	where: $\boldsymbol{L_{(\gamma_i,\gamma_i\hat{})}}$ is the loss function that measures the difference between the predicted value $\gamma_i\hat{}$ and the actual value γ_i, $\boldsymbol{\Omega(f_k)}$ is the regularization term for the complexity of the model, which helps to prevent overfitting by penalizing more complex models. **n** is the number of instances in the training set. **K** is the number of trees (or base learners) in the model.

Time-series analysis techniques, including ARIMA and SARIMAX, are utilised to analyse sentiment trends over time and forecast future trends in pricing and demand. By incorporating sentiment scores as exogenous variables, these models provide enhanced predictive accuracy offering insights into the predictive power of sentiment analysis and its application in understanding market dynamics (Box et al., 2015; He et al., 2021).

Table 3. Time-series model equation

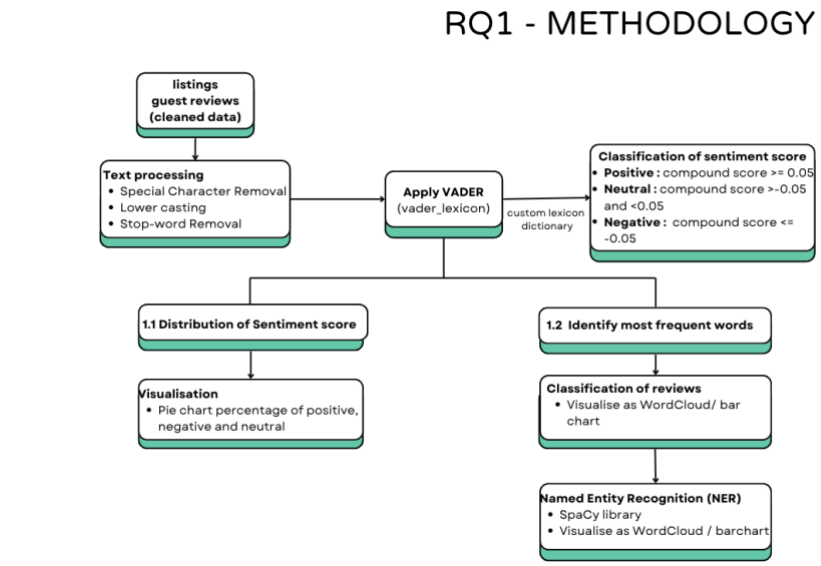
Time-series Model Equation	Description
ARIMA (Box et al., 2008) $y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t$ The integrated component involves differencing the series to remove trends and make it stationary, so the model becomes: $y'_t = y_t - y_{t-1}$ For an ARIMA model with parameters (p, d, q) where: • **p** is the number of lag observations in the model (autoregressive order) (y_t) • **d** is the number of times that the raw observations are differenced to make the time series stationary (differencing order) • **q** is the size of the moving average window (moving average order), determines the number of lagged forecast errors (ε_t)	where: $\boldsymbol{y_t}$ is the actual value at time t $\boldsymbol{\phi}$ are the coefficients for the autoregressive terms $\boldsymbol{\varepsilon_t}$ is the error at time t $\boldsymbol{\theta}$ are the coefficients for the moving average terms
SARIMAX (Hyndman & Athanasopoulos, 2018) $Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} + \beta_1 X_{1,t} + \beta_2$ For SARIMAX model, (p, d, q) are same with ARIMA but include exogenous variables which are external predictors that can affect time series	where: $\boldsymbol{Y_t}$ is the time series value at time t $\boldsymbol{X_{1,t}, X_{2,t}...}$ are the exogenous variables (independent predictors) at time t $\boldsymbol{\beta_1, \beta_{2..}}$ are the coefficients for the exogenous variables.

Data Analysis Procedures

The data analysis involves several steps to ensure a comprehensive understanding of the sentiments expressed in the reviews.

Quantitative analysis

The quantitative analysis utilises sentiment analysis to evaluate attitudes, opinions, and emotions expressed in online Airbnb reviews. Initially, a comparison was conducted between two sentiment analysis tools, TextBlob and VADER, to identify which model better suited the data. Following this comparison, VADER was selected due to its superior handling of nuanced expressions in the reviews. To further enhance its accuracy in sentiment classification, a custom lexicon (Suresh, 2019) tailored to Airbnb-specific terms was developed. Additionally, thresholds of 0.05 and -0.05 were chosen to classify texts as positive or negative only when a clear inclination was present, while texts with compound scores close to zero were classified as neutral (Britzolakis et al., 2020). This customised version of VADER was then used for a detailed analysis allowing for a more accurate assessment of guest sentiment. The results include a distribution of sentiment presented as a pie chart, along with word clouds representing positive and negative reviews, and a list of the most frequently mentioned terms as illustrated in Figure 2.



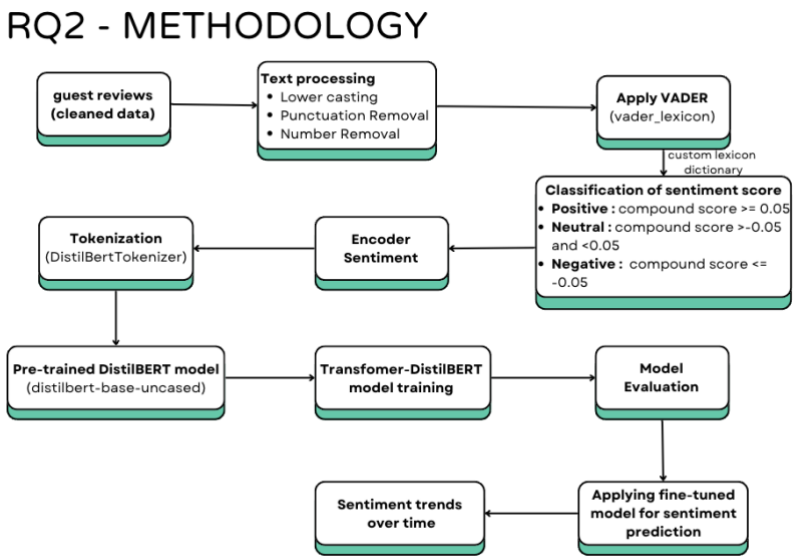
Import library

```
import pandas as pd
import string
from collections import Counter
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS
from textblob import TextBlob
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score, confusion_matrix
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix
import nltk
from nltk.tokenize import word_tokenize
```

```
# Custom lexicon dictionary
custom_lexicon = {
    "good": 2.0, "bad": -2.0, "excellent": 3.0, "poor": -3.0, "amazing": 3.0, "awesome": 3.0, "fantastic": 3.0,
    "great": 2.5, "wonderful": 3.0,"love": 3.0, "perfect": 3.0, "superb": 3.0, "terrific": 3.0,"satisfactory": 1.5,
    "delightful": 2.5, "happy": 2.0, "pleased": 2.0, "excellent service": 3.5, "highly recommend": 3.5,
    "top-notch": 3.0, "terrible": -3.0, "awful": -3.0, "horrible": -3.0, "disgusting": -3.0,
    "hate": -3.0, "unacceptable": -2.5, "disappointing": -2.0, "poor service": -3.5, "not recommend": -3.0,
    "waste": -2.5, "regret": -2.5, "bad experience": -3.0, "never again": -3.0, "frustrating": -2.0,
    "worst": -3.0, "average": 0.0, "ok": 0.0,"mediocre": -0.5, "sufficient": 0.5,
}

# Function to update VADER lexicon
def update_vader_lexicon():
    vader_lexicon = SentimentIntensityAnalyzer().lexicon
    vader_lexicon.update(custom_lexicon)
    return SentimentIntensityAnalyzer()

# Initialize the updated VADER sentiment analyzer
sid = update_vader_lexicon()

# Function to classify comments as positive, negative, or neutral using VADER
def classify_sentiment_vader(text):
    score = sid.polarity_scores(text)
    if score['compound'] > 0.05:
        return 'positive'
    elif score['compound'] < -0.05:
        return 'negative'
    else:
        return 'neutral'

# Apply the sentiment analysis to the reviews DataFrame
language_custom_2['sentiment'] = language_custom_2['processed_comments'].apply(classify_sentiment_vader)
```

Figure 2. Sentiment Analysis procedure and python programming for custom VADER

DistilBERT (Bidirectional Encoder Representations from Transformers), a distilled version of BERT (Hugging Face, n.d.), was used for time-series sentiment analysis to capture and analyse sentiments expressed in Airbnb guest reviews over time. This model enables the calculation of the proportion of each sentiment category. Reviews are segmented into relevant periods, such as pre-Brexit, during Brexit, pre-COVID-19, during COVID-19, and post-COVID-19, to assess how significant events impact guest sentiment as shown in Figure 3. DistilBERT's ability to efficiently capture contextual relationships between words and phrases efficiently, making it suitable for large-scale sentiment analysis tasks. This approach provides insights into how political and economic events influence guest experiences.



Import Library

```
[ ]  1 import pandas as pd
     2 import numpy as np
     3 from sklearn.model_selection import train_test_split
     4 from sklearn.preprocessing import LabelEncoder
     5 import torch
     6 from torch.utils.data import Dataset, DataLoader
     7 from transformers import DistilBertTokenizer, DistilBertForSequenceClassification, Trainer, TrainingArguments
     8 from nltk.sentiment.vader import SentimentIntensityAnalyzer
     9 import nltk
    10 import re
    11 import matplotlib.pyplot as plt
    12 from datetime import datetime
    13 from nltk.corpus import stopwords
    14 from wordcloud import WordCloud
    15 from collections import Counter
    16 from nltk.tokenize import word_tokenize
    17
    18 # Ensure necessary NLTK packages are downloaded
    19 nltk.download('vader_lexicon')
    20 nltk.download('stopwords')
    21 nltk.download('punkt')
```

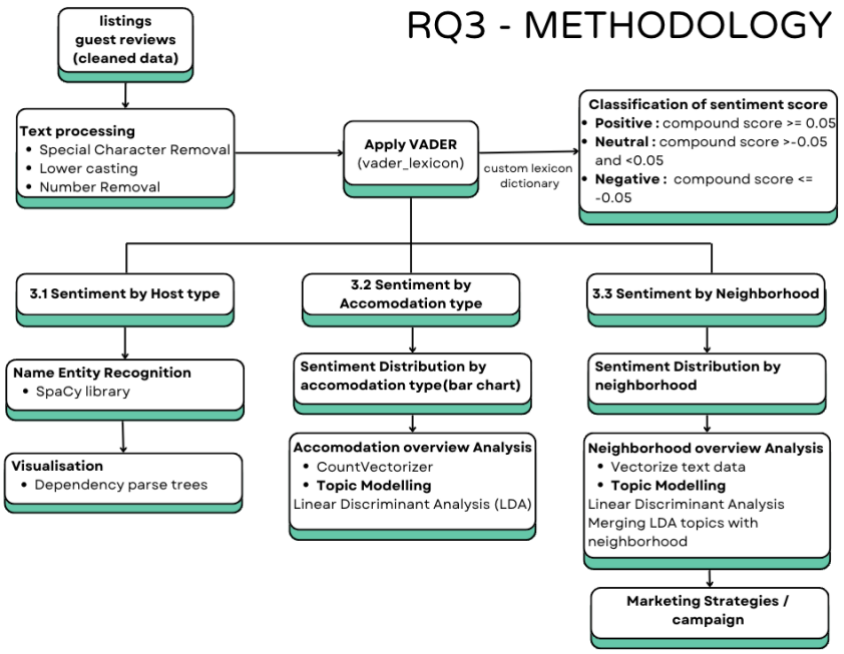
Key Parameters

```
# Create train and validation datasets
train_dataset = ReviewsDataset(train_texts, train_labels)
val_dataset = ReviewsDataset(val_texts, val_labels)

# Load pre-trained DistilBERT model
model = DistilBertForSequenceClassification.from_pretrained('distilbert-base-uncased', num_labels=3)

# Use GPU if available
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model.to(device)

# Define training arguments
training_args = TrainingArguments(
    output_dir='./results',          # output directory
    num_train_epochs=3,              # number of training epochs
    per_device_train_batch_size=16,  # batch size for training
    per_device_eval_batch_size=64,   # batch size for evaluation
    warmup_steps=500,                # number of warmup steps for learning rate scheduler
    weight_decay=0.01,               # strength of weight decay
    logging_dir='./logs',            # directory for storing logs
    logging_steps=10,
    evaluation_strategy="epoch",     # Evaluate every epoch
    save_strategy="epoch",           # Save every epoch
    load_best_model_at_end=True,     # Load the best model at the end of training
)

# Define Trainer with logging
trainer = Trainer(
    model=model,                         # the instantiated Transformers model to be trained
    args=training_args,                  # training arguments, defined above
    train_dataset=train_dataset,         # training dataset
    eval_dataset=val_dataset             # evaluation dataset
)
```

Figure 3. Sentiment analysis over time with DistilBERT model procedure and programming

Qualitative analysis

The qualitative analysis is divided into three parts to gain deeper insights into guest experiences. Firstly, sentiment distribution by host type employs NER (Named Entity Recognition) to identify and classify entities such as host names and locations. This method enables the analysis of patterns in guest perceptions of Super hosts compared to regular hosts revealing areas for service enhancement. Secondly, sentiment distribution by accommodation type utilises LDA (Latent Dirichlet Allocation) and topic modelling to identify recurring themes. This approach highlights differences in sentiment across various accommodation types providing insights into how hosts can align their offerings with guest preferences. Lastly, sentiment distribution by neighbourhood applies LDA and topic modelling to assess how geographical factors influence guest evaluations. All these implementation steps are as Figure 4 and 5.



Import Library

```
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import spacy
import matplotlib.pyplot as plt
import seaborn as sns
from gensim import corpora
from gensim.models import LdaModel
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis
from sklearn.feature_extraction.text import CountVectorizer
from collections import Counter
from IPython.display import display, HTML
```

Figure 4. Thematic Analysis with LDA and NER procedure

NER Implementation for sentiment by host type

```
# Apply NER to identify host mentions
def extract_hosts(text):
    doc = nlp(text)
    hosts = [ent.text for ent in doc.ents if ent.label_ == 'PERSON']
    return hosts

reviews_2['hosts'] = reviews_2['cleaned_comments'].apply(extract_hosts)
```

LDA Implementation for sentiment by accommodation

```
# Use bigrams and trigrams
bigram = Phrases(reviews_3['tokens'], min_count=5, threshold=100)
trigram = Phrases(bigram[reviews_3['tokens']], threshold=100)

bigram_mod = Phraser(bigram)
trigram_mod = Phraser(trigram)

def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]

def make_trigrams(texts):
    return [trigram_mod[bigram_mod[doc]] for doc in texts]

reviews_3['bigrams'] = make_bigrams(reviews_3['tokens'])
reviews_3['trigrams'] = make_trigrams(reviews_3['tokens'])

# Use trigrams for topic modeling
tokenized_docs = reviews_3['trigrams']

# Create a dictionary representation of the documents
dictionary = Dictionary(tokenized_docs)

# Filter extremes
dictionary.filter_extremes(no_below=10, no_above=0.5)

# Create a corpus from the dictionary representation
corpus = [dictionary.doc2bow(doc) for doc in tokenized_docs]

# Fit LDA model to identify latent topics
lda_model = LdaModel(corpus, num_topics=5, id2word=dictionary, passes=20, random_state=0)
```

```
# Assign the dominant topic to each review
merged_data['topic'] = [max(lda_model[corpus[i]], key=lambda x: x[1])[0] for i in range(len(corpus))]

# Group data by property type and topic
accommodation_topics = merged_data.groupby(['property_type', 'topic']).size().unstack().fillna(0)
```

LDA Implementation for sentiment by neighbourhood

```
# Vectorize the text data
vectorizer = CountVectorizer(stop_words='english')
neighbourhood_overview_matrix = vectorizer.fit_transform(listings_1['neighborhood_overview'])

# Apply LDA to extract topics
lda = LatentDirichletAllocation(n_components=5, random_state=0)
lda.fit(neighbourhood_overview_matrix)

# Display the topics
def display_topics(model, feature_names, num_top_words):
    topics = []
    for topic_idx, topic in enumerate(model.components_):
        topic_words = [feature_names[i] for i in topic.argsort()[:-num_top_words - 1:-1]]
        topics.append(topic_words)
        print(f"Topic {topic_idx}: {' '.join(topic_words)}")
    return topics

num_top_words = 10
feature_names = vectorizer.get_feature_names_out()
topics = display_topics(lda, feature_names, num_top_words)

# Merge LDA topics with neighbourhoods
topics_matrix = lda.transform(neighbourhood_overview_matrix)
listings_1['topic'] = topics_matrix.argmax(axis=1)
```

Figure 5. Implementation coding for Thematic Analysis

Correlation Analysis

Correlation analysis as Figure 6 involves statistical methods, including chi-square tests and machine learning approaches, which are used to analyse correlations between sentiment scores and apartment features such as cleanliness, location, and amenities. This approach helps identify significant relationships and patterns within the data providing insights into how specific features impact guest sentiments.

To evaluate the performance of the machine learning models, a confusion matrix (Berrajaa, 2022) is utilised, as shown in Table 4. The confusion matrix provides a detailed breakdown of the model's predictions and evaluation metric as Table 5 allows for a comprehensive assessment of the model offering valuable insights into its effectiveness in capturing correlations between sentiment scores and apartment features.

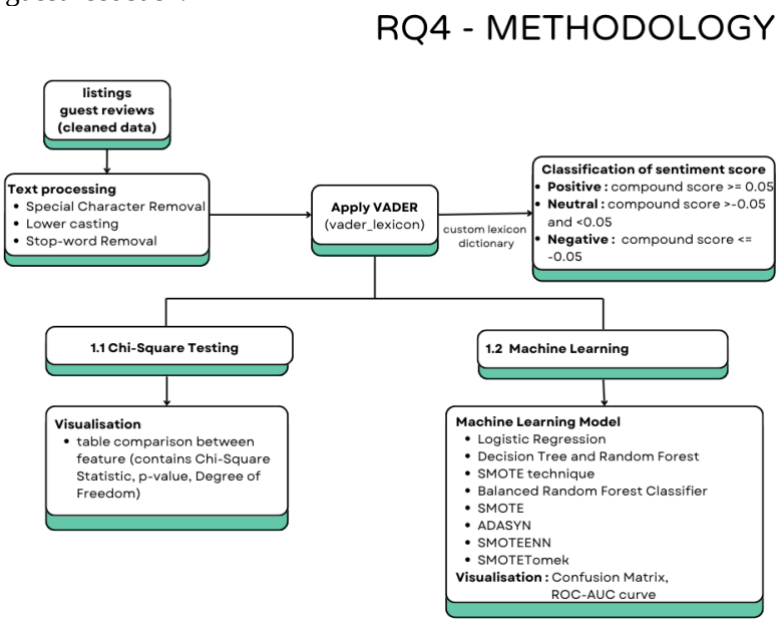
Table 4. Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Table 5. Evaluation metrics and equation

Evaluation Metrics	Equation
Accuracy	$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$
Precision	$\text{Precision} = \frac{TP}{TP+FP}$
Recall	$\text{Recall} = \frac{TP}{TP+FN}$
F1-score	$\text{F1 - measure} = \frac{2*Recall*precision}{recall+precision}$

Logistic regression is utilised as a baseline model to predict sentiment based on various features of the listings. To improve predictive accuracy and find the best-fit model, various machine learning techniques are applied as shown in Figure 6. The models are evaluated using the ROC-AUC curve (Narkhede, 2018) as Table 6, which measures their ability to accurately classify sentiment and identify correlations between listing features and guest feedback.



Logistic Regression

```
from sklearn.metrics
import classification_report, confusion_matrix, ConfusionMatrixDisplay
```

```
# Standardize features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.2, random_state=42)

# Logistic regression model
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
```

This coding is based on logistic regression equation where :

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}}$$

X_1 , X_2 ... are the predictor variables (e.g., average_price, review_scores_location, review_scores_value)

SMOTEENN + XGBoost Model

SMOTEENN : Synthetic Minority Over-sampling Technique + Edited Nearest Neighbors	
SMOTE is applied to create synthetic samples of the minority class, thereby increasing the representation of the minority class.	**ENN** is applied to the resulting dataset. ENN removes noisy or borderline examples from both the majority and minority classes, which helps in cleaning the dataset.

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features_scaled, target_mapped, test_size=0.2, random_state=42)

# Apply SMOTEENN to the training data
smoteenn = SMOTEENN(random_state=42)
X_train_smoteenn, y_train_smoteenn = smoteenn.fit_resample(X_train, y_train)

# Train the XGBoost model with best parameters
xgb_smoteenn = XGBClassifier(
    random_state=42,
    colsample_bytree=0.9477831645730916,
    learning_rate=0.1758679267213984,
    max_depth=9,
    n_estimators=423,
    subsample=0.5808533263327152,
    use_label_encoder=False,
    eval_metric='mlogloss'
)
xgb_smoteenn.fit(X_train_smoteenn, y_train_smoteenn)

# Predictions
y_pred_train_smoteenn = xgb_smoteenn.predict(X_train_smoteenn)
y_pred_test_smoteenn = xgb_smoteenn.predict(X_test)
```

Remark: The parameters have been obtained from the Gustiyan Islahuzaman (2023) study.

Figure 6. [RQ4] Correlation analysis procedure and important coding implementation

Table 6. ROC-AUC Analysis and Implementation

ROC-AUC Analysis	Python Implementation

True Positive Rate (TPR) $= \frac{TP}{TP+FN}$
False Positive Rate (FPR) $= \frac{FP}{FP+TN}$
 	
AUC value ranges from 0 to 1:
 	
AUC = 1: Perfect model.
AUC = 0.5: Model has no discriminative ability, equivalent to random guessing.
AUC < 0.5: Model performs worse than random guessing, indicating poor model performance.
 	
(Narkhede, 2018)
	`from sklearn.metrics import roc_auc_score, roc_curve, auc, RocCurveDisplay`
	
	`# Define a function to plot ROC-AUC curves for multi-class classification`
	`def plot_roc_auc_multiclass(models, X_test, y_test, class_names):`
	`  plt.figure(figsize=(12, 8))`
	
	`  # Binarize the test labels for each class`
	`  y_test_binarized = label_binarize(y_test, classes=[0, 1, 2])`
	
	`  for model, label in models:`
	`  y_prob = model.predict_proba(X_test)`
	`  for i in range(len(class_names)):`
	`  fpr, tpr, _ = roc_curve(y_test_binarized[:, i], y_prob[:, i])`
	`  auc_score = auc(fpr, tpr)`
	`  plt.plot(fpr, tpr, label=f'{label} (class {class_names[i]}) ROC-AUC = {auc_score:.2f}')`
	
	`  plt.title('Multi-Class ROC-AUC Curve Comparison')`
	`  plt.xlabel('False Positive Rate')`
	`  plt.ylabel('True Positive Rate')`
	`  plt.legend(loc='best')`
	`  plt.grid(True)`
	`  plt.show()`

Airbnb Market Dynamics analysis

To explore how the growth and dynamics of the Airbnb market in London influence guest sentiments in apartment-type accommodations, various analytical approaches are employed. This involves correlation analysis, occupancy rate calculations, and seasonal decomposition to comprehensively examine the factors impacting guest perceptions. Firstly, correlation analysis is utilised to identify relationships between market variables such as the number of listings, average pricing, occupancy rates, and sentiment scores. A correlation heatmap is generated to visually depict the strength and direction of these relationships.

Secondly, occupancy rates are calculated using the formula as below

$$\text{Occupancy Rate (\%)} = \frac{Number\ of\ booked\ nights}{Total\ available\ nights} * 100$$

to assess demand patterns and booking behaviours over specified time periods. Lastly, by performing seasonal decomposition on sentiment scores, the analysis separates trend, seasonal, and residual components which illustrate how significant events and seasonal variations influence guest perceptions, as depicted in Figure 7.

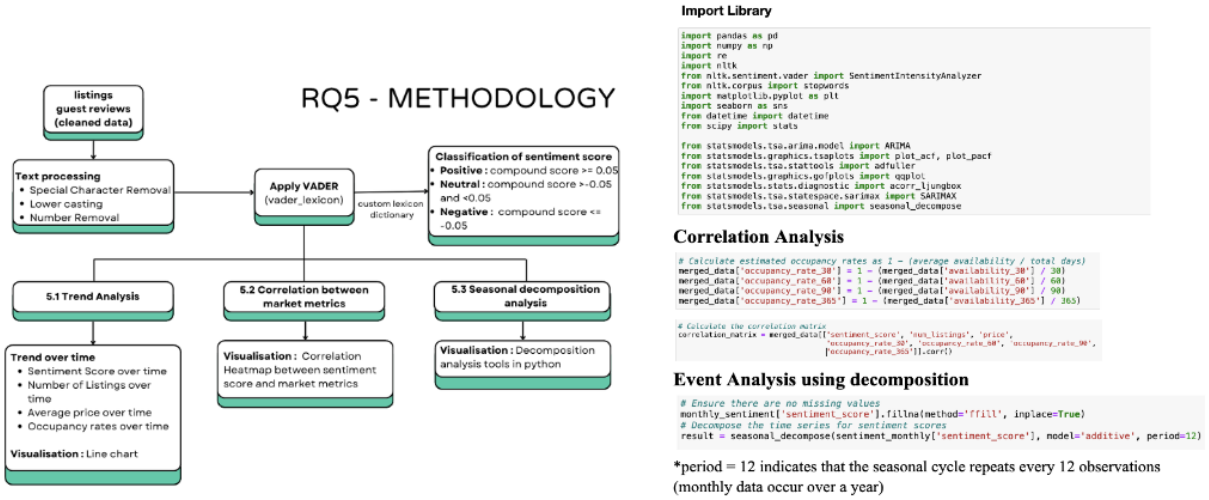


Figure 7. Market growth and dynamics analysis procedure

Predictive Modelling

Time-series models, including ARIMA and SARIMAX, are used to forecast future trends in pricing and demand based on sentiment analysis. Additionally, a HistGradientBoostingRegressor was employed alongside the ARIMA model as Figure 9 and Table 7 and 8 for pricing, while the SARIMAX model as Figure 10 and Table 9 integrates exogenous factors for demand forecasting. The models are evaluated using the performance metrics and model criteria as Figure 8. This comprehensive approach ensures that both temporal patterns and feature-based influences on sentiment are thoroughly analysed and leveraged for forecasting.

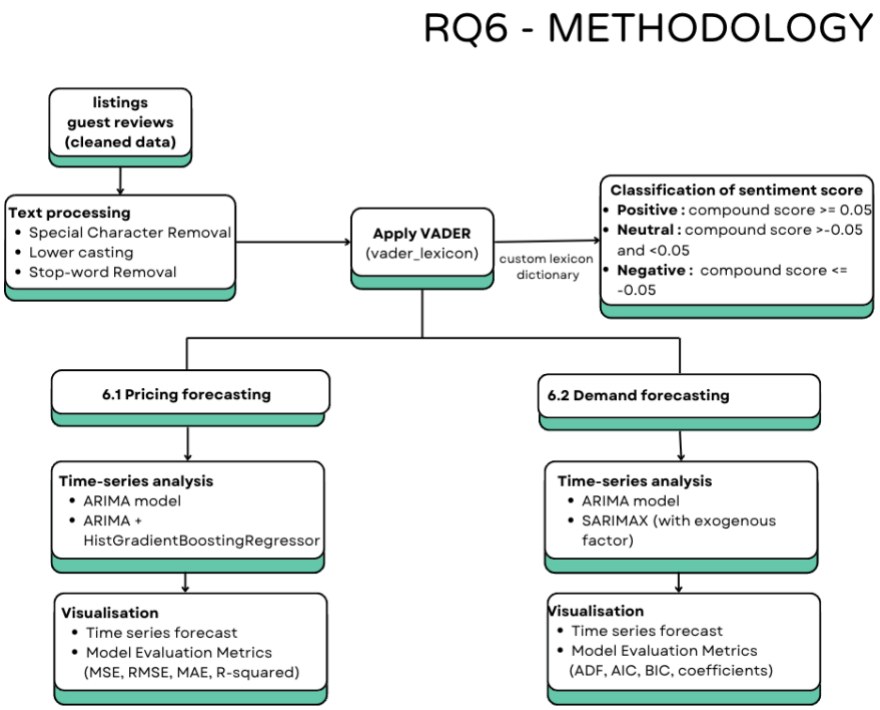


Figure 8. [RQ6] Forecasting time-series modelling for pricing and demand procedure

Import Library

```
import pandas as pd
import numpy as np
import re
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.corpus import stopwords
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller
from scipy import stats
from statsmodels.graphics.gofplots import qqplot
from statsmodels.stats.diagnostic import acorr_ljungbox
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingRegressor
```

ARIMA model

```
# Fit ARIMA model
model = ARIMA(ts_data['average_price'], order=(1, 1, 1))
model_fit = model.fit()

# Forecast future values
forecast_steps = 12  # Forecast for the next 12 months
forecast = model_fit.forecast(steps=forecast_steps)
```

ARIMA + HistGradientBoostingRegressor

```
# Set a random seed for reproducibility
np.random.seed(42)

# Add lagged features
ts_data['price_lag1'] = ts_data['average_price'].shift(1)
ts_data['price_lag2'] = ts_data['average_price'].shift(2)
ts_data['sentiment_lag1'] = ts_data['sentiment_score'].shift(1)
ts_data.dropna(inplace=True)

# Split the data
X = ts_data[['sentiment_score', 'price_lag1', 'price_lag2', 'sentiment_lag1']]
y = ts_data['average_price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the regression model
model_combined = HistGradientBoostingRegressor()
model_combined.fit(X_train, y_train)

# Predict
y_pred_combined = model_combined.predict(X_test)

# Fit ARIMA model
arima_order = (1, 1, 1)  # Set your chosen ARIMA order
model_arima = ARIMA(ts_data['average_price'], order=arima_order)
model_fit_arima = model_arima.fit()

# Forecast future values with ARIMA
forecast_steps = 12  # Forecast for the next 12 months
forecast_arima = model_fit_arima.forecast(steps=forecast_steps)

# Create a DataFrame for the ARIMA forecast
forecast_index_arima = pd.date_range(start=ts_data.index[-1] + pd.DateOffset(months=1),
                                     periods=forecast_steps, freq='MS')
forecast_df_arima = pd.DataFrame({'forecast': forecast_arima}, index=forecast_index_arima)

# Create test set for ARIMA from the last 12 months
test_arima_start = len(ts_data) - forecast_steps
test_arima_end = len(ts_data) - 1

# Predict using ARIMA
arima_predictions = model_fit_arima.predict(start=test_arima_start, end=test_arima_end)

# Mocking sentiment data for future predictions with the combined model
future_features = pd.DataFrame({
    'sentiment_score': np.random.normal(loc=0.5, scale=0.1, size=forecast_steps),  # Mock sentiment data
    'price_lag1': forecast_arima.shift(1).fillna(ts_data['average_price'].iloc[-1]),
    'price_lag2': forecast_arima.shift(2).fillna(ts_data['average_price'].iloc[-2]),
    'sentiment_lag1': np.random.normal(loc=0.5, scale=0.1, size=forecast_steps)  # Mock sentiment data
}, index=forecast_index_arima)

# Predict with the combined model for the next 12 months
future_pred_combined = model_combined.predict(future_features)
```

Figure 9. Python Programming for pricing forecasting

Import Library

```
import pandas as pd
import numpy as np
import re
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.corpus import stopwords
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller
from scipy import stats
from statsmodels.graphics.gofplots import qqplot
from statsmodels.stats.diagnostic import acorr_ljungbox
from statsmodels.tsa.statespace.sarimax import SARIMAX
```

ARIMA model

```
# Fit the ARIMA model
model = ARIMA(monthly_reviews, order=(1, 1, 1))
model_fit = model.fit()

# Forecast the next 6 months
forecast_steps = 6
forecast = model_fit.get_forecast(steps=forecast_steps)
forecast_index = pd.date_range(start=monthly_reviews.index[-1] + pd.DateOffset(months=1),
                               periods=forecast_steps, freq='M')

# Get the forecasted mean and confidence intervals
forecast_mean = forecast.predicted_mean
forecast_conf_int = forecast.conf_int()
```

SARIMAX Model

```
# Define the SARIMAX model
sarimax_model = SARIMAX(combined_df['review_count'],
                        exog=combined_df['sentiment_score'],
                        order=(2, 1, 2), seasonal_order=(1, 1, 1, 12))

# Fit the model
sarimax_fit = sarimax_model.fit()
```

Remark : This coding is based on the SARIMAX equation where:

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} + \beta_1 X_{1,t} + \beta_2$$

X_1 is exogenous variable (sentiment scores)

Model Order p = 2 (using the past two period) d = 1 (first-order differencing) q = 2 (past two period error prediction)
Seasonal Order P = 1 (model uses the value from the same season in the previous cycle) D = 1 (seasonal differencing is applied once) Q = 1 (one lag forecast error) s = 12 (12 period of season)

Figure 10. Python Programming for demand forecasting

Table 7. ARIMA model and parameter selection

Parameter	Description	Selection Method	Justification
p	Auto-Regressive order: Number of lag observations included in the model.	Selected based on the Partial Autocorrelation Function (PACF) plot.	Set **p = 1** if the PACF shows a significant correlation at lag 1 and none thereafter.
d	Differencing order: Number of times the data have been different to achieve stationarity.	Chosen by examining the data for trends and using differencing to stabilise variance.	Set **d = 1** if the time series becomes stationary after the first differencing.
q	Moving Average order: Number of lagged forecast errors included in the model.	Determined using the Autocorrelation Function (ACF) plot.	Set **q = 1** if the ACF plot indicates a significant correlation at lag 1.
p = 1, d = 1, q = 1 were chosen based on PACF and ACF plot analysis which aim for simplicity and effectiveness in capturing the data's autocorrelation structure.			

Table 8. HistGradientBoostingRegressor : Feature Engineering and Model training

Feature	Description	Engineering process	Purpose
Sentiment score	The score indicates the sentiment of the review (positive or negative).	Calculated using sentiment analysis tools.	To capture the qualitative influence of guest sentiments on pricing and demand.
Price Lag (1, 2)	Past prices at lag 1 and lag 2.	Generated by shifting the 'average_price' series by 1 and 2 lags, respectively.	To incorporate the effect of recent past prices on current pricing predictions.
Sentiment Lag (1)	Past sentiment score at lag 1.	Generated by shifting the 'sentiment_score' series by 1 lag.	To include the influence of recent sentiment changes on future pricing trends.
The combined model uses HistGradientBoostingRegressor to predict prices by learning from historical data, sentiment scores, and their lagged values offering a nuanced understanding of market dynamics that go beyond linear dependencies.			

Table 9. SARIMAX Model: Seasonal and Exogenous Parameters setting

Parameter	Description	Selection Method	Setting
P	**Seasonal Auto-Regressive order:** Number of lag observations in the seasonal component.	Selected using the Seasonal PACF plot.	For monthly data with an annual cycle and a significant lag at 12, set **P = 1**.
D	**Seasonal Differencing order:** Number of times the seasonal data is different to achieve stationarity.	Chosen based on observing seasonal trends and applying seasonal differencing.	Set **D = 1** if the seasonal trend is removed after one seasonal differencing.
Q	**Seasonal Moving Average order:** Number of lagged forecast errors in the seasonal model.	Determined using the Seasonal ACF plot.	Set **Q = 1** if the seasonal ACF plot shows a significant correlation at lag 12.
s	**Seasonal cycle length:** Number of periods in a season.	Based on the frequency of the data and the length of the seasonal cycle.	For monthly data with an annual seasonality, **s = 12**.
X	**Exogenous Variables:** External variables that might influence the time series (e.g., sentiment scores).	Assessed through statistical tests (e.g., AIC, BIC) or model comparison metrics.	Sentiment scores were included to capture their impact on pricing trends, assessed for significance using AIC/BIC criteria.
PACF/ACF Plots: - PACF plots show the correlation of a time series with its own lagged values, which helps identify the number of autoregressive terms. - ACF plots display the correlation between the time series and lagged versions of itself, which is useful for identifying the number of moving average terms. **Seasonal Components and Exogenous Variables in SARIMAX**: Discuss the importance of capturing seasonality and external factors. Explain how including exogenous variables, like sentiment scores, can improve model accuracy by accounting for external influences on the dependent variable.			

Parameter	Description	Selection Method	Justification
p	Auto-Regressive order: Number of lag observations included in the model.	Selected based on the Partial Autocorrelation Function (PACF) plot.	Set **p = 2** Two autoregressive terms are used suggesting that the current value of the series is influenced by the last two observations.
d	Differencing order: Number of times the data have been different to achieve stationarity.	Chosen by examining the data for trends and using differencing to stabilise variance.	Set **d = 1** if the time series becomes stationary after the first differencing.
q	Moving Average order: Number of lagged forecast errors included in the model.	Determined using the Autocorrelation Function (ACF) plot.	Set **q = 2** Two moving average terms are used to account for the past two forecast errors.

Data Overview

This study utilises a comprehensive dataset comprising 33,169 reviews and 1,987 listings from 2010 to 2024. The data is sourced from Inside Airbnb, a reputable provider of extensive datasets on Airbnb listings and reviews (Inside Airbnb, n.d.).

Data Sources:

- **Reviews Data**: The primary data source consists of reviews from ‘reviews.csv,’ which correspond to listings classified as "apartment-type" in ‘listing.csv.’
- **Listings Data:** This data includes context, and features related to each review from ‘listings.csv’.

Table 10. Data dictionary of reviews

Field	Description
listing_id	Identifies the listing associated with each review.
id	Unique identifier for each review.
date	Date of the review, used for temporal analysis.
reviewer_id	Unique identifier for the reviewer.
reviewer_name	Name of the reviewer (PII - anonymized).
comments	Text of the review, used for sentiment analysis and NLP tasks.

Table 11. Data dictionary of listing

Field	Description
listing_id	Unique identifier for the listing.
name	Name of the listing.
description	Description of the listing
neighborhood_overview	Overview of the neighbourhood
host_id	Unique identifier for the host.
host_name	Name of the host (PII - anonymized).
host_since	Date since the host is active
host_is_superhost	Indicates if the host is a superhost (binary variable: 1 for superhost, 0 for normal host)
neighbourhood_cleansed	Neighbourhood where the listing is located.
latitude	Geographical coordinate, used for spatial analysis.
longitude	Geographical coordinate, used for spatial analysis.
property_type	Type of property
room_type	Type of accommodation (e.g., entire place, private room)
accommodates	Number of guests the listing can accommodate.
bathrooms	Number of bathrooms.
bathrooms_text	Text description of the bathrooms.
bedrooms	Number of bedrooms.
beds	Number of beds.
amenities	List of amenities provided in the listing.
price	Price per night for the listing, used in correlation and forecasting models.
minimum_nights	Minimum number of nights required to book.
maximum_nights	Maximum number of nights allowed to book.
availability_30	Availability of the listing in the next 30 days.
availability_60	Availability of the listing in the next 60 days.
availability_90	Availability of the listing in the next 90 days.
availability_365	Availability of the listing in the next 365 days.
number_of_reviews	Total number of reviews.
number_of_reviews_ltm	Number of reviews in the last 12 months.
number_of_reviews_l30d	Number of reviews in the last 30 days.
first_review	Date of the first review.
last_review	Date of the last review.
review_scores_rating	Overall rating score.
review_scores_accuracy	Accuracy rating score.
review_scores_cleanliness	Cleanliness rating score.
review_scores_checkin	Check-in rating score.
review_scores_communication	Communication rating score.
review_scores_location	Location rating score.
review_scores_value	Value rating score.
instant_bookable	Indicates if the listing is instantly bookable.
calculated_host_listings_count calculated_host_listings_count_entire_homes calculated_host_listings_count_private_rooms calculated_host_listings_count_shared_rooms	Total listings count for the host/host by accommodation type
reviews_per_month	Number of reviews per month.

Data Quality and Preprocessing

Some data from Inside Airbnb exhibits noise, inconsistencies, and missing values, which can potentially impact the accuracy of sentiment analysis. User-generated content often contains spelling errors, slang, and diverse grammatical structures posing challenges for data processing. To address these issues effectively, data cleansing will be implemented as Figure 11. This procedure will focus on identifying and rectifying inconsistencies, handling missing values, and standardising the text data to ensure it is clean and reliable for analysis.

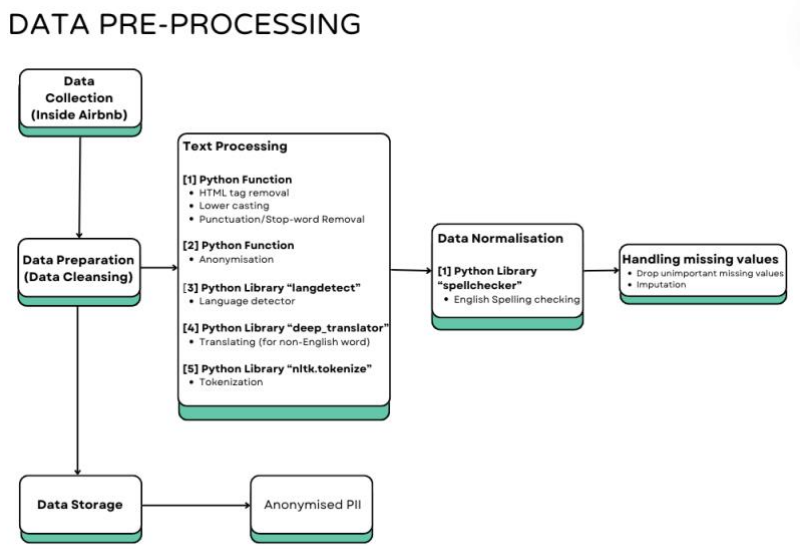


Figure 11. Data pre-processing for sentiment analysis

Result Analysis and Discussion

This section presents the findings of the study, which aimed to analyse guest sentiments expressed in Airbnb reviews for apartment-type accommodations in London. The results are organised according to the research questions outlined in the introduction.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to understand the structure and content of the dataset. These initial analyses provided valuable insights that guided the subsequent sentiment analysis and interpretation of results. This step includes examining the data types and column names in the dataset which includes 'reviews.csv' and 'listings.csv' as shown in and Figure 12 and Figure 13.

```
file_path = 'listings.csv'
df = pd.read_csv(file_path)

# Filter the dataframe to include only rows where 'property_type' contains 'apartment'
apartment_df = df[df['property_type'].str.contains('apartment', case=False, na=False)]

# Columns to drop
columns_to_drop = [
    'listing_url', 'scrape_id', 'last_scraped', 'source', 'picture_url',
    'host_url', 'host_thumbnail_url', 'host_picture_url', 'host_about',
    'host_response_time', 'host_response_rate', 'host_acceptance_rate',
    'host_neighbourhood', 'host_verifications', 'host_has_profile_pic',
    'host_identity_verified', 'neighbourhood_group_cleansed', 'calendar_updated',
    'calendar_last_scraped', 'minimum_minimum_nights', 'maximum_minimum_nights',
    'minimum_maximum_nights', 'maximum_maximum_nights', 'minimum_nights_avg_ntm',
    'maximum_nights_avg_ntm', 'host_listings_count', 'host_total_listings_count',
    'license','host_location','has_availability','neighbourhood'
]
apartment_df_cleaned = apartment_df.drop(columns=columns_to_drop)

# Rename 'id' column in apartment_data to 'listing_id' for consistency
apartment_df_cleaned.rename(columns={'id': 'listing_id'}, inplace=True)

print("Shape of the cleaned dataframe:", apartment_df_cleaned.shape)
print(apartment_df_cleaned.info())
```

```
Shape of the cleaned dataframe: (1987, 44)
```

```
file_path = 'reviews.csv'
reviews_df = pd.read_csv(file_path,low_memory=False )
apartment_ids = apartment_df['id']

# Filter the reviews dataframe to include only reviews for the filtered apartment listings
filtered_reviews_df = reviews_df[reviews_df['listing_id'].isin(apartment_ids)]

# Count the number of reviews for the filtered apartment listings
total_reviews = filtered_reviews_df.shape[0]

print(filtered_reviews_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 33169 entries, 8752 to 1618281
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   listing_id      33169 non-null  int64
 1   id              33169 non-null  int64
 2   date            33169 non-null  object
 3   reviewer_id     33169 non-null  int64
 4   reviewer_name   33169 non-null  object
 5   comments        33162 non-null  object
dtypes: int64(3), object(3)
memory usage: 1.8+ MB
None
```

Figure 12. listings and reviews file shape

```
<class 'pandas.core.frame.DataFrame'>
Index: 1987 entries, 64 to 90809
Data columns (total 44 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   listing_id                                    1987 non-null   int64
 1   name                                          1987 non-null   object
 2   description                                   1804 non-null   object
 3   neighborhood_overview                         1230 non-null   object
 4   host_id                                       1987 non-null   int64
 5   host_name                                     1987 non-null   object
 6   host_since                                    1987 non-null   object
 7   host_is_superhost                             1983 non-null   object
 8   neighbourhood_cleansed                        1987 non-null   object
 9   latitude                                      1987 non-null   float64
 10  longitude                                     1987 non-null   float64
 11  property_type                                 1987 non-null   object
 12  room_type                                     1987 non-null   object
 13  accommodates                                  1987 non-null   int64
 14  bathrooms                                     1715 non-null   float64
 15  bathrooms_text                                1975 non-null   object
 16  bedrooms                                      1972 non-null   float64
 17  beds                                          1713 non-null   float64
 18  amenities                                     1987 non-null   object
 19  price                                         1715 non-null   object
 20  minimum_nights                                1987 non-null   int64
 21  maximum_nights                                1987 non-null   int64
 22  availability_30                               1987 non-null   int64
 23  availability_60                               1987 non-null   int64
 24  availability_90                               1987 non-null   int64
 25  availability_365                              1987 non-null   int64
 26  number_of_reviews                             1987 non-null   int64
 27  number_of_reviews_ltm                         1987 non-null   int64
 28  number_of_reviews_l30d                        1987 non-null   int64
 29  first_review                                  1561 non-null   object
 30  last_review                                   1561 non-null   object
 31  review_scores_rating                          1561 non-null   float64
 32  review_scores_accuracy                        1561 non-null   float64
 33  review_scores_cleanliness                     1561 non-null   float64
 34  review_scores_checkin                         1561 non-null   float64
 35  review_scores_communication                   1561 non-null   float64
 36  review_scores_location                        1561 non-null   float64
 37  review_scores_value                           1561 non-null   float64
 38  instant_bookable                              1987 non-null   object
 39  calculated_host_listings_count                1987 non-null   int64
 40  calculated_host_listings_count_entire_homes   1987 non-null   int64
 41  calculated_host_listings_count_private_rooms  1987 non-null   int64
 42  calculated_host_listings_count_shared_rooms   1987 non-null   int64
 43  reviews_per_month                             1561 non-null   float64
dtypes: float64(13), int64(16), object(15)
memory usage: 698.6+ KB
```

Figure 13. reviews.csv information

A bar chart visualising the distribution of reviews languages reveals that 82.3% of the reviews are in English, while French is the most common language, accounting for 5.1% of the reviews. All non-English reviews are subjected to data preprocessing for translation to ensure consistency in the analysis.

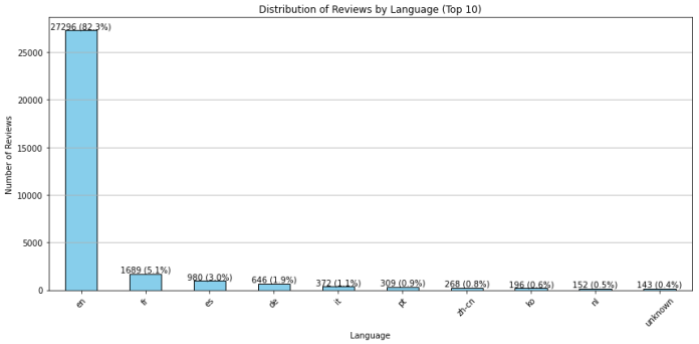


Figure 14. Distribution of reviews by language

A line chart as Figure 15 depicting the number of reviews over time shows a significant increase in review counts, particularly after 2018. This growth is attributed to the rising popularity of Airbnb among tourists. The chart highlights peaks corresponding to major events, such as COVID-19 pandemic, which affected travel patterns and review frequencies.

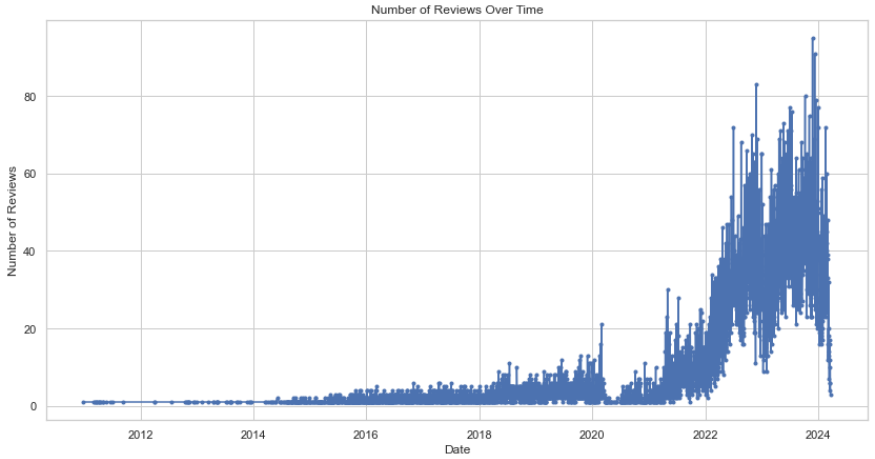


Figure 15. Number of reviews over time

The histograms as Figure 16 illustrate the distribution of reviews score across various features within the dataset, showing a right-skewed pattern, indicating that most reviews have high scores. This suggests that guests generally have positive experiences with Airbnb accommodations in London.

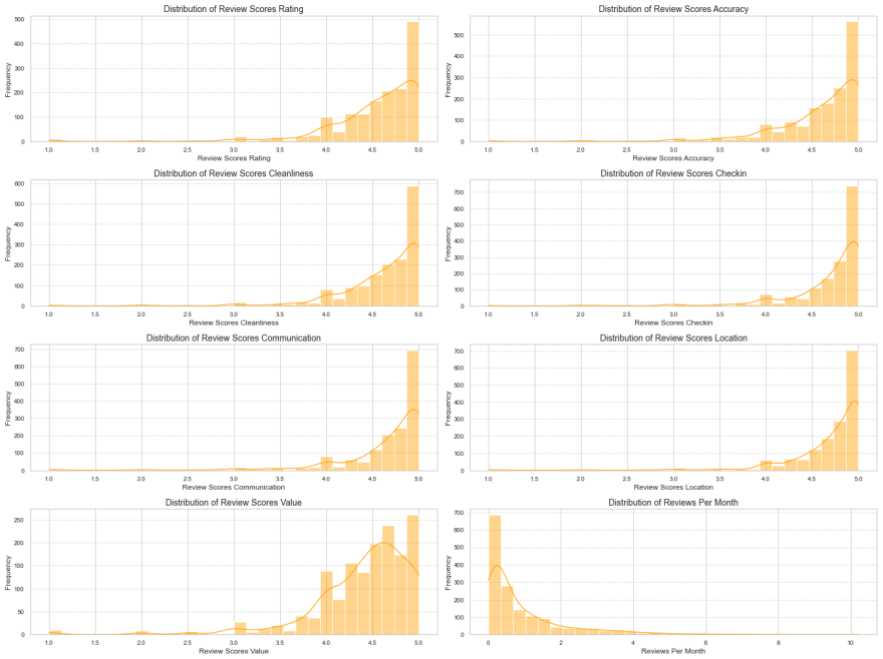


Figure 16. Distribution of reviews score across various features

An overall statistical summary of prices shows that the mean price is approximately $267.03 per night, and the median price is $194.00.

```
Overall Statistical Summary of Prices:
count     1715.000000
mean       267.029738
std        542.243922
min         36.000000
25%        134.000000
50%        194.000000
75%        295.000000
max      20000.000000
Name: price, dtype: float64

Overall Mean Price: $267.03
Overall Median Price: $194.00
```

Figure 17. Overall statistical summary of prices

A histogram as Figure 18 illustrating the price distribution of listings reveals that the average price distribution is skewed, with a higher concentration of listings in the lower price range.

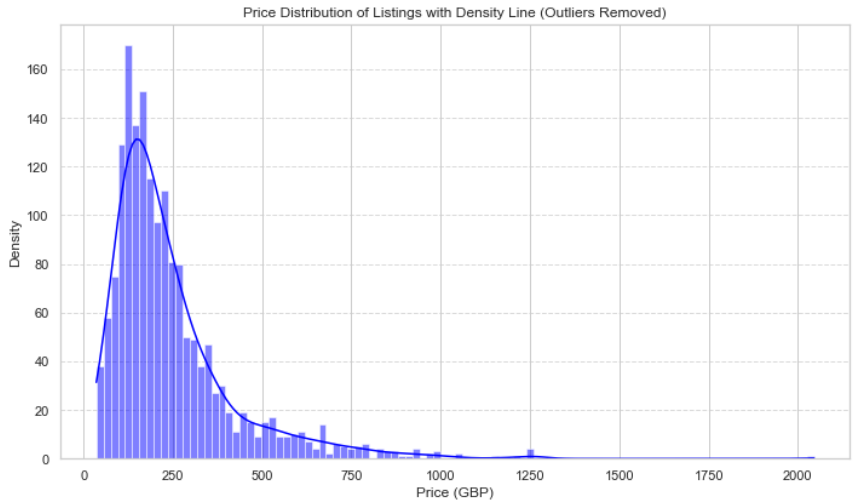


Figure 18. Price distribution of listings

A bar chart as Figure 19 visualises the average price by neighbourhood, highlighting significant price variations across different areas in London. Additionally, a box plot in Figure 20 provides further detail, showing the spread and distribution of listing prices within each neighbourhood. This combination of visualisations helps identify which neighbourhoods tend to have higher or lower listing prices offering a clearer understanding of the geographical pricing dynamics. Notably, neighbourhoods such as Kensington and Chelsea, and Westminster exhibit the highest prices considering the interquartile range.

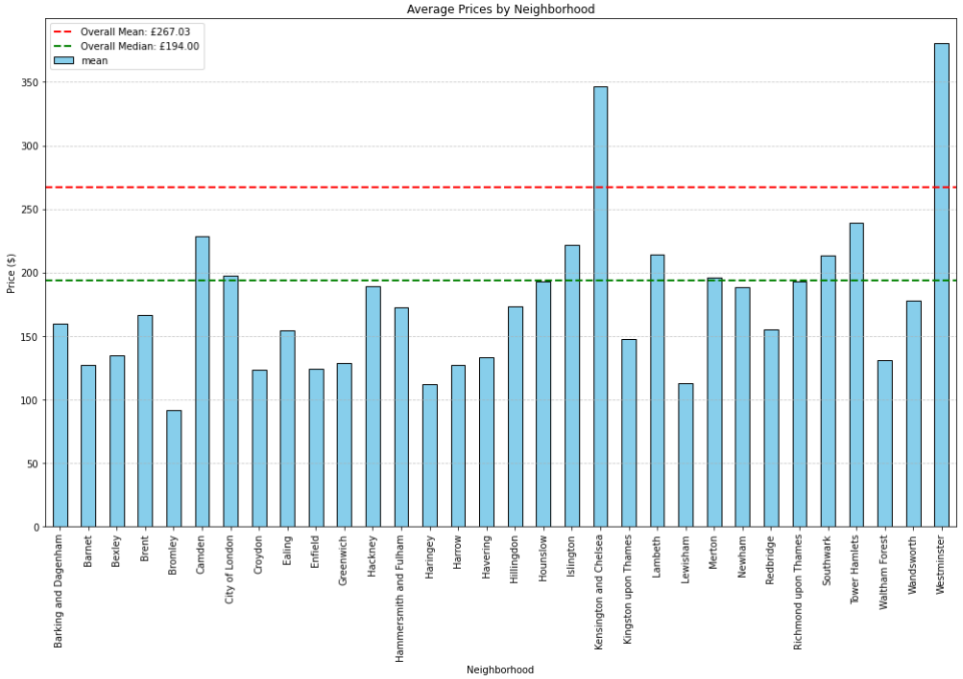


Figure 19. Average price by Neighbourhood

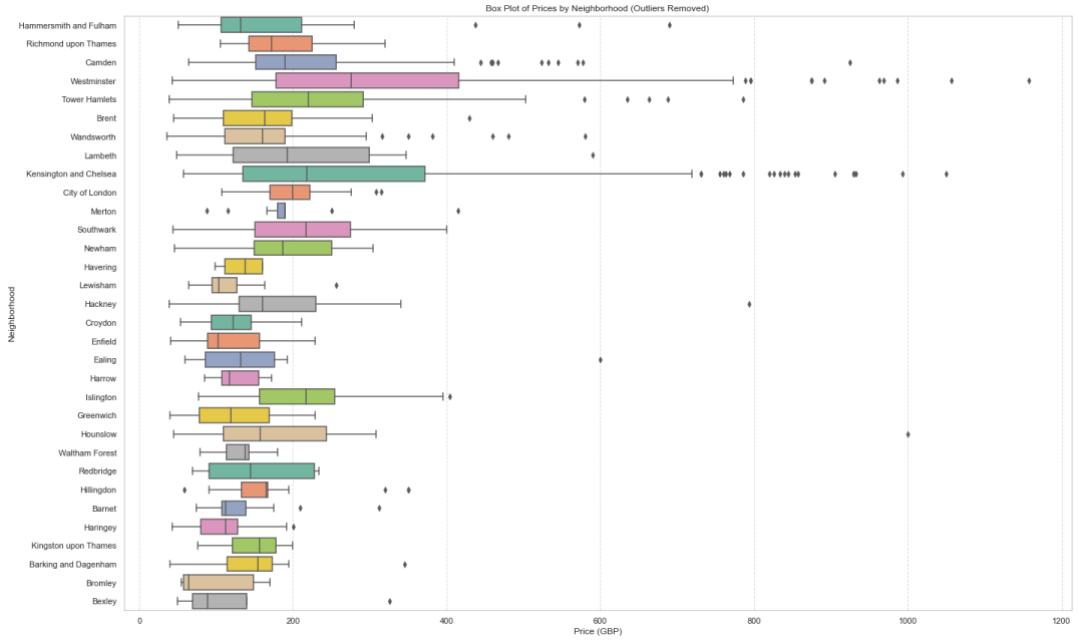


Figure 20. Box plot of prices by neighbourhood (outlier removed)

An analysis of the combinations of amenities provided by listings reveals that the top five most offered amenities include Wi-Fi, kitchen, heating, washer and TV. These amenities are highly favoured by guests reflecting common preference and essential features that enhance the guest experience.

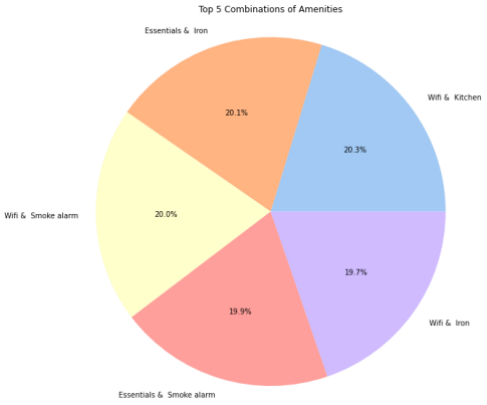


Figure 21. Top 5 combination of amenities

The average length of reviews shows a decreasing trend over time. As the line chart in Figure 22 indicates that as Airbnb becomes more popular, guests tend to leave shorter reviews, possibly due to the increased volume of reviews and the perceived effort required to leave detailed feedback.

Figure 22. Average review length over time

Sentiment Analysis

The objective of sentiment analysis conducted on Airbnb reviews is to classify the sentiments expressed by guests into positive, negative, and neutral categories. In this study, two principal sentiment analysis tools, VADER (Valence Aware Dictionary for sentiment Reasoning) and TextBlob (python library for processing textual data), were initially evaluated to determine the most effective method for sentiment classification.

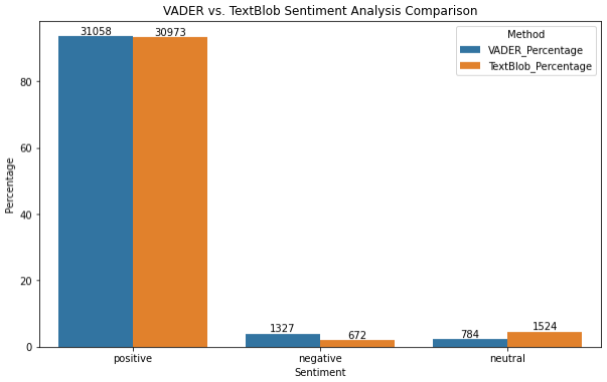


Figure 23. Comparison of sentiment analysis between VADER and TextBlob

Table 12. Comparison between VADER and TextBlob

Sentiment	VADER		TextBlob	
	Count	Percentage	Count	Percentage
Positive	31,058	93.64%	30,973	93.38%
Negative	1,327	4.00%	672	2.03%
Neutral	784	2.36%	1,524	4.59%

Figure 23 and Table 12 illustrates the comparison of the performance of VADER and TextBlob in classifying Airbnb reviews into positive, negative, and neutral sentiments. These findings highlight that while both tools are relatively consistent in identifying positive sentiments, with VADER identifying 93.64% (31,058 reviews) and TextBlob identifying 93.38% (30,973 reviews), TextBlob tends to classify more comments as neutral, while VADER identifies a higher percentage of negative sentiments.

Table 13. Example of mismatch sentiment analysis between two analytics tools

Guest	Reviews	VADER	TextBlob
guest_518	I like it	positive	neutral
guest_838	everything was fantastic i would absolutely stay at his flat again josh is user communicative and responsive he even replaced the broken hair dryer on the same day that i asked him about it	positive	neutral
guest_1096	Anna is a superstar she goes beyond the expected	neutral	negative
guest_1238	i didn't have a good time here and ended up having to leave early and requesting a partial refund hoping list will sort out the issues with the flat cleanliness broken shower no hot water issues with heating before charging anyone else to stay there	negative	positive
guest_2530	comfy places to stay close to public transit as well definitely will be back again	positive	neutral
guest_2675	decent stay for the price but not the most hygienic	neutral	positive
guest_2754	very cozy for two closes to everything train station 1 in walking distance very private rarely see anyone very welcomed and understanding felt like at home will recommend anyone to stay there	positive	neutral
guest_7541	it doesn't feel very worthwhile the location is very remote, and the room is very small	negative	positive

TextBlob was initially selected for its straightforward implementation and user-friendly interface. However, it was observed that TextBlob frequently misclassified sentiments, particularly in contexts involving sarcasm, abbreviations, and colloquial language typical of social media and review texts. VADER demonstrated superior performance in handling the nuances of social media text, including slang, abbreviations, and emoticons. VADER also provided a more detailed and nuanced sentiment classification.

To improve accuracy, VADER was supplemented with a custom lexicon tailored specifically to the unique context of Airbnb reviews. This custom lexicon included domain-specific terms and phrases frequently used by guests, thereby enhancing the precision of the sentiment classification process.

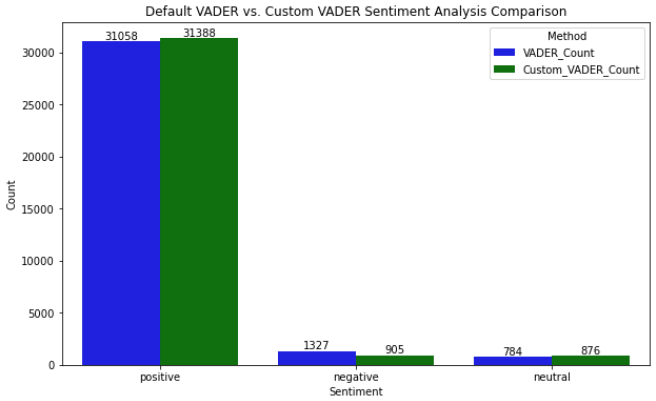


Figure 24. Comparison of Sentiment Analysis between custom and default VADER

Table 14. Comparison between custom-lexicon and default VADER

Sentiment	Default VADER		Custom VADER	
	Count	Percentage	Count	Percentage
Positive	31,058	93.63%	31,388	94.63%
Negative	1,327	4.00%	905	2.73%
Neutral	784	2.36%	876	2.64%

Table 15. Example of mismatch sentiment analysis between two analytics tools

Guest	Reviews	VADER	Custom
guest_1144	very practical apartment for families and well located we were very well received	neutral	positive
guest_1232	well placed and comfy flat	neutral	positive
guest_4011	had everything worked out without a problem thank you	negative	neutral
guest_15476	well if i could i'd buy the place and just live there	neutral	positive
guest_24676	the apartment as shown in the photos very well located i would stay there again	neutral	positive

The decision was made to proceed with using the custom VADER sentiment analysis tool due to its enhanced accuracy in capturing the nuances of Airbnb reviews. This custom lexicon incorporated domain-specific terms frequently used by guests, thereby refining the sentiment classification process. The final analysis showed that positive reviews constituted 94.63% of the total, reflecting a generally favourable guest experience. In contrast, negative reviews accounted for 2.73%, and neutral reviews represented 2.64% as Figure 25.

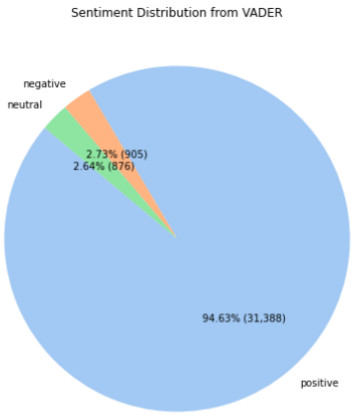


Figure 25. Sentiment Distribution

To further understand the distribution of sentiments and the key terms associated with each sentiment category, visualisations such as bar charts and word clouds were employed. These tools provided a comprehensive overview of the data and in-depth analysis of guest experiences as reflected in their reviews.

The bar chart and word clouds illustration in Figure 26,27 and 28 highlight the most frequently mentioned terms in Airbnb reviews. In the positive word cloud, words like “great,” “location,” “clean,” “apartment,” and “host” are prominent, reflecting guests’ appreciation for the overall quality of accommodations and satisfaction with cleanliness and host interaction.

In contrast, the negative reviews word cloud highlights significant areas of dissatisfaction. Words such as “small”, “bad”, “didn’t”, “however”, and “dirty” are standing out. Concerns related to room size, amenities, host responsiveness, and cleanliness are frequently mentioned in negative reviews, indicating key areas where improvements can be made to enhance guest satisfaction.

Figure 26. Positive Reviews Word Cloud

Figure 27. Negative Reviews Word Could

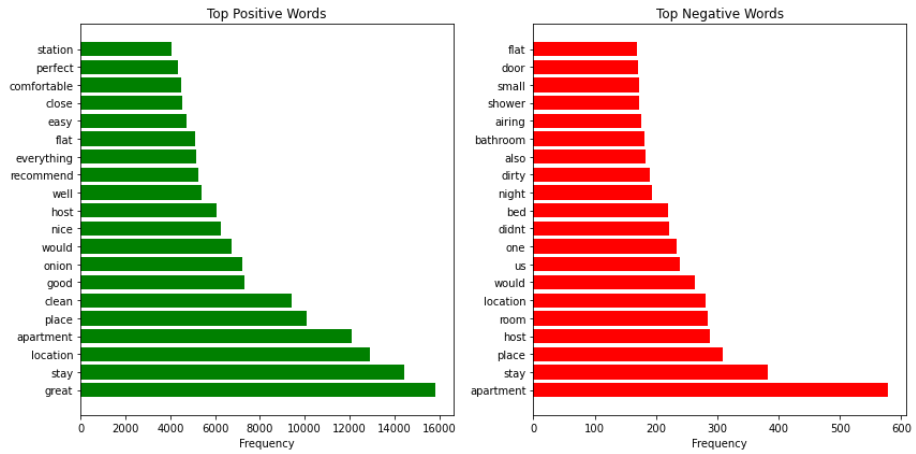


Figure 28. Top positive and negative words

These charts effectively capture the essence of guest sentiments which highlight key areas that contribute to both positive and negative experiences. The analysis of these terms allows Airbnb hosts to pinpoint specific areas of improvement and capitalise on aspects that guests appreciate, thereby enhancing the overall guest experience. Additionally, the frequent mention of words related to location underscores the importance of a property's geographical setting in shaping guest perceptions. By addressing the concerns highlighted in negative reviews and reinforcing positive experiences, hosts can improve their service offerings and potentially increase guest satisfaction and loyalty.

Sentiment trends over time

The analysis of sentiment trends over time provides valuable insights into how external events and changes in the socio-economic landscape impact guest experiences and perceptions of Airbnb accommodations. In this study, the focus was on significant events, such as Brexit and the COVID-19 pandemic which have profoundly affected the tourism and hospitality industries. Visualisations depict the sentiment trends, highlighting notable shifts in guest sentiments during these periods.

For this analysis, DistilBERT was employed due to its ability to capture the contextual relationship between words and phrases. The use of DistilBERT in this analysis provided several advantages over traditional sentiment analysis models, primarily due to its efficiency in processing large datasets and its ability to capture nuanced sentiment expressions. The confusion metrics evaluated by DistilBERT, as shown in Figure 29, indicates a high level of accuracy (98.34%) and precision (98.29%) in classifying sentiments into negative, neutral, and positive categories. This level of performance underscores DistilBERT's robustness in handling varied sentiment expressions within Airbnb reviews.

```
Accuracy: 0.9834
Precision: 0.9829
Recall: 0.9834
F1-Score: 0.9831

Classification Report:

              precision    recall  f1-score   support

    negative       0.75      0.70      0.72       171
     neutral       0.95      0.86      0.90       173
    positive       0.99      0.99      0.99      6290

    accuracy                           0.98      6634
   macro avg       0.89      0.85      0.87      6634
weighted avg       0.98      0.98      0.98      6634
```

Figure 29. Confusion Metrics for DistilBERT Model

The training process in Figure 30, which spanned three epochs, exhibited a steady decline in both training and validation loss, indicating the model's successful learning and adaptation to the complexities of the dataset.

Starting training...

[4977/4977 16:42, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	0.111700	0.078107
2	0.077300	0.061327
3	0.056600	0.068708

Starting evaluation...

[104/104 00:23]

{'eval_loss': 0.06132666394114494, 'eval_runtime': 23.3212, 'eval_samples_per_second': 284.4

Figure 30. DistilBERT Model training and validation loss

Table 16. Sentiment distribution from DistilBERT model

Sentiment	Counts	Percentage
Positive	31,440	94.79%
Negative	961	2.90%
Neutral	768	2.31%

Table 16 presents the sentiment distribution results from the DistilBERT model providing a comprehensive breakdown of sentiment classification across all Airbnb reviews analysed, which will be further examined over time as illustrated in Figure 31.

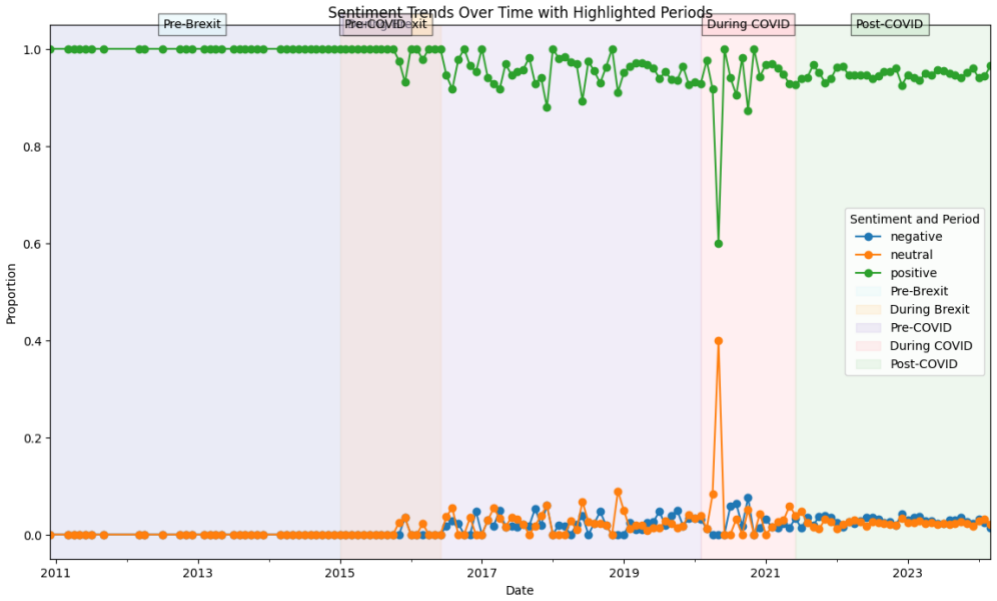


Figure 31. Sentiment trends over time with highlighted periods.

During the pre-Brexit period, sentiments were predominantly positive indicating stable and favourable perception of Airbnb accommodations. However, during Brexit, a slight decline in positive sentiments was observed suggesting uncertainty and hesitancy among guests, possibly due to political and economic instability. Pre-COVID-19, sentiments returned to a more stable positive trend as guests regained confidence in travel and accommodation services. During COVID-19, there was a significant dip in positive sentiments, accompanied by a spike in neutral and negative sentiments reflecting the widespread

disruption caused by the pandemic including travel restrictions and health concerns. Post-COVID-19, an upward trend in positive sentiments indicates recovery and adaptation within the Airbnb market as guests adjust to new norms and safety measures. Airbnb hosts can use these insights to prioritize and emphasize cleanliness and health. By doing so, hosts can not only enhance guest satisfaction but also build trust which is crucial during uncertain times.

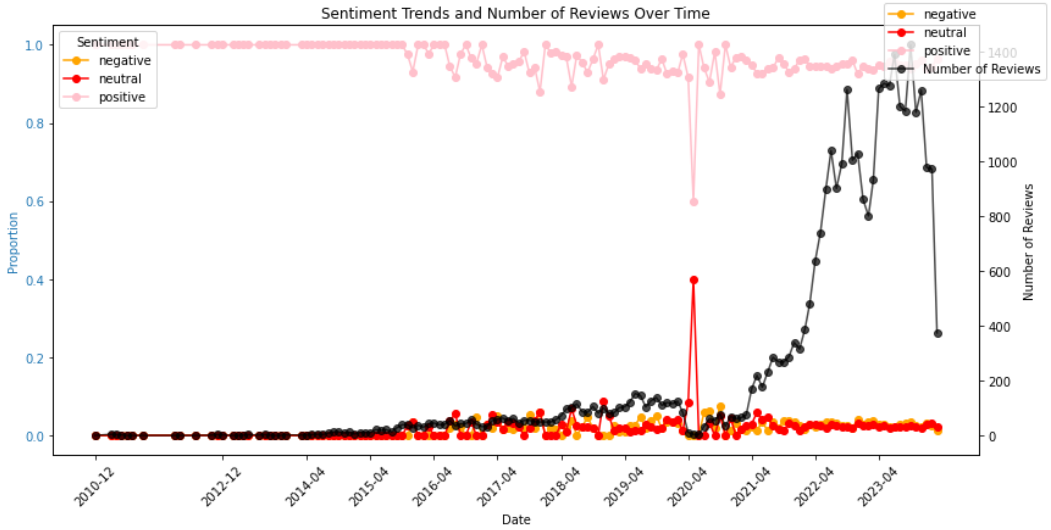


Figure 32. Sentiment trends and number of reviews over time

Figure 32 explores the relationship between sentiment proportions and reviews. The graphs collectively illustrate that while positive sentiments are predominant, external events significantly impact the sentiment landscape. The analysis indicates a noticeable decrease in sentiment during significant periods, suggesting that factors such as political and economic uncertainties, along with disruptions in travel, influenced guest perceptions.

The spike in neutral reviews observed from January 4, 2020, to June 1, 2020, is analysed further in Figure 33 in neutral reviews during the early COVID-19 period is due to the low total review count which makes a small number of neutral reviews appear more significant due to the normalisation method used in plotting.

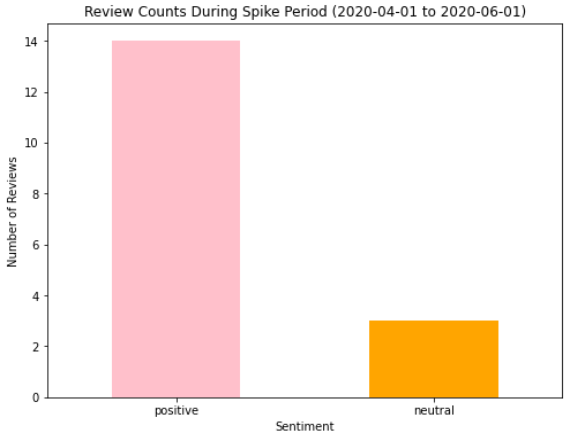


Figure 33. The proportion of review during spike period

Further analysis of guest reviews during the COVID-19 period reveals that discussions primarily focused on concerns about cleanliness and travel restrictions as shown in Figure 34. The emphasis on cleanliness reflects heightened awareness and expectations for hygiene standards which is being driven by the pandemic's impact on public health awareness. Guests had concerns about the thoroughness of cleaning processes.

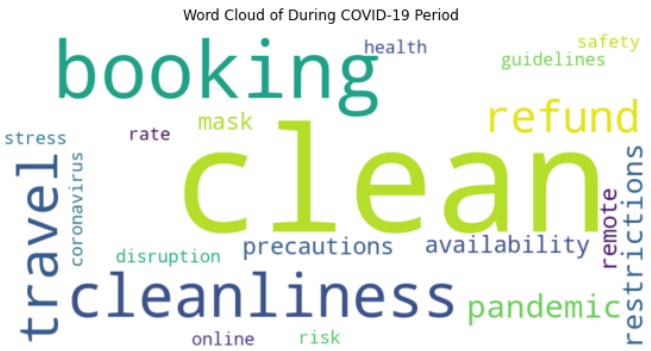


Figure 34. Word Cloud during Covid-19 period

Additionally, further analysis of the word cloud during the Brexit period as Figure 35 was conducted, but the words did not provide any significant insights related to the event.

Figure 35. Word Cloud during Brexit period

These findings underscore the importance of adapting to changing consumer priorities, particularly during periods of uncertainty. For Airbnb hosts, maintaining high cleanliness standards and transparent communication regarding safety measures are crucial for building trust and ensuring positive guest experiences. Moreover, offering flexible booking policies can help mitigate the negative impacts of travel disruptions fostering guest loyalty in the long term. Policymakers could use these insights to set best practices in the sharing economy, ensuring hosts meet cleanliness and safety standards. This might involve certification programs or resources to help hosts enhance their services improving market quality and boosting consumer confidence. Additionally, regulations can be developed to balance market growth with consumer protection, ensuring the sharing economy benefits all stakeholders.

Thematic Analysis – NER and Topic Modelling

The thematic analysis of Airbnb guest reviews through NER and LDA has revealed valuable insights into the factors influencing guest satisfaction. These themes emerged from an analysis of sentiment distribution across various factors, including host type, accommodation type, and neighbourhood.

Sentiment distribution by host type

The study as shown in Figure 33 and summarised in Table 17, indicates that Superhosts (a host who has consistently provided exceptional hospitality and met specific criteria set by Airbnb (Airbnb, n.d.) receive a higher percentage of positive reviews (96.65%) compared to normal hosts (92.05%). Although Superhosts have fewer total reviews, their lower percentages of negative (1.78%) and neutral (1.57%) reviews highlight their ability to provide superior guest experiences. This suggests that superhosts are more successful in delivering exceptional service which is positively reflected in guest feedback.

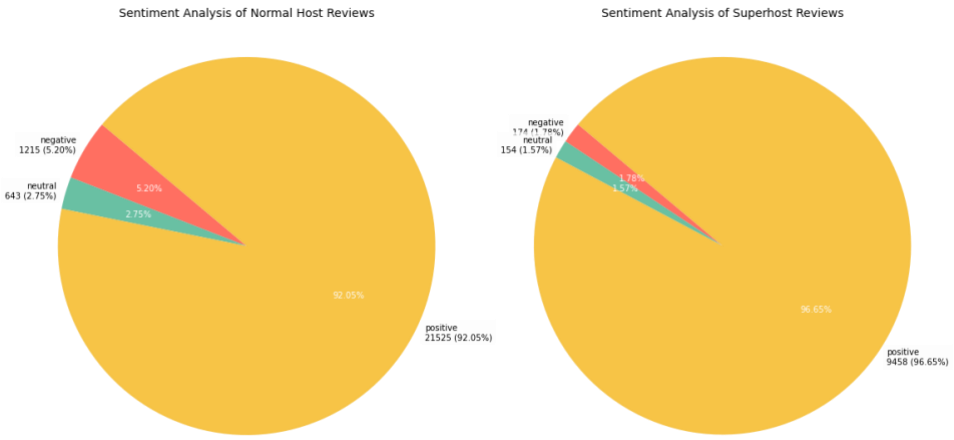


Figure 36. Sentiment Analysis by type of host

Table 17. Distribution of sentiment by host type

Host type	Sentiment		
	Positive	Negative	Neutral
Host	21,525 (92.05%)	1,215 (5.20%)	643 (2.75%)
Superhost	9,458 (96.65%)	174 (1.78%)	154 (1.57%)

Consequently, NER is employed to extract entities from guest reviews. As the result from Figure 37, Cardinal numbers were the most prevalent highlighting the importance of numerical details such as pricing, room capacity, and location distances in guest feedback. Ordinal numbers and time-related entities further emphasise the importance of specific scheduling and sequence information, such as check-in/check-out times or visit durations. Despite being less frequent, mentions of people (PERSON) hint at the influence of host interactions and personal experiences on guest satisfaction.

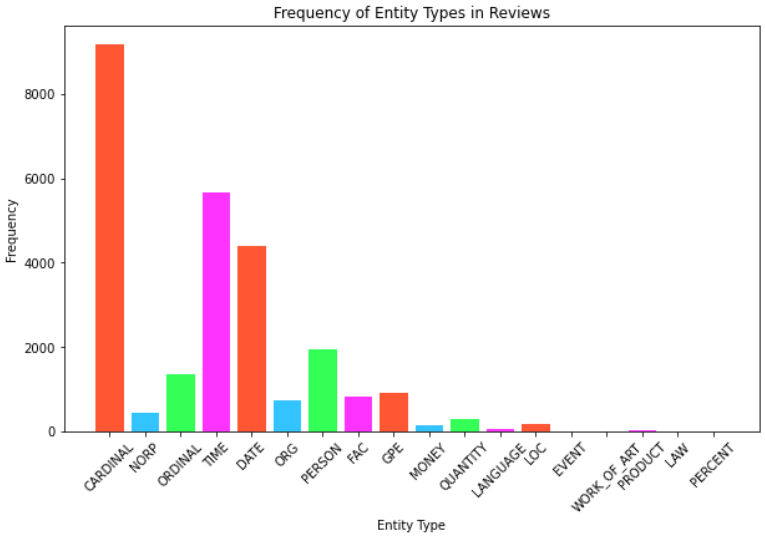


Figure 37. Frequency of Entity types in reviews

The analysis reveals a clear distinction in guest sentiment between Superhosts and normal hosts. This finding aligns with the established reputation of Superhosts for providing exceptional service quality, which is consistently reflected in guest feedback.

Sentiment distribution by accommodation type

The study reveals that entire serviced apartments and private rooms tend to receive more positive reviews suggesting that guests value the privacy and amenities offered by these accommodations. These options are more popular as indicated by the higher number of reviews. In contrast, shared rooms and budget options often show more neutral or mixed sentiments, likely due to concerns about privacy and limited amenities as depicted in Figure 38 and summarised in Table 18.

Figure 38. Distribution of sentiment by accommodation type

Table 18. Distribution of sentiment by accommodation type

Accommodation type	Sentiment	Counts	Percentage
Entire serviced apartment	Positive	26,686	93.46%
	Negative	1,198	4.20%
	Neutral	669	2.34%
Private room	Positive	3,342	93.53%
	Negative	136	3.81%
	Neutral	95	2.66%
Room in serviced apartment	Positive	951	91.71%
	Negative	54	5.21%
	Neutral	32	3.09%
Shared room	Positive	5	83.33%
	Negative	0	0.00%
	Neutral	1	16.67%

Topic modelling using Latent Dirichlet Allocation (LDA) was applied in this study to uncover hidden themes within Airbnb guest reviews. The results from Figure 39 show that the topics are divided into five main categories, which are then summarised in Table 19. This can help hosts and property managers identify key areas of focus for improving guest satisfaction and tailor their services to meet specific guest needs. By understanding these core themes, stakeholders can address common concerns, enhance positive experiences and ultimately drive better reviews and increased bookings.

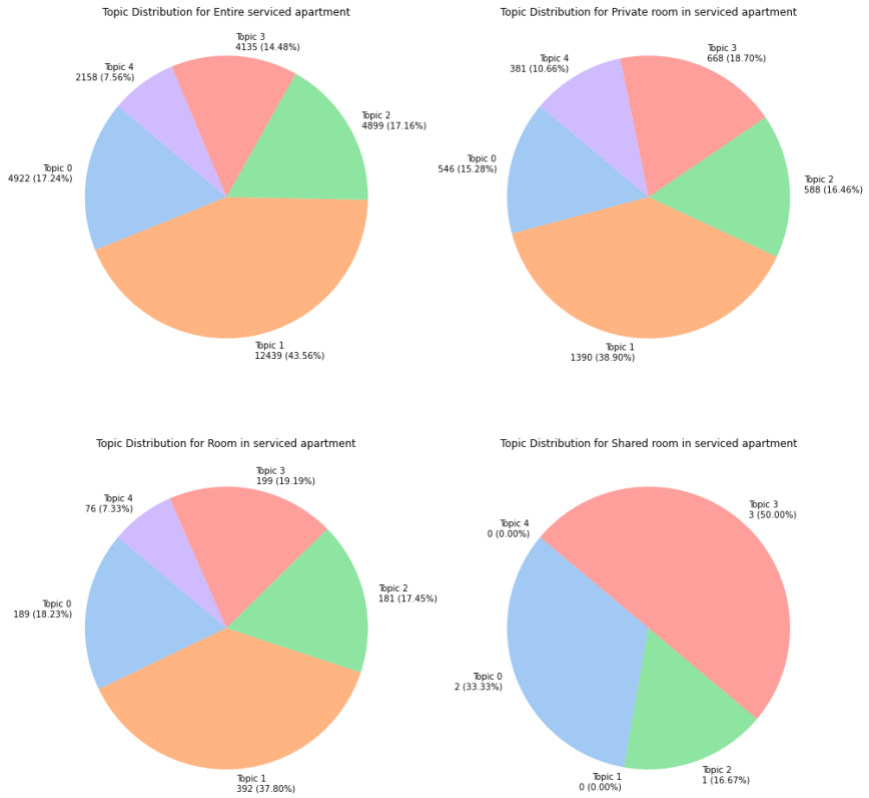


Figure 39. Topic Distribution by accommodation type

Table 19. Summarise of topic distribution by accommodation type

Property type	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4
Entire serviced apartment	4,922 (17.24%)	12,439 (43.56%)	4899 (17.16%)	4,135 (14.48%)	2158 (7.56%)
Private room in serviced apartment	546 (15.28%)	1,390 (38.90%)	588 (16.46%)	668 (18.70%)	381 (10.66%)
Room in serviced apartment	189 (18.23%)	392 (37.80%)	181 (17.45%)	199 (19.19%)	76 (7.33%)
Shared room	2 (33.33%)	0 (0.00%)	1 (16.67%)	3 (50.00%)	0 (0.00%)

Table 20. LDA topic modelling by accommodation type

Topic	Keywords	Key findings
0	room, apartment, bed, would, bit, bathroom, one, small, night, kitchen	This topic likely focuses on specific features and layout of the apartment or room, highlighting aspects like the size of the bed, bathroom, and kitchen. It also includes comments about the space being compact or cosy. This can provide insights into how guests perceive the apartment's physical characteristics.
1	great, stay, place, location, would, clean, recommend, host, apartment, definitely	This topic emphasises positive guest experiences, focusing on location, cleanliness, and the host's hospitality. It also suggests high satisfaction, indicating that these are key factors contributing to positive reviews. This can guide property owners on what to maintain or improve to ensure positive feedback.
2	station, close, walk, London, tube, restaurants, minutes, easy, great, apartment	This topic centres around the property's proximity to public transportation and amenities. It also highlights the convenience of the location, which is a crucial factor for guests seeking accessible accommodations. This topic can help identify properties that are popular for their strategic location.
3	good, well, apartment, clean, accommodation, nice, location, stay, located, comfortable	Similar to Topic 1, this topic focuses on the general quality of the accommodation. However, it appears more balanced, with an emphasis on "comfortable" and "nice," suggesting overall satisfaction with the living conditions. This reinforces the importance of cleanliness and comfort in guest experiences.
4	us, host, airing, time, check, stay, even, checking, made, one	This topic seems to focus on the interaction between guests and the host, specifically around check-in and check-out experiences. It also suggests an emphasis on logistics and personal interactions, potentially indicating that guests value smooth and efficient check-in/check-out processes and the host's efforts to accommodate them.

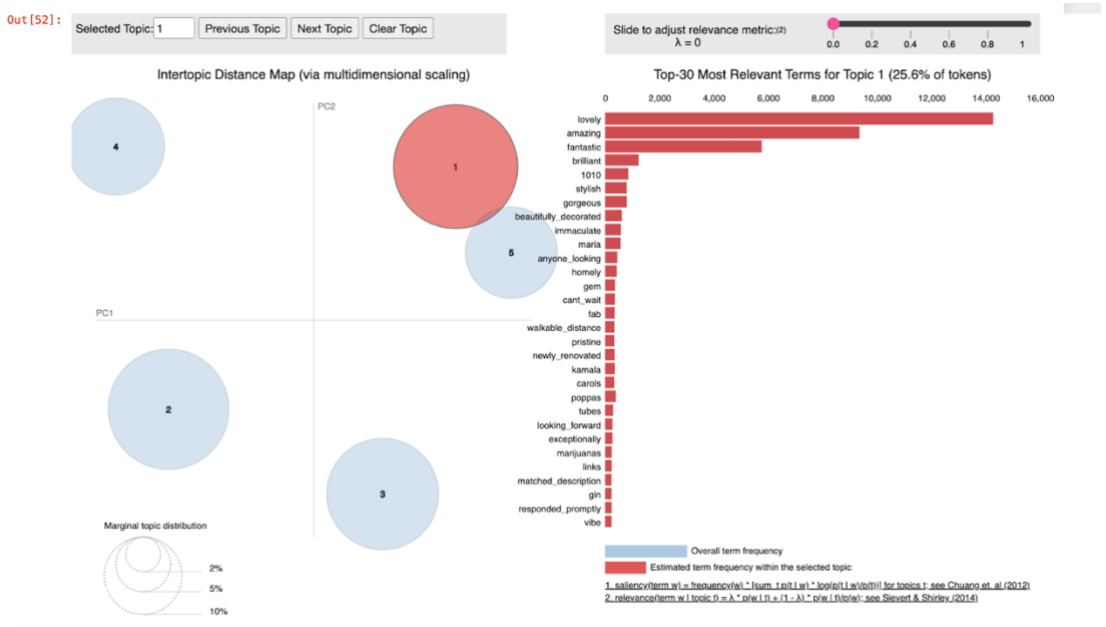


Figure 40. pyLDAvis visualisation of topic modelling

Sentiment distribution by neighbourhood

The study highlights geographical factors that influence guest evaluations. Figure 41 provides a comprehensive overview of sentiment distribution across various neighbourhoods, as analysed through the LDA topic modelling. This geographical breakdown reveals significant patterns and insights into how different areas within the city are perceived by guests. Neighbourhoods like Kensington and Westminster tend to receive more positive reviews due to proximity to tourist attractions and vibrant nightlife. Understanding these geographical insights allows hosts to tailor their offerings to meet specific guest preferences and create personalised experiences that enhance satisfaction.

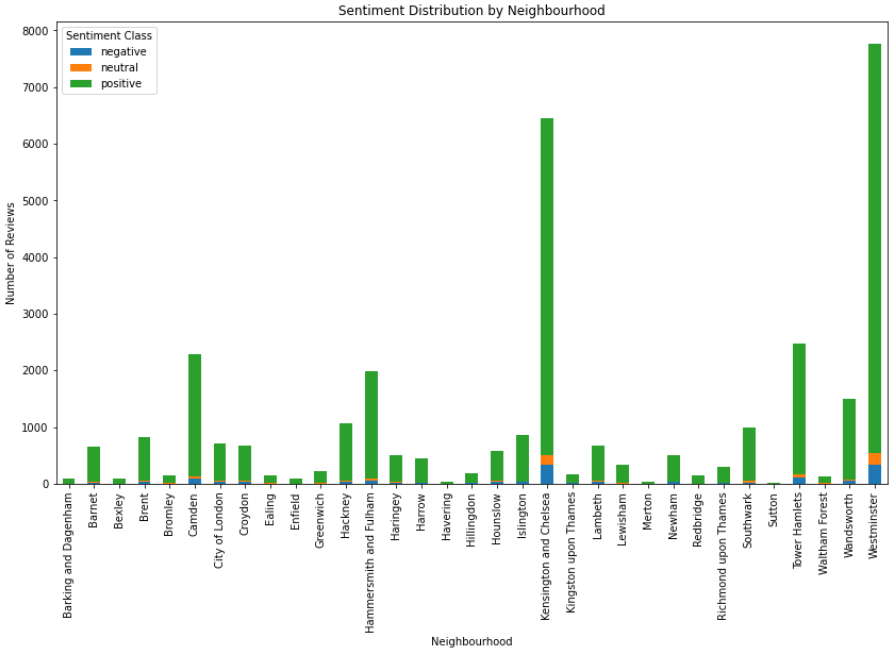


Figure 41. Sentiment Distribution by neighbourhood

Topic modelling, illustrated in Figure 42 and summarised in Table 21, breaks down guest reviews into five distinct topics. Each topic highlights key aspects of guest experiences that influence sentiment distribution. These insights can guide trip planning enabling hosts to create personalised itineraries and themed experiences that align with guest interests, such as dining, walkability, and luxury shopping. By understanding the themes that resonate with guests, stakeholders can improve the quality of their listings, optimise marketing strategies, and effectively position their properties within the competitive Airbnb market.

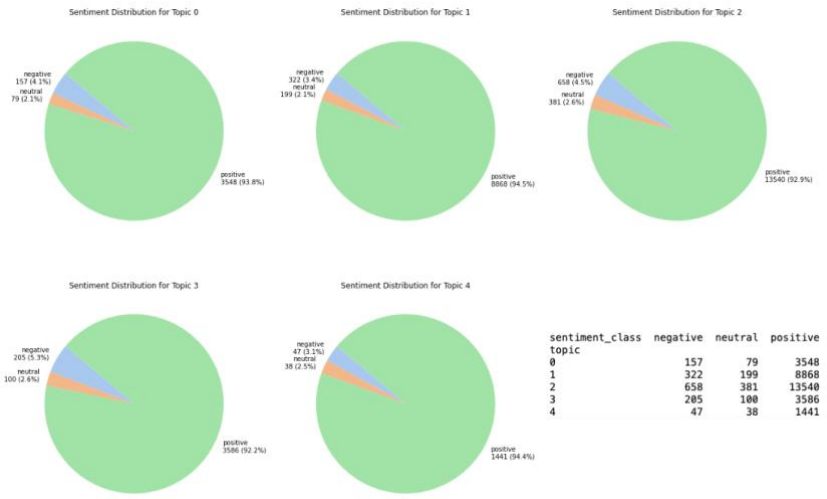


Figure 42. Sentiment distribution of topic

Table 21. LDA topic modelling by neighbourhood

Topic	Keywords	Key Findings
0	London, street, located, restaurants, area, home, famous, central, west, just	This topic centres around properties located in well-known areas of London, emphasising their central location and proximity to restaurants. It suggests a sense of prestige or fame associated with these neighbourhoods.
1	Walk, London, minutes, away, street, station, park, minute, restaurants, road	This topic emphasises the walkability and convenience of neighbourhoods, highlighting easy access to parks, restaurants, and public transportation.
2	Provided, London, Kensington, area, famous, royal, south, road, city, shopping	This topic highlights upscale neighbourhoods, especially Kensington, known for its royal associations and shopping options.
3	London, area, street, city, located, shopping, restaurants, residential, world, wharf	This topic revolves around urban residential areas that offer shopping, dining, and potentially waterfront views (e.g., Canary Wharf).
4	London, Kensington, lots, park, west, area, famous, restaurants, spaces, transport	This topic discusses areas with abundant green spaces and transport options, particularly in West London, such as Kensington.

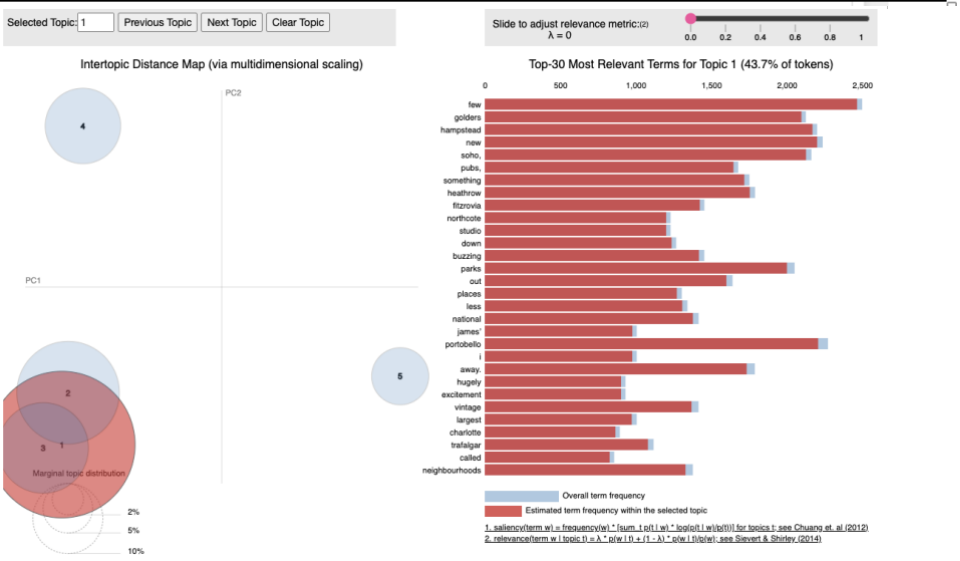


Figure 43. pyLDAvis visualisation of topic modelling

By mapping topics and identifying common themes in guest reviews, hosts can better understand what aspects of the experience guests value most. For example, frequent mentions of local attractions or dining experiences can lead hosts to develop personalised tours or experiences tailored to these interests. Segmenting guests by preferences enables hosts to offer customised activities that enhance their stay, foster community and create lasting memories. This strategic approach not only boosts guest satisfaction but also allows hosts to stand out in a competitive market by offering unique services that go beyond traditional accommodation offerings.

For policymakers, these insights could inform the development of targeted regulations to ensure that hosts maintain high standards of cleanliness, safety and service quality. By promoting best practices and establishing clear guidelines, policymakers can help improve overall market quality and enhance consumer confidence in the sharing economy.

Correlation Analysis

The analysis begins with a chi-square test to determine the statistical significance of these relationships. Results from the chi-square test illustrated in Figure 44, show significant relationship between certain accommodation features and guest sentiments. Amenities, location convenience, and room type have a substantial impact on the sentiments expressed in guest reviews, suggesting that improvements in these areas could enhance guest satisfaction. Additionally, review scores across categories such as cleanliness, communication, and value are strongly correlated with positive guest sentiments indicating that higher scores in these categories lead to more favourable reviews.

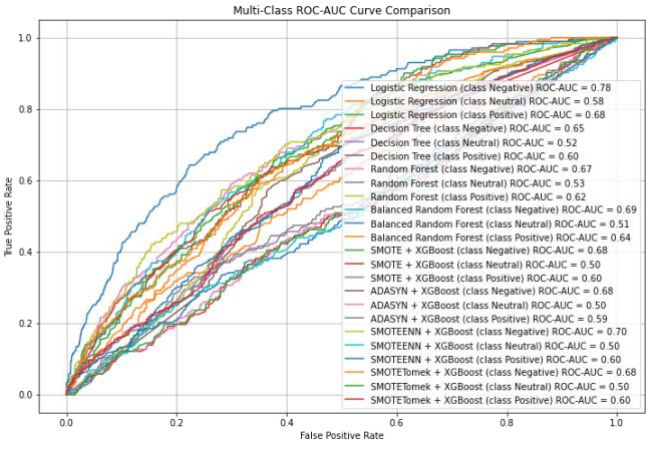
Feature	Chi-Square Statistic	p-value	Degrees of Freedom
Amenities	4551.598136	2.178685e-86	2810
Location Convenience	106.760211	6.403189e-04	64
Room Type	23.230330	7.228933e-04	6
Pricing	9.803795	1.331615e-01	6
review_scores_rating	822.932930	8.275104e-177	4
review_scores_accuracy	717.352327	6.093027e-154	4
review_scores_cleanliness	620.713086	5.094762e-133	4
review_scores_checkin	509.974115	4.663864e-109	4
review_scores_communication	616.089722	5.103149e-132	4
review_scores_location	324.088714	6.875996e-69	4
review_scores_value	796.883169	3.634727e-171	4

Figure 44. Chi-square testing

Given the dataset's imbalance with predominantly positive sentiments, the analysis proceeds with logistic regression as a baseline model, followed by the application of machine learning techniques as listed in Table 22. Various models are evaluated using the ROC-AUC curve, as shown in Figure 45, to determine the most effective method for predicting guest sentiments.

Table 22. Comparison of ROC-AUC curve on Machine Learning

Machine Learning Model	Sentiment	ROC-AUC score
Logistic Regression	Positive	0.68
	Negative	0.78
	Neutral	0.58
Decision Tree	Positive	0.6
	Negative	0.65
	Neutral	0.52
Random Forest	Positive	0.62
	Negative	0.67
	Neutral	0.53
Balanced Random Forest	Positive	0.64
	Negative	0.69
	Neutral	0.51
SMOTE	Positive	0.6
	Negative	0.68
	Neutral	0.5
ADASYN	Positive	0.59
	Negative	0.68
	Neutral	0.5
SMOTEENN	Positive	0.6
	Negative	0.7
	Neutral	0.5



The results indicate that the SMOTEENN model outperformed other techniques by effectively addressing class imbalance (Mustafa, 2019). This model demonstrated relatively superior performance in predicting the negative class which is the minority in the dataset. Furthermore, the confusion matrix presented in Figure 46, shows that the accuracy for this model is approximately 82.39%, which is notably high and satisfactory.

```
Training Classification Report (SMOTEENN):
              precision    recall  f1-score   support

    Negative       1.00      1.00      1.00     13010
     Neutral       1.00      1.00      1.00     12704
    Positive       1.00      1.00      1.00     21481

    accuracy                           1.00     47195
   macro avg       1.00      1.00      1.00     47195
weighted avg       1.00      1.00      1.00     47195

Testing Classification Report (SMOTEENN):
              precision    recall  f1-score   support

    Negative       0.07      0.19      0.10       171
     Neutral       0.03      0.10      0.05       173
    Positive       0.95      0.86      0.91      6290

    accuracy                           0.82      6634
   macro avg       0.35      0.38      0.35      6634
weighted avg       0.91      0.82      0.86      6634

Training Accuracy (SMOTEENN): 0.9999364339442738
Testing Accuracy (SMOTEENN): 0.8239372927343985
```

Figure 45. Classification report for XGBoost SMOTEENN

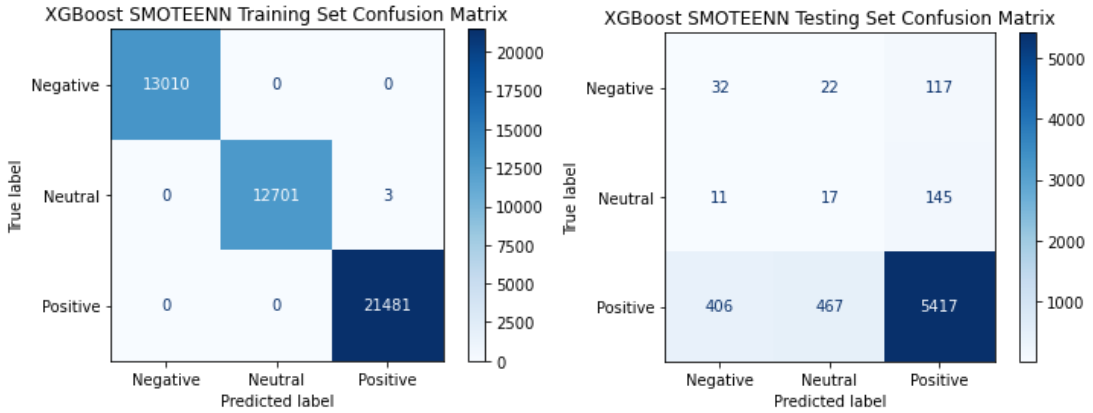


Figure 46. Confusion Matrix for XGBoost SMOTEENN

Consequently, SMOTEENN model was selected to explore the correlation between specific features of apartment-type accommodations and guest sentiments. As demonstrated in Figure 47, the analysis reveals that review scores are the most significant predictors of guest sentiment. This finding emphasises the critical role that guest ratings and perceived value play in shaping overall guest sentiment.

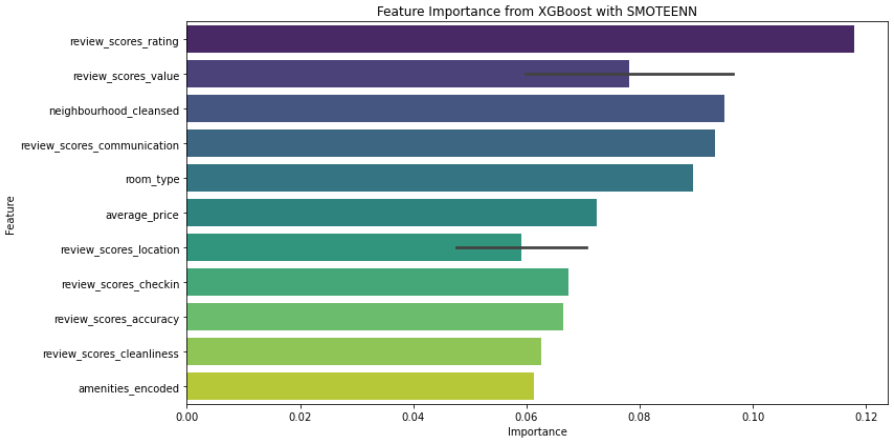


Figure 47. Feature importance from SMOTEENN

The insights from this analysis offer practical applications for both Airbnb hosts and policymakers. For hosts, recognizing which features most influence guest sentiments allows for targeted enhancements. Improving amenities, ensuring accurate location details, and better communication can directly boost guest satisfaction and lead to more positive reviews. Maintaining high review scores in areas such as cleanliness, value, and communication is also crucial encouraging positive sentiments. Policymakers can use these insights to develop regulations that set minimum standards for cleanliness, listing accuracy, and amenities to ensure consistent quality of service across the Airbnb market.

Market Dynamics on Sentiments

An analysis of market dynamics in Airbnb, including listing growth, pricing changes, and occupancy rates, reveals how these factors influence guest sentiments.

Market dynamics and Sentiment analysis

Figure 48 illustrate how the dynamics of the Airbnb market in London influence guest sentiment over time. Three key metrics; sentiment scores, the number of listings, and average price are analysed. Sentiment scores fluctuate in response to significant market events and global disruptions, such as the COVID-19 pandemic indicating that guest satisfaction is influenced by broader socio-economic conditions. The number of listings over time reflects market growth and increased competition which can affect service quality and guest reviews. Despite the increase in listings reflecting market growth, the slight decline in positive sentiment suggests that market saturation may lead to competitive pressures. Interestingly, while average price trends reveal periods of dramatic price spikes, these spikes do not necessarily correlate with negative sentiments. This finding suggests that factors such as service quality and location may be more critical in guest evaluations than pricing.

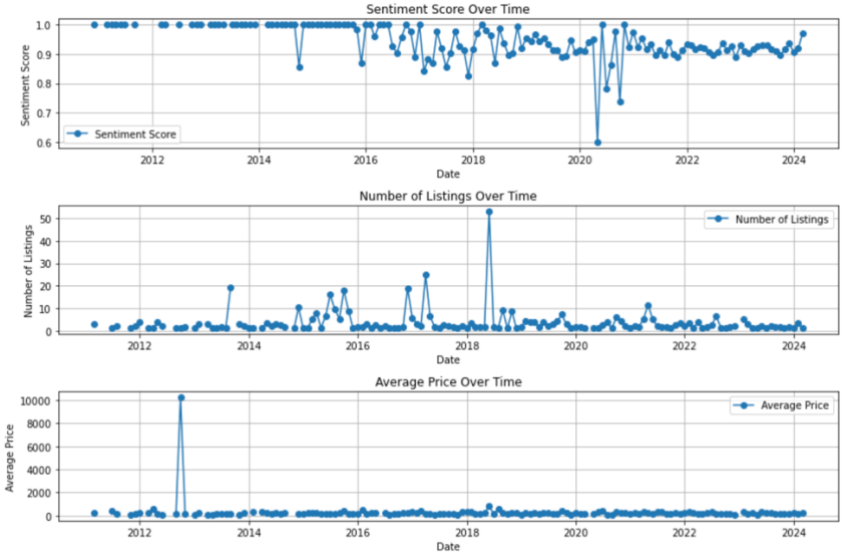


Figure 48. Sentiment scores, number of listing and average price over time

Occupancy Rates and sentiment correlation

The graph in Figure 49 illustrates notable fluctuations in occupancy rates across different time periods (30, 60, 90, and 365 days) highlighting the dynamic nature of the Airbnb market. Short-term occupancy rates (30 and 60 days) demonstrate greater volatility which indicates a sensitivity to immediate market conditions and booking patterns. This variability suggests that short-term rentals are more responsive to changes in demand and availability, likely influenced by seasonal trends and external events.

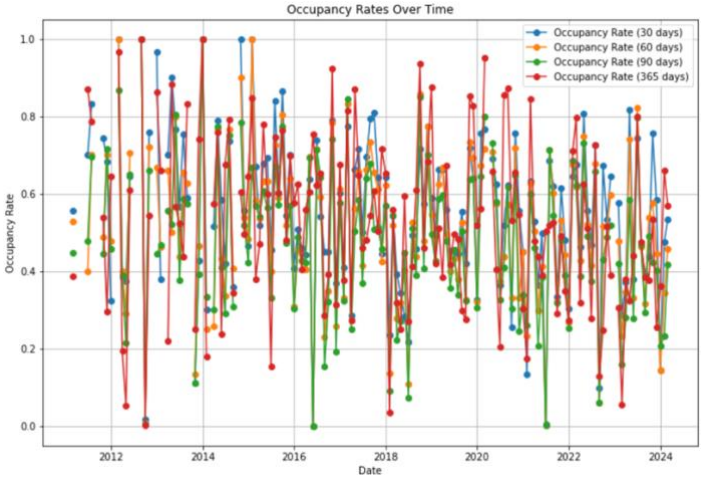


Figure 49. Occupancy rates over time

The correlation heatmap as Figure 50 reveals that sentiment scores are weakly related to the number of listings, and prices indicate that guest satisfaction is not significantly impacted by these factors. The slight negative correlation between sentiment scores and the number of listings suggests that guest satisfaction might decrease marginally as the market expands.

The analysis of market dynamics suggests a more notable relationship between prices and occupancy rates. Specifically, higher prices are moderately associated with lower occupancy rates implying that properties with higher pricing may experience less frequent bookings.

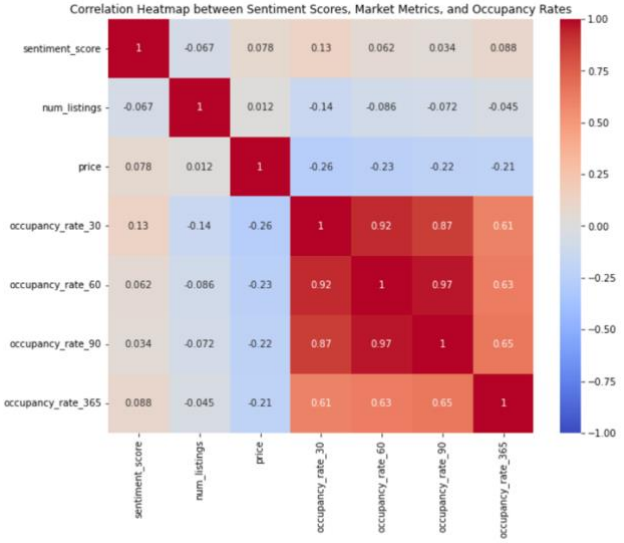


Figure 50. Correlation Heatmap between sentiment scores and market metrics

Impact of External Events on Sentiment

The seasonal decomposition of sentiment scores as Figure 51 reveals key patterns affecting Airbnb guest perceptions over time. The trend component shows a gradual decline in sentiment, with significant drop during major events like Brexit and the COVID-19 pandemic suggesting these events negatively impacted guest satisfaction. Seasonal components highlight minor yet consistent annual fluctuations indicating recurring sentiment patterns. The residual component shows random fluctuations around zero which imply the model captures the main patterns effectively. These findings suggest that sentiment patterns driven by predictable seasonal trends.

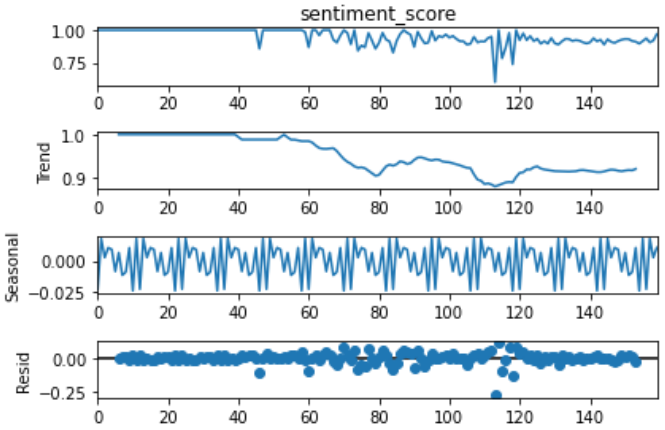


Figure 51. Seasonal decomposition analysis

By understanding these dynamics, hosts can better anticipate shifts in guest satisfaction and adjust their strategies accordingly, such as optimizing pricing during low-demand periods or enhancing service quality during peak times. Policymakers can use these insights to develop targeted regulations that ensure a balanced market and protect consumer interests while fostering healthy competition among hosts.

Predictive and forecast modelling

Pricing forecasting

The graph as Figure 52 illustrates fluctuations in Airbnb pricing over time. Initially, there was a rapid increase followed between 2014 and 2017, likely due to rising demand and competition. During the COVID-19 pandemic, prices significantly dropped as demand decreased. Post-pandemic, prices have stabilised as travel demand rebounds. This analysis highlights how external factors and market dynamics influence pricing trends in the Airbnb market. Overall, Airbnb pricing is sensitive to market dynamics and external events highlighting the importance of strategic pricing to optimise revenue.

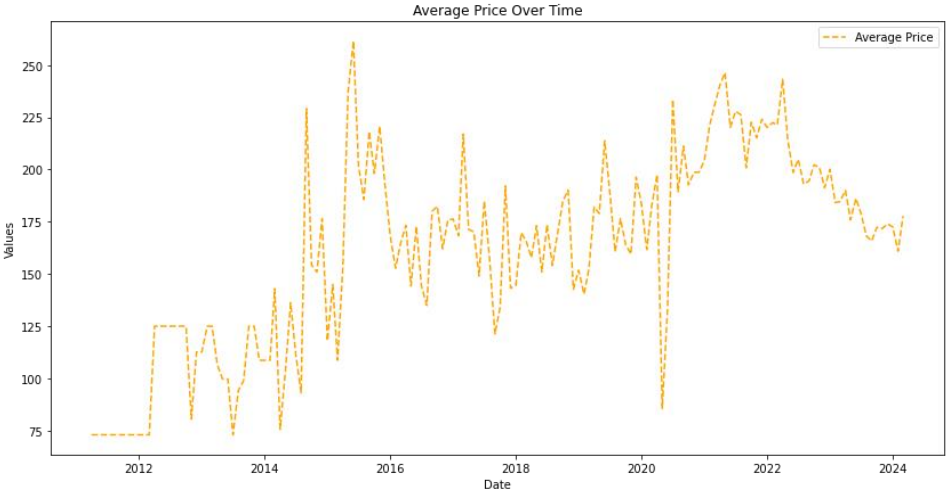
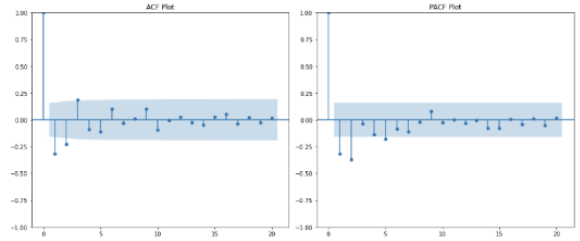


Figure 52. Average price over time

The ARIMA model was selected for forecasting due to its ability to capture the time-dependent structures and seasonal patterns present in the Airbnb pricing data. However, ARIMA alone may not adequately capture all complex patterns in the data. To enhance forecasting accuracy, HistGradientBoostingRegressor was integrated with ARIMA. This combination takes advantage of ARIMA's time series capabilities and HistGradientBoostingRegressor capacity to simulate non-linear relationships result in more reliable prediction as can be seen as Figure 53.



ACF and PACF plot interpretation (Zvornicanin, 2021)

ACF (Autocorrelation Function)

- A significant spike at lag 1 followed by a rapid decline to near-zero correlations (as seen in the ACF plot) suggests that the series has a short-term dependency.
- In the context of the ARIMA model, this pattern is indicative of the presence of a moving average component. If the ACF cuts off sharply after lag 1, it suggests that a first-order moving average model (MA(1)) is appropriate. Therefore, **q=1** is selected for the ARIMA model.

PACF (Partial Autocorrelation Function)

- A significant spike at lag 1 in the PACF plot, followed by an abrupt drop to values around zero, indicates that there is a direct relationship with the first lag but not beyond.
- This pattern suggests that a first-order autoregressive model (AR(1)) is sufficient to explain the linear dependence on its lagged values. Therefore, **p=1** is chosen for the ARIMA model.

Selection of Differencing Order (d)

- The differencing order is determined by the number of times the series needs to be different to achieve stationarity. If the series shows a trend, differencing once (d = 1) is often sufficient to stabilise the mean.
- This is typically assessed using stationarity tests like ADF (Augmented Dickey-Fuller), which is performed prior to plotting ACF/PACF (Prabhakaran, 2019). The series was different once to achieve stationarity, which aligns with the need to set d=1.

```
# Step 1: Check for Stationarity
def check_stationarity(timeseries):
    result = adfuller(timeseries)
    print('ADF Statistic:', result[0])
    print('p-value:', result[1])
    if result[1] < 0.05:
        print("Series is Stationary")
    else:
        print("Series is Non-Stationary")

# Initial stationarity check
check_stationarity(ts_data['average_price'])
```

```
ADF Statistic: -2.54060888384005
p-value: 0.10590371955049105
Series is Non-Stationary
ADF Statistic: -8.305419151880688
p-value: 3.9156776318520356e-13
Series is Stationary
```

Figure 53. Interpretation of ACF and PACF plot

The evaluation metrics for this combined technique, as presented in Table 23 show satisfactory performance indicating a well-fitted model.

Table 23. Evaluation metrics of combining ARIMA with HistGradientBoostingRegressor

Evaluation Metrics	ARIMA	ARIMA + HistGradient
MSE (Mean Square Error)	82.02	531.94
RMSE (Root Mean Square Error)	9.06	23.06
MAE	6.97	18.38
R-square	-0.36	0.71

The ARIMA forecast model as Figure 54 illustrates both historical average prices (blue line) and predicted prices for the next 12 months (red line). The model suggests a stable or slightly declining trend in the near future, based on historical data fluctuations of roughly $173 per night.

In comparison, the combined ARIMA model (green line) provides a more nuanced prediction, capturing potential nonlinear patterns in the data. It predicts a slight downward trend towards an average of $168 per night. The combined model demonstrates improved accuracy in capturing both short-term variations and overall trends.

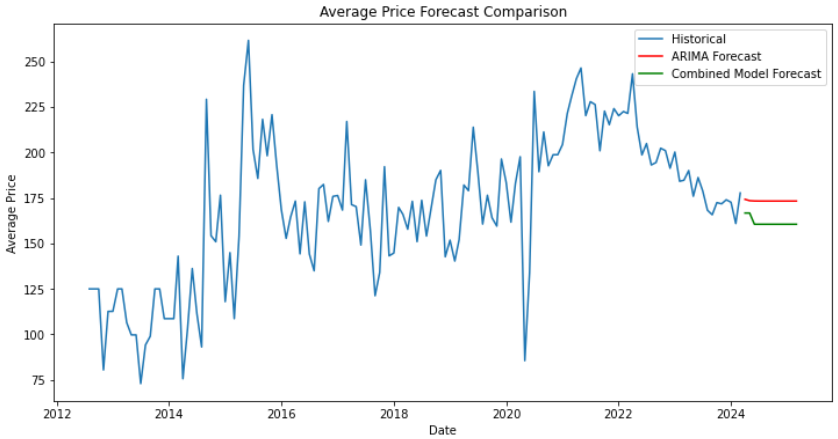


Figure 54. Average price forecast using ARIMA and HistGradientBoostingRegressor

By using advanced predictive modelling, hosts can gain insights into future market conditions and adjust pricing strategies proactively. Anticipating price fluctuations helps hosts optimize pricing to stay competitive and maximize occupancy rates, especially during seasonal changes and economic shifts. These insights also guide marketing strategies and operational adjustments to match expected market trends. Anticipating pricing changes based on historical data and market dynamics enables hosts to navigate a complex marketplace effectively. Adopting data-driven pricing strategies allows Airbnb hosts to seize market opportunities and mitigate risks related to price volatility.

Demand forecasting

The analysis was conducted by using ARIMA and SARIMAX models to assess the impact of sentiment analysis on booking patterns, offering valuable insights into future trends. The ARIMA model provides a baseline understanding of booking trends, whereas the SARIMAX model incorporates exogenous variables such as sentiment scores, thereby enhancing predictive accuracy.

To compare the performance of ARIMA and SARIMAX models for demand forecasting, a detailed analysis was conducted based on key metrics, as outlined in Table 24. The evaluation metrics reveal that SARIMAX offers superior predictive performance with lower AIC and BIC values compared to ARIMA, indicating a better fit for the data by integrating sentiment scores as exogenous variables, which account for market dynamics and guest feedback.

Table 24. Comparison of model performance

Evaluation Metrics	ARIMA	SARIMAX
ADF statistics	-5.0486	-5.0486
AIC	1850.148	1536.969
BIC	1859.355	1559.970
Coefficients (p-values)	ar.L1 : 0.000 ma.L1 : 0.000 Sigma2 : 0.000	ar.L1 : 0.632 ar.L2 : 0.000 ma.L1 : 0.784 ma.L2 : 0.886 ar.S.L12 : 0.699 ma.S.L12 : 0.149 sigma2 : 0.885

The line graph as Figure 54 and Figure 55 comparison reveals distinct differences in forecasting accuracy between the ARIMA and SARIMAX models for Airbnb demand predictions. The ARIMA model displays a stable trend that mirrors historical data but falls short of accounting for external influences. In contrast, the SARIMAX model, which integrates sentiment scores as exogenous variables, offers a more accurate reflection of actual booking patterns. By capturing recent fluctuations and aligning more closely with observed trends, SARIMAX demonstrates superior predictive accuracy and responsiveness to market dynamics. This emphasises the significance of integrating sentiment analysis into forecasting models, which provides a more nuanced understanding of demand shifts and improves the accuracy of future projections in the Airbnb market.

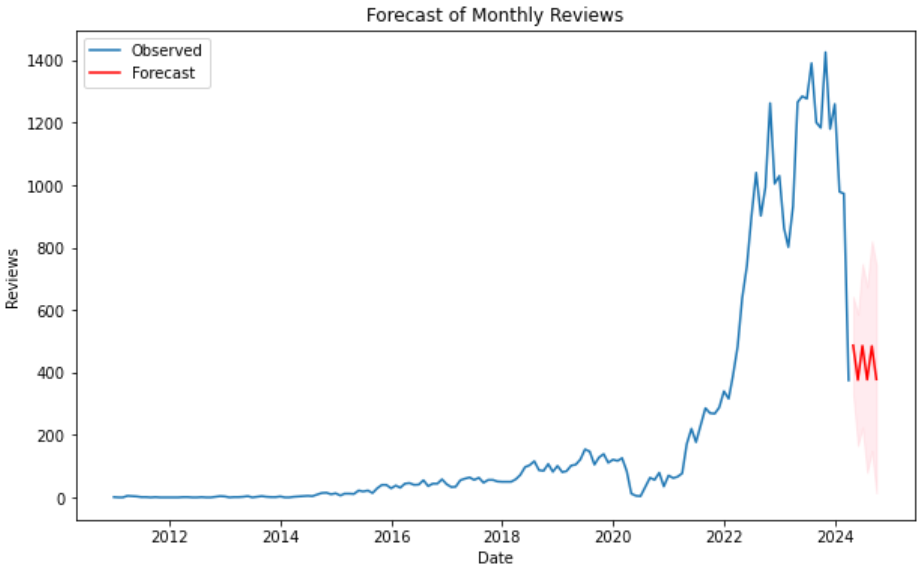


Figure 54. ARIMA model for demand forecasting

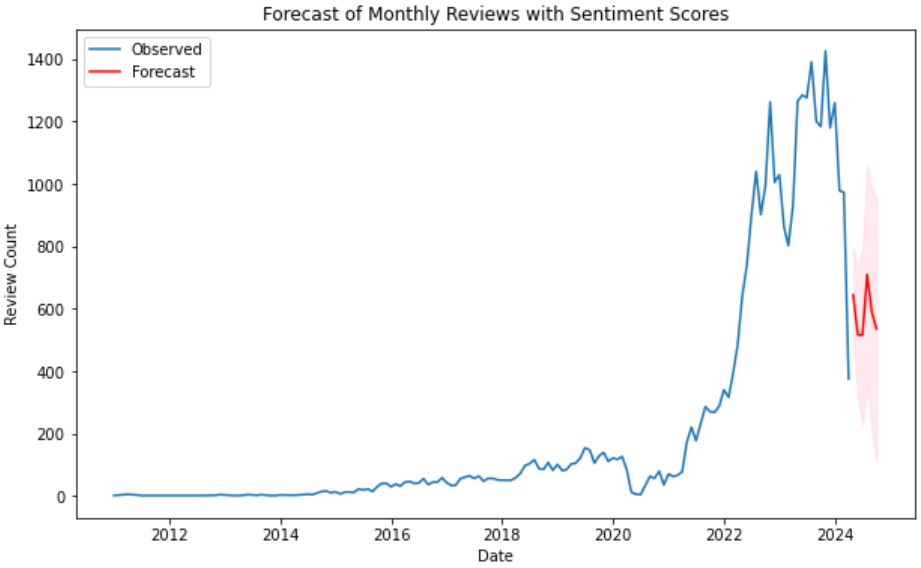


Figure 55. SARIMAX model for demand forecasting

The study's implications for Airbnb hosts and market strategists are significant. By prioritising guest satisfaction and using sentiment analysis, hosts can anticipate demand shifts and optimise pricing strategies. The study highlights the importance of aligning business strategies with market dynamics which also help hosts to maintain a competitive edge and adapt to changing consumer preferences effectively.

Cross Validation

To ensure the robustness and reliability of our findings, it is essential to cross-validate our results with prior research in the field.

Our sentiment analysis findings are consistent with previous studies that explore guest satisfaction within the Airbnb context. For example, Liu et al. (2019) demonstrated that factors such as location, cleanliness, and amenities are primary factors of guest satisfaction in short-term rentals, a finding closely mirrored in our own results. This study also identified a preference among guests for entire serviced apartments and private rooms, likely due to the privacy and comprehensive amenities these options offer. This preference aligns with Tussyadiah and Pesonen's (2016) emphasis on aligning service offerings with consumer expectations to enhance satisfaction. Furthermore, our finding that pricing does not significantly correlate with sentiment aligns with Guttentag (2015) who suggested that guests prioritize experience quality over cost rather than pricing.

Our study also highlighted significant shifts in guest sentiments during major external events, such as Brexit and the COVID-19 pandemic. These shifts align with Yang et al. (2020), who observed a trend toward more cautious and less optimistic sentiments in the tourism sector due to external shocks like pandemics. Similarly, Ert and Fleischer (2019) found that economic and political instability can lead to uncertainty, negatively impacting consumer confidence and sentiment, consistent with our findings on Brexit's impact on guest reviews. These shifts in sentiment are further supported by broader research on crisis impacts on tourism demand, such as Sönmez et al. (1999) which shows that political instability and health crises adversely affect tourist behavior and hotel occupancy rates.

Additionally, our analysis highlighted the significance of service quality, factors like cleanliness, communication, and amenities, in shaping guest satisfaction more than price, supporting findings by Tussyadiah and Pesonen (2016). This suggests that while cost influences booking decisions, it does not substantially affect the sentiment expressed in reviews, aligning with Guttentag (2015). This insight contrasts with traditional economic theories, aligning with Porter’s (1985) view on market saturation and competitive differentiation where unique value propositions are critical for maintaining high satisfaction levels in saturated markets.

Regarding market dynamics and predictive analytics, our application of time-series models like ARIMA and SARIMAX to predict future pricing and demand trends aligns with the methodologies employed by Song and Li (2008), who emphasized the importance of econometric models in tourism demand forecasting. Our use of sentiment analysis as an exogenous variable in these models reflects the approach suggested by Kim et al. (2017), advocating for integrating qualitative sentiment indicators into quantitative models to enhance prediction accuracy. Moreover, our employment of advanced machine learning models like XGBoost and the implementation of SMOTEENN for data balancing align with the innovative strategies described by Bertsimas and Dunn (2017) in their work on predictive analytics in dynamic markets. The high predictive accuracy achieved in our study confirms the effectiveness of these combined methods, as also noted by Shao et al. (2019) in their comparative study of predictive models for forecasting market behavior.

Limitation of the Study

This study presents several limitations that should be acknowledged to provide context for the findings. Firstly, the dataset comprises a substantial number of both English and non-English reviews. Although non-English reviews were translated using Python software, the potential inaccuracies inherent in machine translation could affect the sentiment analysis results, leading to a language bias.

Secondly, the data utilised in this research was sourced from Inside Airbnb which scrapes information from Airbnb listings, may also have the issues related with the completeness. This can impact the reliability of the study.

Thirdly, the sentiment analysis was conducted using the VADER tool, including custom implementation which may lead to biassed interpretations of reviews. These tools might not fully capture the nuances of real-world review tones.

Lastly, the application of machine learning techniques was challenged by the imbalanced nature of the dataset which contains a higher proportion of positive reviews. Although imbalance-handling methods such as SMOTEENN were employed, the predictive accuracy for negative and neutral sentiments may still be compromised. While these techniques partially address the issue, they do not fully eliminate the impact of class imbalance which remains a critical concern in achieving comprehensive sentiment analysis.

Ethical Implications

The integrity of the research findings is ensured through a rigorous approach to data collection and analysis, with a strong emphasis on ethical data handling. This research adhered to high ethical standards by sourcing data from reputable platforms like Inside Airbnb, which comply with high confidentiality and privacy. Although the data is publicly available, the study prioritises the responsible use of digital data in strict adherence to GDPR and other relevant regulations ensuring that all personally identifiable information (PII) is anonymized. The anonymized data will be securely stored with encryption and access controls, restricting access to authorised personnel only within the University of Exeter OneDrive. The data will be retained for a maximum period of 4 months to allow for any necessary verification or follow-up studies, after which it will be securely destroyed in November 2024 after degree award. In addition, this research aligns with ethical principles outlined in the EU AI Act emphasising transparency, accountability, and fairness in the use of AI tools like VADER for sentiment analysis (European Parliament, 2023). AI tools like VADER are employed for sentiment analysis, with all outputs reviewed and validated to avoid biases. The integration of AI in research is done with careful consideration of its ethical implications, ensuring the accuracy and fairness of the analysis.

To mitigate bias, multiple sentiment analysis tools are used and results are cross validated. Transparent reporting of any identified biases ensures the research process remains fair and balanced. The research also considers potential overrepresentation or underrepresentation by assessing the diversity of the review content. The comprehensive ethical framework is aligned with both the EU AI Act's emphasis on transparency, accountability, and fairness (European Parliament, 2023), and the UK's pro-innovation regulatory approach (Gov.UK, 2024) which highlights the study's commitment to responsible innovation and its impact on stakeholders, including Airbnb hosts and policymakers.

Conclusion

Summary of Key Findings

This research set out to investigate the various factors within the Airbnb market in London that influence guest sentiments which focus on apartment-type accommodations. The study successfully employed advanced analytical methods, including sentiment analysis, time-series trends, and predictive modelling to achieve a comprehensive understanding of how these factors affect guest satisfaction and consumer behaviour. The findings reveal that most reviews are positive with guests particularly appreciating aspects such as cleanliness, location, and convenience. These elements play a crucial role in shaping favourable guest experiences and overall satisfaction. Conversely, negative feedback often involves issues such as cleanliness and space, indicating areas for potential improvement.

The initial goal was to examine how socio-political events like Brexit and the COVID-19 pandemic influenced guest sentiments over time. The research was conducted by using DistilBERT for time-series analysis which revealed that these events led to notable shifts in guest perceptions, particularly heightened concerns about cleanliness during the pandemic. These findings align with the study's objective to explore how external factors influence guest perceptions demonstrating that consumer sentiments are highly sensitive to broader economic and political uncertainties.

Named Entity Recognition (NER) and Latent Dirichlet Allocation (LDA) were effectively used to categorise guest sentiments by host type, accommodation type, and neighbourhood, providing deeper insights into factors influencing guest reviews. Moreover, the insights gained from LDA and NER can also assist hosts in creating customised experiences for different guest demographics, such as families, business travellers or leisure seekers by aligning accommodations with their specific needs and interests. These targeted strategies not only improve guest satisfaction but also increase the likelihood of repeat bookings and positive word-of-mouth referrals.

Correlation analysis, enhanced by the machine learning model (SMOTEENN) effectively illustrated a strong relationship between positive sentiments and features such as reviews scores, location convenience, and room type. This finding supports the hypothesis that guests prioritise quality and experience over pricing which showed weak correlation with sentiment scores. The success of this model in addressing data imbalance highlights the importance of specialisation in achieving accurate insights. This achievement further emphasises the crucial role that exceptional service and strategic location play in boosting guest satisfaction.

The research reveals that while listing growth and pricing changes have minimal impact on sentiments, occupancy rates strongly correlate with guest satisfaction. Seasonal trends further underscore the importance of strategic occupancy management over pricing strategies. These findings suggest that availability and booking patterns are crucial for maintaining high levels of guest satisfaction and competitiveness in the Airbnb market. This points out the importance of optimising the strategies and offerings of hosts.

The successful integration of sentiment scores as exogenous variables into predictive models like SARIMAX significantly improve accuracy for pricing and demand in the Airbnb market. This approach equips hosts and stakeholders with strategic insights to adapt to market changes, enhancing decision-making and maintaining competitiveness in the dynamic sharing economy. The study concludes that leveraging qualitative data in quantitative models is crucial for future-proofing strategies in the evolving short-term rental industry.

The broader ethical considerations involve ensuring that the insights derived from this research are used to enhance guest experiences without exploiting consumer behaviour. The research also highlights the need for ongoing adaptation and iteration in response to changing socio-economic conditions to ensure

that both hosts and policymakers are equipped to maintain a sustainable and competitive Airbnb market in the long term.

Contributions to the Field

This research makes a significant contribution to the fields of hospitality and tourism by providing a nuanced understanding of the factors influencing guest satisfaction in Airbnb accommodations. The integration of sentiment analysis and machine learning techniques has been demonstrated to be valuable in capturing the complexity of guest feedback. By highlighting the importance of host quality, accommodation features, and the impact of external events, the findings offer actionable insights for Airbnb hosts, investors, and policymakers seeking to enhance guest experiences and optimise service delivery. Moreover, the study emphasises the need for personalised guest interactions and strategic market positioning in acquiring a better knowledge of customer behaviour in the sharing economy. As a result, this research enriches our comprehension of the dynamic interplay between service quality and guest satisfaction, paving the way for more targeted strategies in the industry.

Recommendation for future research

To deepen the understanding of guest experiences and preferences, future research should broaden the scope of sentiment analysis by incorporating non-English reviews, which can be initially translated using Python tools. As advancements in translation technology continue to emerge, incorporating more precise tools will further improve the accuracy of sentiment detection. Examining cultural and regional influences on guest sentiments could uncover patterns specific to different demographics, providing valuable insights for developing tailored strategies for hosts and policymakers.

Additionally, investigating the long-term effects of market dynamics and external events, such as economic fluctuations or global crises, on guest satisfaction will yield critical insights for strategic decision-making within the sharing economy. Researchers are encouraged to employ real-time data analytics and conduct more detailed sentiment analysis to capture the evolving preferences of consumers which can guide the adaptation of strategies to meet these changing demands. Furthermore, interdisciplinary studies that combine insights from behavioural science and economics have the potential to enhance our understanding of Airbnb guests' motivations and expectations.

References

Abeysinghe, P., & Bandara, T. (2022). A novel self-learning approach to overcome incompatibility on TripAdvisor reviews. *Data Science and Management,* 5(1), 1–10. https://doi.org/10.1016/j.dsm.2022.02.001

Beheshti, N. (2022, February 10). Guide to confusion matrices & classification performance metrics. *Medium*. https://towardsdatascience.com/guide-to-confusion-matrices-classification-performance-metrics-a0ebfc08408e

Bertsimas, D., & Dunn, J. (2017). Optimal classification trees. *Machine Learning, 106*(7), 1039-1082.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3,* 993–1022. https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th ed.). Wiley. https://www.wiley.com/en-us/Time+Series+Analysis%3A+Forecasting+and+Control%2C+5th+Edition-p-9781118675021

Britzolakis, A., Kondylakis, H., & Papadakis, N. (2020). A review on lexicon-based and machine learning political sentiment analysis using tweets.*Hellenic Mediterranean University*. https://www.researchgate.net/publication/350761338_A_Review_on_Lexicon-Based_and_Machine_Learning_Political_Sentiment_Analysis_Using_Tweets.

Buhalis, D., & Sinarta, Y. (2019). Real-time co-creation and nowness service: Lessons from tourism and hospitality. *Journal of Tourism Management, 72,* 232-245. https://doi.org/10.1016/j.tourman.2018.12.005

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://www.jair.org/index.php/jair/article/view/10302

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. https://doi.org/10.1145/2939672.2939785

Cocola-Gant, A. (2016). "Holiday Rentals: The New Gentrification Battlefront." *Sociological Research Online, 21*(3), 10.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://aclanthology.org/N19-1423/

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://arxiv.org/abs/1810.04805

Ert, E., & Fleischer, A. (2019). The evolution of trust in Airbnb: A case of home rental. *Annals of Tourism Research*, 75, 279-287.

European Parliament. (2023, June 8). EU AI Act: first regulation on artificial intelligence. *European Parliament.* https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence.

Garcia, D., & Wang, Y. (2020). Sentiment in the sharing economy: Evidence from Airbnb. *Tourism Management, 79*, 104095. https://doi.org/10.1016/j.tourman.2019.104095

Gov.UK. (2024, February 6). A pro-innovation Approach to AI regulation: Government Response. *GOV.UK*. https://www.gov.uk/government/consultations/ai-regulation-a-pro-innovation-approach-policy-proposals/outcome/a-pro-innovation-approach-to-ai-regulation-government-response

Guttentag, D. (2015). Airbnb: Disruptive innovation and the rise of an informal tourism accommodation sector. *Current Issues in Tourism, 18*(12), 1192–1217. https://doi.org/10.1080/13683500.2013.827159

Gustiyan Islahuzaman. (2023, August 29). *Mastering Imbalanced Classification: XGBoost and SMOTE + ENN on Bank Marketing Data With Python*. Medium; Medium. https://gustiyaniz.medium.com/mastering-imbalanced-classification-xgboost-and-smote-enn-on-bank-marketing-data-with-python-c53e4198a375

He, K., Zhang, X., Ren, S., & Sun, J. (2021). Deep learning for time series analysis: Principles and algorithms. *Journal of Statistical Science, 8*(2), 34–57.

Herring, E. (2018, December 12). London Airbnb statistics: Growth rate, average occupancy + more. *Finder UK*. https://www.finder.com/uk/mortgages/london-airbnb-statistics

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. Wiley. https://www.wiley.com/en-us/Applied+Logistic+Regression%2C+Third+Edition-p-9780470582473

How to become a Superhost - Airbnb Help Centre. (n.d.). Airbnb. https://www.airbnb.co.uk/help/article/829

Huggingface.co. (2024). https://huggingface.co/docs/transformers/en/model_doc/distilbert

Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, 216–225. https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109

Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts.

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research, 50*, 723–762.

Li, Y., Li, H., & Zhang, J. (2021). Regulating the sharing economy: Evidence from Airbnb. *Marketing Science, 40*(2), 307–331.

Li, Y., Li, H., & Zhang, J. (2023). The impact of online reviews on Airbnb pricing. *International Journal of Hospitality Management, 113*, 103450.

Liu, X., Li, X., & Zhu, H. (2021). Sentiment analysis: A comparison of traditional lexicon-based and machine learning approaches. *IEEE Access, 9*, 93273–93284. https://doi.org/10.1109/ACCESS.2021.3098438

London, England, United Kingdom. (2024, March 19). Inside Airbnb. https://insideairbnb.com/get-the-data/

Loria, S. (2018). TextBlob: Simplified text processing. *TextBlob Documentation*. https://textblob.readthedocs.io/en/dev/

Kim, J., Yoon, S., & Joo, Y. (2017). Examining the role of sentiment in online reviews: Evidence from the movie industry. *Electronic Commerce Research and Applications*, 21, 13-24.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal, 5*(4), 1093–1113.

Mustafa Abusalah. (2019, March 19). Re-sampling imbalanced training corpus for sentiment analysis. *Medium; Analytics Vidhya*. https://medium.com/analytics-vidhya/re-sampling-imbalanced-training-corpus-for-sentiment-analysis-c9dc97f9eae1

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*(1), 3–26. https://doi.org/10.1075/li.30.1.03nad

Narkhede, S. (2018, June 27). *Understanding AUC - ROC Curve*. Medium; Towards Data Science. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Narkhede, S. (2018). Understanding confusion matrix. *Medium; Towards Data Science*. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

OECD. (2016). Policies for the tourism sharing economy. *www.oecd-Ilibrary.org*, 89–120. https://doi.org/10.1787/tour-2016-7-en

Pine, B. J., & Gilmore, J. H. (1998). Welcome to the experience economy. *Harvard Business Review, 76*(4), 97–105.

Porter, M. E. (1985). *Competitive advantage: Creating and sustaining superior performance*. Free Press.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*. https://arxiv.org/abs/1910.01108

Santos, G., Mota, V. F. S., Benevenuto, F., & Silva, T. H. (2020). Neutrality may matter: Sentiment analysis in reviews of Airbnb, Booking, and Couchsurfing in Brazil and USA. *Social Network Analysis and Mining, 10*(1). https://doi.org/10.1007/s13278-020-00656-5

Shao, X., Chen, X., & Zhang, C. (2019). Comparative study of machine learning models for stock price prediction: A case study of listed companies in China. *Physica A: Statistical Mechanics and its Applications*, 531, 121754.

Shukla, A. (2020). Re-sampling imbalanced training corpus for sentiment analysis. *Medium; Analytics Vidhya*. Retrieved from https://medium.com/analytics-vidhya/re-sampling-imbalanced-training-corpus-for-sentiment-analysis-c9dc97f9eae1.

Song, H., & Li, G. (2008). Tourism demand modeling and forecasting—A review of recent research. *Tourism Management*, 29(2), 203-220.
Sönmez, S. F., Apostolopoulos, Y., & Tarlow, P. (1999). Tourism in crisis: Managing the effects of terrorism. *Journal of Travel Research*, 38(1), 13-18.

Suresh, M. (2019, August 13). *Vader : Customizing the library - Malavika Suresh - Medium*. Medium; Medium. https://medium.com/@malavika.suresh0794/vader-customizing-the-library-71d9e8ed6eda

World. (2019, December 19). *news-article*. Wttc.org; World Travel & Tourism Council. https://wttc.org/news-article/ai-set-to-revolutionise-travel-and-tourism-says-latest-wttc-report#:~:text=To%20promote%20sustainability%20and%20excellent.

World Tourism Organization (UNWTO). (2021). *Harnessing Big Data and Sentiment Analysis for Tourism Management*. UNWTO Publications. https://www.e-unwto.org

Wu, J., Fan, Z., & Zhao, W. (2020). Sentiment-driven demand prediction for short-term rental services: Evidence from Airbnb. *Journal of Business Research, 117*, 173–184. https://doi.org/10.1016/j.jbusres.2020.05.019

Yang, Y., Zhang, H., & Chen, X. (2020). Coronavirus pandemic and tourism: Dynamic stochastic general equilibrium modeling of infectious disease outbreak. *Annals of Tourism Research*, 83, 102912.

Zainal, G. (2020). Mastering imbalanced classification: XGBoost and SMOTE-ENN on bank marketing data with Python. *Medium*. https://gustiyaniz.medium.com/mastering-imbalanced-classification-xgboost-and-smote-enn-on-bank-marketing-data-with-python-c53e4198a375

Zvornicanin, E. (2021, September 3). *Choosing the best q and p from ACF and PACF plots in ARMA-type modeling | Baeldung on Computer Science*. Www.baeldung.com. https://www.baeldung.com/cs/acf-pacf-plots-arma-modeling

Appendix

Appendix A

BEMM466 Project Proposal - Sentiment Analysis of guest reviews for apartment-type Airbnb accommodation in London

Project objective

The primary aim of this research is to conduct a comprehensive analysis of the sentiments expressed in guest reviews for apartment-type Airbnb accommodations in London. By leveraging advanced sentiment analysis techniques, the study seeks to understand the distribution of positive, negative, and neutral sentiments, while also identifying the temporal trends that influence guest experiences. The research will explore the contextual dynamics of the Airbnb market in London, with a particular focus on forecasting future trends in pricing and demand based on the sentiment data.

To achieve these objectives, the research will employ both cross-sectional and time-series approaches. The cross-sectional analysis will examine sentiments at a specific point in time, providing a snapshot of the overall distribution of guest sentiments. In contrast, the time-series analysis will track sentiment trends over significant periods, offering insights into how guest experiences evolve in response to external events such as Brexit and the COVID-19 pandemic. By integrating these approaches, the research aims to deliver a comprehensive understanding of guest sentiments, capturing both the static and dynamic aspects of guest feedback. This dual approach will provide Airbnb hosts, policymakers, and stakeholders within the hospitality industry with actionable insights to enhance service quality, inform regulatory frameworks, and adapt to changing market conditions.

Research question

- **RQ 1:** What is the distribution of positive, negative, and neutral sentiments among reviews for apartment-type Airbnb accommodations in London, and what specific words or phrases are most frequently mentioned in these reviews?
- **RQ 2:** How do sentiment trends vary over time, particularly during significant events such as economic crisis (e.g., Brexit in UK) and COVID-19 pandemic, and are there notable differences in sentiment expressions during those specific phases?
- **RQ 3:** How do various factors influence the predominance of positive or negative evaluations in guest reviews for apartment-type Airbnb accommodations?
- **RQ 4:** To what extent specific features of apartment-type accommodations (e.g., room amenities, location convenience, pricing) correlate with the sentiments expressed in guest reviews?
- **RQ 5:** How do the growth and dynamics of the Airbnb market in London influence guest sentiments in apartment-type accommodations?
- **RQ 6:** How can sentiment analysis of guest reviews for apartment-type Airbnb accommodations in London be used to forecast future demand and pricing trends?

Problem statement

Airbnb has emerged as a significant player in the global hospitality industry, especially in urban environments like London (Guttentag, 2015). The city's diverse tourist population and the high demand for unique accommodations have resulted in a substantial increase in apartment-type Airbnb listings

(Li, Li, & Zhang, 2021). However, the subjective nature of guest reviews poses a challenge for systematic sentiment analysis to derive meaningful conclusions (Garcia & Wang, 2020).
The study utilises a comprehensive dataset comprising 1,987 listings and 33,169 reviews from apartment-type Airbnb accommodations in London, spanning from 2010 to 2024. . This research addresses the challenge of interpreting unstructured guest feedback through sentiment analysis, aiming to uncover key factors that drive guest satisfaction and dissatisfaction. It also seeks to identify trends over time, especially in relation to significant events like Brexit and the COVID-19 pandemic. Additionally, the study will explore the correlation between sentiment scores and specific apartment features, while forecasting future pricing and demand based on the analysed sentiments.

The findings of this research are highly relevant to multiple stakeholders:

- **Airbnb Hosts:** By understanding guest sentiments, hosts can improve their service quality, enhance guest satisfaction, and potentially increase their ratings and bookings.
- **Policymakers:** Insights into guest experiences can help in formulating regulations and supportive frameworks for short-term rentals.
- **Tourism Industry:** Enhanced guest experiences can contribute to a positive perception of London among international visitors, boosting the city's tourism economy.

Literature and Data Review

This research will draw upon a range of academic articles, books, and business sources to explore the intersection of sentiment analysis and the Airbnb market in London. Key literature includes Guttentag (2015), which examines the disruptive impact of Airbnb on traditional hospitality sectors, providing a foundational understanding of Airbnb's role in the sharing economy. Li, Li, and Zhang (2021) offer insights into the diversity of Airbnb accommodations emphasising the platform's appeal to a broad spectrum of travellers. Garcia and Wang (2020) delve into the challenges of analysing unstructured guest reviews, highlighting the need for sophisticated analytical tools in sentiment analysis.

To address these challenges, the study will utilise advanced Natural Language Processing (NLP) techniques as outlined in works by Hutto and Gilbert (2014) and Devlin et al. (2019). These include the application of VADER for lexicon-based sentiment analysis and DistilBERT for context-aware sentiment analysis, both of which are well-suited to handling the nuances of Airbnb review data. Additionally, Named Entity Recognition (NER) will be employed to extract specific entities mentioned in reviews, following methodologies established by Nadeau and Sekine (2007). Latent Dirichlet Allocation (LDA), as described by Blei, Ng, and Jordan (2003), will be used for topic modelling to identify recurring themes in guest feedback.

The study will also incorporate time-series analysis techniques such as ARIMA and SARIMAX, as detailed by Box et al. (2015) and He et al. (2021), to analyse sentiment trends over time and forecast future pricing and demand. These models will help to understand how sentiments correlate with market dynamics and significant events like Brexit and the COVID-19 pandemic.
The primary data source for this research will be a comprehensive dataset provided by Inside Airbnb (Inside Airbnb, n.d.), which includes 1,987 listings and 33,169 reviews from apartment-type Airbnb accommodations in London, spanning from 2010 to 2024. This dataset will enable the examination of sentiment trends over time, particularly in relation to significant events.

In summary, this research will leverage a combination of established literature and advanced analytical techniques to explore guest sentiments and market dynamics in the London Airbnb market, offering valuable insights for both academic and industry audiences.

Methodology

Data Overview

This study utilises a comprehensive dataset comprising 33,169 reviews and 1,987 listings from 2010 to 2024.

Table 1. Data overview information

Data sources	Original data	Descriptions
Listings data	listing.csv	This data includes context, and features related to each review, such as room amenities, location, and other listing-specific information.
Reviews data	reviews.csv	The data source of reviews from guests which related to listing id.

Data preprocessing

Some data from Inside Airbnb exhibits noise, inconsistencies, and missing values, which can potentially impact the accuracy of sentiment analysis. To address these issues effectively, a comprehensive data cleansing plan will be implemented as Figure 1.

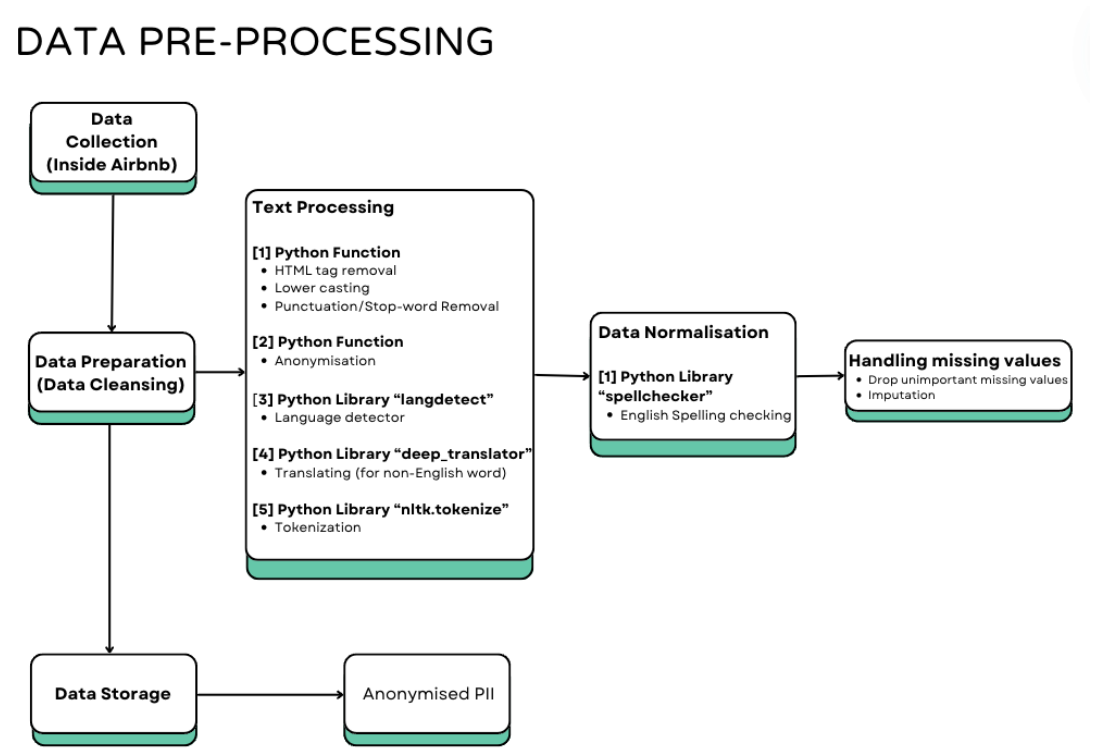


Figure 1. Data preprocessing flow

Research Design

The research design for this study is structured to systematically analyse and forecast the sentiments expressed in guest reviews for apartment-type Airbnb accommodations in London. The design incorporates both quantitative and qualitative methods to ensure a comprehensive understanding of the

data and provide a holistic view of guest sentiments. The quantitative aspect involves the use of sentiment analysis tools to classify and quantify sentiments expressed in the reviews. The qualitative aspect includes the application of Named Entity Recognition (NER) and topic modelling to uncover deeper insights into the themes and entities discussed in the reviews. Correlation analysis examines relationships between market factors such as pricing and occupancy rates, while seasonal decomposition assesses the influence of market dynamics on sentiment trends. Predictive modelling, including ARIMA and SARIMAX, forecasts pricing and demand trends.

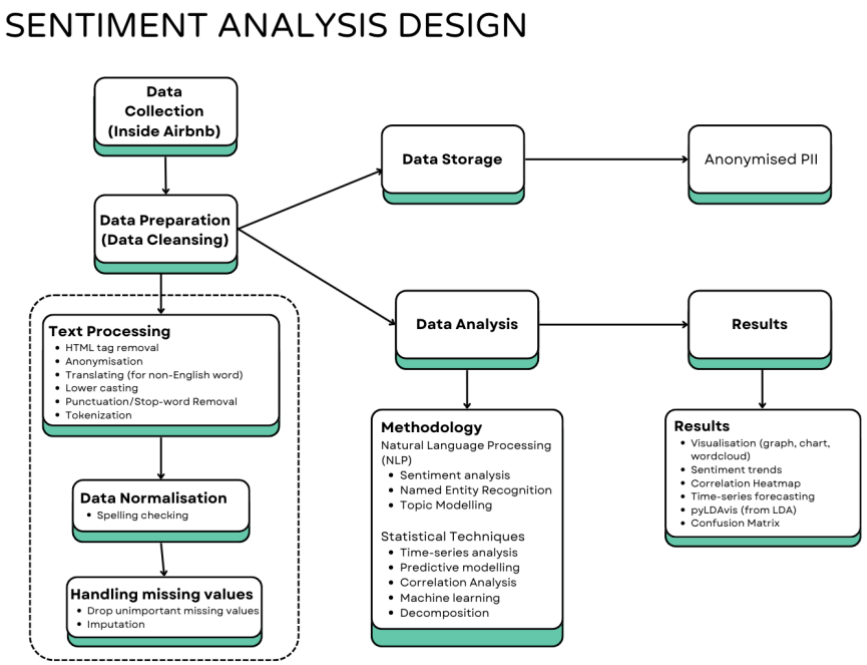


Figure 2. Sentiment Analysis project research design

Quantitative Analysis

The quantitative analysis employs sentiment analysis to evaluate attitudes, opinions, and emotions expressed in online Airbnb reviews. Initially, VADER (Valence Aware Dictionary and Sentiment Reasoner) and TextBlob will be compared to determine the most effective tool for classifying sentiments into positive, negative, and neutral categories. A custom lexicon may be developed to enhance the accuracy of sentiment classification. Additionally, DistilBERT will be utilised to analyse sentiment trends over time, particularly during significant events such as Brexit and the COVID-19 pandemic.

Table 2. Research question 1 methodology and approach

Methodology	**Descriptions**
Python Libraries	Pandas, numpy - data manipulation nltk ,re - text pre-processing TextBlob, VADER - sentiment analysis
Classification	Positive: Sentiment score > 0.05 Negative: Sentiment score < -0.05 Neutral: Sentiment score between -0.05 and 0.05
Implementation	Use Python with the NLTK library, which includes the VADER sentiment analysis tool.

Procedure	- Preprocess the review text by converting it to lowercase, removing punctuation, and tokenizing. - Apply the VADER sentiment analysis tool to each review. - Classify the sentiment scores based on the thresholds in classification.
Visualisation	Pie chart represents sentiment distribution, word cloud for frequent positive and negative words.

Table 3. Research question 2 methodology and approach

Methodology	**Descriptions**
Python Libraries	Pandas, numpy - data manipulation nltk ,re - text pre-processing VADER - sentiment analysis Transformer, torch - DistilBERT model
Classification	Positive: Sentiment score > 0.05 Negative: Sentiment score < -0.05 Neutral: Sentiment score between -0.05 and 0.05
Implementation	- Calculate the proportion of each sentiment category (positive, negative, neutral). - Segment reviews into relevant periods (pre-Brexit, during Brexit, pre-COVID-19, during COVID-19, post-COVID-19). - Analyse trends by examining the content of the evaluations and the distribution of sentiments across different time periods or significant events.
Procedure	- Fine-tune the BERT model for sentiment analysis to capture dependencies between words in a sentence or paragraph. - Conduct time-series analysis on the classified sentiment data to identify specific trends in sentiment over time.
Visualisation	Line graph with specific period of significant events (COVID-19 and Brexit).

Qualitative Analysis

The qualitative analysis is divided into three parts to gain deeper insights into guest experiences. Firstly, Sentiment Distribution by Host Type employs Named Entity Recognition (NER) to identify and classify entities such as host names and locations. This method enables the analysis of patterns in guest perceptions of Superhosts compared to regular hosts, revealing areas for service enhancement. Secondly, Sentiment Distribution by Accommodation Type utilises Latent Dirichlet Allocation (LDA) and Topic Modelling to identify recurring themes. This approach elucidates differences in sentiment across various accommodation types, providing insights into how hosts can align their offerings with guest preferences. Lastly, Sentiment Distribution by Neighbourhood applies Linear Discriminant Analysis (LDA) and Topic Modelling to assess how geographical factors influence guest evaluations.

Table 4. Research question 3 methodology and approach

Methodology	Descriptions
Python Libraries	Pandas, numpy - data manipulation nltk ,re - text pre-processing VADER - sentiment analysis Spacy - for Named Entity Recognition (NER) CountVectorizer - Latent Dirichlet Allocation (LDA) document-term matrix
Classification	Positive: Sentiment score > 0.05 Negative: Sentiment score < -0.05 Neutral: Sentiment score between -0.05 and 0.05
Sentiment by type of host	
Implementation	- Apply NER to enhance the understanding of host mentions in the reviews.
Visualisation	Pie chart - distribution of sentiment Bar chart - identify frequency of entity types in reviews
Sentiment by type of accommodation	
Implementation	- Use CountVectorizer to create the document-term matrix. - Fit an LDA model to identify latent topics. - Display the top words for each topic to understand common themes in the reviews
Visualisation	- Visualise as document-matrix for each topic - Visualise Topics using interactive pyLDAvis
Sentiment by type of neighbourhood	
Implementation	- Use CountVectorizer to create the document-term matrix. - Fit an LDA model to identify latent topics. - Display the top words for each topic to understand common themes in the reviews
Visualisation	- Visualise as document-matrix for each topic - Visualise Topics using interactive pyLDAvis

Correlation Analysis

Correlation analysis involves statistical methods, including chi-square tests and machine learning approaches, which are used to analyse correlations between sentiment scores and apartment features such as cleanliness, location, and amenities. This approach helps identify significant relationships and patterns within the data providing insights into how specific features impact guest sentiments.

Table 5. Research question 4 methodology and approach

Methodology	Descriptions
Python Libraries	Pandas, numpy - data manipulation nltk ,re - text pre-processing VADER - sentiment analysis Chi2 - for chi-square testing
Classification	Positive: Sentiment score > 0.05 Negative: Sentiment score < -0.05 Neutral: Sentiment score between -0.05 and 0.05
Implementation	- Perform Chi-Square tests to reveal associations between sentiment and categorical features. - Perform various machine learning models to identify the best model capturing correlations between sentiment score and features.
Machine learning models	- Logistic Regression - Decision Tree and Random Forest - SMOTE (Synthetic Minority - Over-sampling Technique) - GradientBoostingClassifier - BalancedRandomForestClassifier - SMOTE + XGBoost - Adaptive Synthetic Sampling (ADASYN) - SMOTEENN (Synthetic Minority - Over-sampling Technique + Edited Nearest Neighbors) - SMOTETomek
Visualisation	- Table summary for chi-square statistic values - Confusion matrix for each machine learning model - Bar chart for feature importance scores (correlation) - ROC-AUC curve

Market Dynamic Analysis

An analysis of market dynamics in Airbnb, including listing growth, pricing changes, and occupancy rates, reveals how these factors influence guest sentiments toward apartment accommodations in London.

Table 6. Research question 5 methodology and approach

Methodology	Descriptions
Python Libraries	Pandas, numpy - data manipulation nltk ,re - text pre-processing VADER - sentiment analysis statsmodels - for time series analysis
Classification	Positive: Sentiment score > 0.05 Negative: Sentiment score < -0.05

	Neutral: Sentiment score between -0.05 and 0.05
Implementation	- Analyse the Increase in the Number of Listings and analyse Changes in Pricing - Analyse Occupancy Rates and correlation between sentiment scores - Use decomposition of time series analysis to interpret seasonal/event analysis
Visualisation	- Line graph - Correlation heatmaps - Decomposition results (as line graph)

Predicting and forecasting

This part of the analysis will initially focus on forecasting pricing trends by utilising sentiment analysis of guest reviews. By examining guest feedback, sentiment analysis can be employed to predict future pricing trends and offer insights into how consumer perceptions influence pricing dynamics. Following this, the analysis will shift towards demand forecasting, using sentiment data to assess potential fluctuations in booking rates and market demand.

Table 7. Research question 6 methodology and approach

Methodology	**Descriptions**
Python Libraries	Pandas, numpy - data manipulation nltk ,re - text pre-processing VADER - sentiment analysis statsmodels - for time series analysis
Classification	Positive: Sentiment score > 0.05 Negative: Sentiment score < -0.05 Neutral: Sentiment score between -0.05 and 0.05
Pricing forecasting	
Implementation	- Implement ARIMA models to capture time-dependent structures and seasonal patterns in historical pricing data. - Enhance accuracy by integrating ARIMA withHistGradientBoostingRegressor. - Evaluate the model performance by comparing MSE, RMSE, MAE and R-square.
Visualisation	- Generate time-series plots to illustrate historical and forecasted pricing trends as line graphs.
Demand forecasting	
Implementation	- Apply ARIMA models to forecast booking trends and assess demand fluctuations over specified time periods. - Use SARIMAX to incorporate sentiment data into demand forecasting models, improving the accuracy of future market demand predictions.

	- Evaluate the model performance by comparing ADF statistics, AIC, BIC and coefficients.
Visualisation	- Generate time-series plots to illustrate historical and forecasted demand trends as line graphs.

Limitations of study

This study acknowledges several limitations that must be considered when interpreting the findings. The dataset comprises both English and non-English reviews. Although non-English reviews will be translated, potential inaccuracies in machine translation may influence the sentiment analysis outcomes. Additionally, the data sourced from Inside Airbnb may have limitations regarding completeness and accuracy which could affect the overall reliability of the study. While sentiment analysis tools, including those with custom implementations are employed, they may not fully capture the complexities and nuances of real-world review tones, potentially leading to biassed interpretations. Furthermore, the dataset contains a higher proportion of positive reviews, presenting challenges for machine learning models. Despite the application of techniques to address class imbalance, the predictive accuracy for negative and neutral sentiments may still be compromised.

Ethical Challenges

This research presents several ethical challenges, particularly concerning data privacy and confidentiality. Airbnb reviews may include personal information that must be carefully managed to protect individuals' privacy. To address this, data anonymization techniques will be employed to protect personal information, ensuring that no personally identifiable information (PII) can be traced back to specific individuals. Additionally, there is a potential for bias in sentiment analysis tools. To mitigate this, the research will cross-validate results using multiple sentiment analysis tools and perform regular audits to ensure fair representation. The study adheres to ethical guidelines and regulations to maintain the integrity and responsibility of data handling and analysis.

Strict ethical guidelines will be followed throughout the study to ensure responsible data handling. The data will be utilised exclusively for this research project, adhering to strict protocols for secure handling and storage to prevent unauthorised access. Upon the completion of the research and subsequent graduation, all data will be securely destroyed, employing secure deletion methods for digital files. The use of AI tools, such as ChatGPT and Blackbox, for generating code and guiding sentiment analysis tools, will be carefully documented and validated by human researchers to ensure accuracy and reliability.

Project timeline

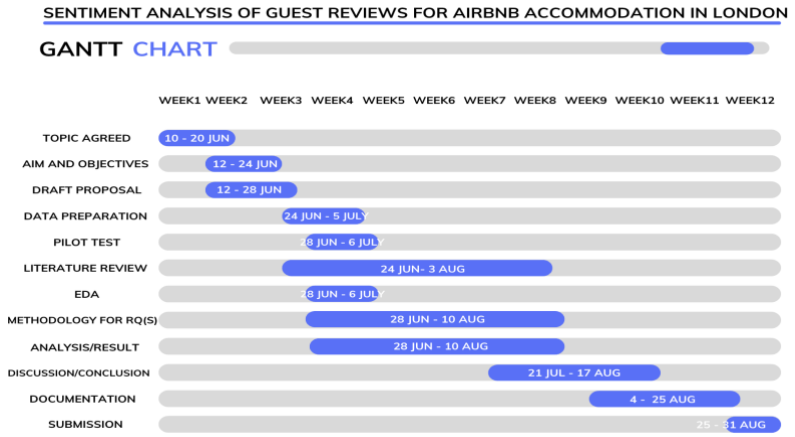


Figure 3. Project timeline – Gantt chart

Reference

1. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research, 3*(Jan), 993-1022.
2. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.
3. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171-4186. https://doi.org/10.18653/v1/N19-1423
4. Garcia, D., & Wang, Y. (2020). Sentiment in the sharing economy: Evidence from Airbnb. *Tourism Management, 79*, 104095. https://doi.org/10.1016/j.tourman.2020.104095
5. Guttentag, D. (2015). Airbnb: Disruptive innovation and the rise of an informal tourism accommodation sector. *Current Issues in Tourism, 18*(12), 1192-1217. https://doi.org/10.1080/13683500.2013.827159
6. He, K., Zhang, X., Ren, S., & Sun, J. (2021). Deep Learning for Time Series Analysis: Principles and Algorithms. *Journal of Statistical Science, 8*(2), 34-57.
7. Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media (ICWSM-14)*, 216-225.
8. Li, Y., Li, H., & Zhang, J. (2021). Regulating the Sharing Economy: Evidence from Airbnb. *Marketing Science, 40*(2), 307-331.
9. Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*(1), 3-26. https://doi.org/10.1075/li.30.1.03nad
10. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108.* https://arxiv.org/abs/1910.01108

Appendix B

GitHub Link:
https://github.com/Babyaimy/BEMM466_sentiment_analysis_Chonchaya

Data preprocessing from 'reviews.csv' to 'spelling_corrected_reviews.csv' (final version of dataset)

Step 1 : translate non-English comment (test4.py)

test4.py > ...

```python
import pandas as pd
from langdetect import detect, LangDetectException
from deep_translator import GoogleTranslator
from concurrent.futures import ThreadPoolExecutor, as_completed
import time
import logging

# Set up logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')

# Load the translated CSV file
translated_file_path = 'translated_reviews.csv'  # Ensure this path is correct
translated_reviews_df = pd.read_csv(translated_file_path)

# Supported languages map for GoogleTranslator
supported_languages = {
    'af': 'af', 'sq': 'sq', 'am': 'am', 'ar': 'ar', 'hy': 'hy', 'as': 'as', 'ay': 'ay', 'az': 'az', 'bm': 'bm',
    'eu': 'eu', 'be': 'be', 'bn': 'bn', 'bho': 'bho', 'bs': 'bs', 'bg': 'bg', 'ca': 'ca', 'ceb': 'ceb', 'ny': 'ny',
    'zh-CN': 'zh-CN', 'zh-TW': 'zh-TW', 'co': 'co', 'hr': 'hr', 'cs': 'cs', 'da': 'da', 'dv': 'dv', 'doi': 'doi',
    'nl': 'nl', 'en': 'en', 'eo': 'eo', 'et': 'et', 'ee': 'ee', 'tl': 'tl', 'fi': 'fi', 'fr': 'fr', 'fy': 'fy',
    'gl': 'gl', 'ka': 'ka', 'de': 'de', 'el': 'el', 'gn': 'gn', 'gu': 'gu', 'ht': 'ht', 'ha': 'ha', 'haw': 'haw',
    'iw': 'iw', 'hi': 'hi', 'hmn': 'hmn', 'hu': 'hu', 'is': 'is', 'ig': 'ig', 'ilo': 'ilo', 'id': 'id', 'ga': 'ga',
    'it': 'it', 'ja': 'ja', 'jw': 'jw', 'kn': 'kn', 'kk': 'kk', 'km': 'km', 'rw': 'rw', 'gom': 'gom', 'ko': 'ko',
    'kri': 'kri', 'ku': 'ku', 'ckb': 'ckb', 'ky': 'ky', 'lo': 'lo', 'la': 'la', 'lv': 'lv', 'ln': 'ln', 'lt': 'lt',
    'lg': 'lg', 'lb': 'lb', 'mk': 'mk', 'mai': 'mai', 'mg': 'mg', 'ms': 'ms', 'ml': 'ml', 'mt': 'mt', 'mi': 'mi',
    'mr': 'mr', 'mni-Mtei': 'mni-Mtei', 'lus': 'lus', 'mn': 'mn', 'my': 'my', 'ne': 'ne', 'no': 'no', 'or': 'or',
    'om': 'om', 'ps': 'ps', 'fa': 'fa', 'pl': 'pl', 'pt': 'pt', 'pa': 'pa', 'qu': 'qu', 'ro': 'ro', 'ru': 'ru',
    'sm': 'sm', 'sa': 'sa', 'gd': 'gd', 'nso': 'nso', 'sr': 'sr', 'st': 'st', 'sn': 'sn', 'sd': 'sd', 'si': 'si',
    'sk': 'sk', 'sl': 'sl', 'so': 'so', 'es': 'es', 'su': 'su', 'sw': 'sw', 'sv': 'sv', 'tg': 'tg', 'ta': 'ta',
    'tt': 'tt', 'te': 'te', 'th': 'th', 'ti': 'ti', 'ts': 'ts', 'tr': 'tr', 'tk': 'tk', 'ak': 'ak', 'uk': 'uk',
    'ur': 'ur', 'ug': 'ug', 'uz': 'uz', 'vi': 'vi', 'cy': 'cy', 'xh': 'xh', 'yi': 'yi', 'yo': 'yo', 'zu': 'zu'
}

# Function to detect language of comments
def detect_language(comment):
    try:
        return detect(comment)
    except LangDetectException:
        return 'unknown'

# Function to map detected language to supported language
def map_supported_language(lang):
    return supported_languages.get(lang, 'en')

# Function to translate non-English comments
def translate_comment(comment):
    try:
        lang = detect(comment)
        mapped_lang = map_supported_language(lang)
        if mapped_lang != 'en':
            translated = GoogleTranslator(source=mapped_lang, target='en').translate(comment)
            return translated
        else:
            return comment
    except LangDetectException:
        return comment

# Detect and translate non-English comments
translated_reviews_df['language'] = translated_reviews_df['comments'].apply(detect_language)
non_english_comments_df = translated_reviews_df[translated_reviews_df['language'] != 'en']

```

```python
# Use ThreadPoolExecutor to parallelize the translation process
def translate_and_update(comment, index):
    translated_comment = translate_comment(comment)
    translated_reviews_df.at[index, 'comments'] = translated_comment

total_comments = len(non_english_comments_df)
processed_comments = 0

with ThreadPoolExecutor(max_workers=10) as executor:
    futures = [executor.submit(translate_and_update, comment, index) for index, comment in non_english_comments_df['comments'].items()]

    start_time = time.time()
    last_logged_time = start_time

    for future in as_completed(futures):
        future.result()  # Ensure all futures are completed
        processed_comments += 1

        current_time = time.time()
        elapsed_time = current_time - last_logged_time

        if elapsed_time >= 60 or processed_comments == total_comments:  # Log progress every minute or when done
            print(f"Processed {processed_comments} out of {total_comments} comments.")
            logging.info(f"Processed {processed_comments} out of {total_comments} comments.")
            last_logged_time = current_time

# Drop the language column and save the updated DataFrame to a new CSV file
translated_reviews_df.drop(columns=['language'], inplace=True)
updated_file_path = 'updated_translated_reviews.csv'
translated_reviews_df.to_csv(updated_file_path, index=False)

print(f"Updated translated reviews have been saved to {updated_file_path}")
```

Step 2 : The result from test4.py still faced problem with chinese language, so this file need to re-translate again (test7.py)

test7.py > ...

```python
import pandas as pd
from langdetect import detect, LangDetectException
from deep_translator import GoogleTranslator
from concurrent.futures import ThreadPoolExecutor, as_completed

# Load the original dataset with non-English comments
original_file_path = 'updated_translated_reviews.csv'  # Update with your actual file path if needed
original_df = pd.read_csv(original_file_path)

# Load the translated non-English comments CSV file
translated_non_english_comments_file_path = 'translated_non_english_comments.csv'  # Update with your actual file path if needed
translated_non_english_comments_df = pd.read_csv(translated_non_english_comments_file_path)

# Function to detect the language of comments
def detect_language(comment):
    try:
        return detect(comment)
    except LangDetectException:
        return 'unknown'

# Check for non-English comments in the translated file
translated_non_english_comments_df['language'] = translated_non_english_comments_df['comments'].apply(detect_language)

# Filter out non-English comments
non_english_comments_df = translated_non_english_comments_df[translated_non_english_comments_df['language'] != 'en']

# Count non-English comments
non_english_comments_count = non_english_comments_df.shape[0]
print(f"Number of non-English comments still present: {non_english_comments_count}")


# Function to map detected language to supported language
def map_supported_language(lang):
    language_map = {
        'zh-cn': 'zh-CN',
        'zh-tw': 'zh-TW',
        'he': 'iw',  # Map Hebrew to the supported language code
        # Add other mappings if needed
    }
    return language_map.get(lang, lang)
```

```python

# Function to translate non-English comments
def translate_comment(comment):
    try:
        lang = detect(comment)
        mapped_lang = map_supported_language(lang)
        if mapped_lang != 'en':
            translated = GoogleTranslator(source=mapped_lang, target='en').translate(comment)
            return translated
        else:
            return comment
    except LangDetectException:
        return comment

# Re-translate non-English comments
def translate_and_update(comment, index):
    translated_comment = translate_comment(comment)
    translated_non_english_comments_df.at[index, 'comments'] = translated_comment

total_comments = len(non_english_comments_df)
processed_comments = 0

with ThreadPoolExecutor(max_workers=10) as executor:
    futures = [executor.submit(translate_and_update, comment, index) for index, comment in non_english_comments_df['comments'].items()

    for future in as_completed(futures):
        future.result()  # Ensure all futures are completed
        processed_comments += 1
        if processed_comments % 100 == 0 or processed_comments == total_comments:
            print(f"Processed {processed_comments} out of {total_comments} comments.")


# Update the original dataset with the re-translated comments
for index, row in translated_non_english_comments_df.iterrows():
    original_df.at[index, 'comments'] = row['comments']

# Save the updated dataset to a new CSV file
updated_file_path = 'final_updated_reviews.csv'
original_df.to_csv(updated_file_path, index=False)

print(f"Final updated dataset has been saved to {updated_file_path}")

# Re-check for any remaining non-English comments
original_df['language'] = original_df['comments'].apply(detect_language)
final_non_english_comments_df = original_df[original_df['language'] != 'en']
final_non_english_comments_count = final_non_english_comments_df.shape[0]
print(f"Number of non-English comments after re-translation: {final_non_english_comments_count}")

# Save any remaining non-English comments for inspection
final_non_english_comments_df.to_csv('remaining_non_english_comments.csv', index=False)
```

Step 3 : The result from test7.py still faced problem, so this will be re-translated with another approach.

test9.py > ...

```python
import pandas as pd
from langdetect import detect, LangDetectException
from deep_translator import GoogleTranslator
import time

# Load the two CSV files
updated_translated_reviews_file_path = 'updated_translated_reviews.csv'  # Update with your actual file path if needed
final_updated_reviews_file_path = 'final_updated_reviews.csv'  # Update with your actual file path if needed

updated_translated_reviews_df = pd.read_csv(updated_translated_reviews_file_path)
final_updated_reviews_df = pd.read_csv(final_updated_reviews_file_path)

# Merge the dataframes on common columns (id and reviewer_id are assumed to be common identifiers)
merged_df = updated_translated_reviews_df.merge(final_updated_reviews_df, on=['id', 'reviewer_id'], suffixes=('_updated', '_final'))

# Identify rows where the comments differ
different_comments_df = merged_df[merged_df['comments_updated'] != merged_df['comments_final']]

# Display the rows with different comments
print(f"Number of differing comments: {different_comments_df.shape[0]}")
print(different_comments_df[['id', 'reviewer_id', 'comments_updated', 'comments_final']])

# Function to detect the language of comments
def detect_language(comment):
    try:
        return detect(comment)
    except LangDetectException:
        return 'unknown'
```

```python

# Function to map detected language to supported language
def map_supported_language(lang):
    language_map = {
        'zh-cn': 'zh-CN',
        'zh-tw': 'zh-TW',
        'he': 'iw',  # Map Hebrew to the supported language code
        # Add other mappings if needed
    }
    return language_map.get(lang, lang)

# Function to translate non-English comments
def translate_comment(comment):
    try:
        lang = detect(comment)
        mapped_lang = map_supported_language(lang)
        if mapped_lang != 'en':
            translated = GoogleTranslator(source=mapped_lang, target='en').translate(comment)
            return translated
        else:
            return comment
    except LangDetectException:
        return comment

# Translate the differing comments in the updated file
total_comments = different_comments_df.shape[0]
processed_comments = 0

start_time = time.time()
last_logged_time = start_time

for index, row in different_comments_df.iterrows():
    translated_comment = translate_comment(row['comments_updated'])
    final_updated_reviews_df.loc[final_updated_reviews_df['id'] == row['id'], 'comments'] = translated_comment
    processed_comments += 1

    current_time = time.time()
    elapsed_time = current_time - last_logged_time

    if elapsed_time >= 60:  # Log progress every minute
        print(f"Processed {processed_comments} out of {total_comments} comments.")
        last_logged_time = current_time

# Save the corrected dataset to a new CSV file
corrected_file_path = 'corrected_final_updated_reviews.csv'
final_updated_reviews_df.to_csv(corrected_file_path, index=False)

print(f"Corrected dataset has been saved to {corrected_file_path}")
```

Step 4 : The result from test9.py successfully translate to English but have some problem with indexing. ‘test10.py’ is introduced to solve the index issue.

test10.py > ...

```python
import pandas as pd
from langdetect import detect, LangDetectException
from deep_translator import GoogleTranslator
import time

# Load the corrected final updated reviews CSV file
corrected_final_updated_reviews_file_path = 'corrected_final_updated_reviews.csv'  # Update with your actual file path if needed
corrected_final_updated_reviews_df = pd.read_csv(corrected_final_updated_reviews_file_path)

# Function to detect the language of comments
def detect_language(comment):
    try:
        return detect(comment)
    except LangDetectException:
        return 'unknown'

# Detect languages in the comments
corrected_final_updated_reviews_df['language'] = corrected_final_updated_reviews_df['comments'].apply(detect_language)

# Identify and count remaining non-English comments
non_english_comments_df = corrected_final_updated_reviews_df[corrected_final_updated_reviews_df['language'] != 'en']
non_english_comments_count = non_english_comments_df.shape[0]
print(f"Number of non-English comments still present: {non_english_comments_count}")

# Display the count of each language
language_counts = non_english_comments_df['language'].value_counts()
print("Languages still present and their counts:")
print(language_counts)
```

```python

# Function to map detected language to supported language
def map_supported_language(lang):
    language_map = {
        'zh-cn': 'zh-CN',
        'zh-tw': 'zh-TW',
        'he': 'iw',  # Map Hebrew to the supported language code
        # Add other mappings if needed
    }
    return language_map.get(lang, lang)

# Function to translate non-English comments
def translate_comment(comment):
    try:
        lang = detect(comment)
        mapped_lang = map_supported_language(lang)
        if mapped_lang != 'en':
            translated = GoogleTranslator(source=mapped_lang, target='en').translate(comment)
            return translated
        else:
            return comment
    except LangDetectException:
        return comment

# Translate the remaining non-English comments
total_comments = non_english_comments_df.shape[0]
processed_comments = 0

start_time = time.time()
last_logged_time = start_time

for index, row in non_english_comments_df.iterrows():
    translated_comment = translate_comment(row['comments'])
    corrected_final_updated_reviews_df.loc[corrected_final_updated_reviews_df['id'] == row['id'], 'comments'] = translated_comment
    processed_comments += 1

    current_time = time.time()
    elapsed_time = current_time - last_logged_time

    if elapsed_time >= 60:  # Log progress every minute
        print(f"Processed {processed_comments} out of {total_comments} comments.")
        last_logged_time = current_time

# Save the updated dataset to a new CSV file
final_corrected_file_path = 'final_corrected_updated_reviews.csv'
corrected_final_updated_reviews_df.to_csv(final_corrected_file_path, index=False)

print(f"Updated dataset with re-translated comments has been saved to {final_corrected_file_path}")

# Re-check for any remaining non-English comments
corrected_final_updated_reviews_df['language'] = corrected_final_updated_reviews_df['comments'].apply(detect_language)
final_non_english_comments_df = corrected_final_updated_reviews_df[corrected_final_updated_reviews_df['language'] != 'en']
final_non_english_comments_count = final_non_english_comments_df.shape[0]
print(f"Number of non-English comments after re-translation: {final_non_english_comments_count}")

# Save any remaining non-English comments for inspection
final_non_english_comments_df.to_csv('remaining_non_english_comments_after_retranslation.csv', index=False)
```

Step 5 : Convert all reviews to lower case and remove punctuation (cleaned_reviews.py)

clean_reviews.py > clean_text

```python
import pandas as pd
import string

# Load the final corrected updated reviews CSV file
final_corrected_updated_reviews_file_path = 'final_corrected_updated_reviews.csv'  # Update with your actual file path if needed
final_corrected_updated_reviews_df = pd.read_csv(final_corrected_updated_reviews_file_path)

# Function to convert text to lowercase and remove punctuation
def clean_text(text):
    if not isinstance(text, str) or text.strip() == '':
        return 'no comment'

    # Convert to lowercase
    text = text.lower()

    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))

    return text

# Apply the cleaning function to the 'comments' column
final_corrected_updated_reviews_df['comments'] = final_corrected_updated_reviews_df['comments'].apply(clean_text)

# Save the updated dataframe to a new CSV file
cleaned_file_path = 'cleaned_final_corrected_reviews.csv'
final_corrected_updated_reviews_df.to_csv(cleaned_file_path, index=False)

print(f"Cleaned dataset has been saved to {cleaned_file_path}")
```

Step 6 : Perform spelling check via Spellchecker python library and save to the file 'spelling_corrected_reviews.csv' (spell_checker.py).

spell_checker.py > ...

```python
import pandas as pd
from spellchecker import SpellChecker
import time

# Load the cleaned reviews CSV file
cleaned_file_path = 'cleaned_final_corrected_reviews.csv'  # Update with your actual file path if needed
cleaned_reviews_df = pd.read_csv(cleaned_file_path)

# Initialize the spell checker outside the loop for efficiency
spell = SpellChecker()

# Function to correct spelling errors
def correct_spelling(text):
    if not isinstance(text, str) or text.strip() == '':
        return 'no comment'

    # Correct spelling errors
    corrected_words = []
    for word in text.split():
        if word:
            corrected_word = spell.correction(word)
            if corrected_word:
                corrected_words.append(corrected_word)
            else:
                corrected_words.append(word)
        else:
            corrected_words.append('')

    return " ".join(corrected_words)

# Measure time for the entire dataset
start_time = time.time()
```

```
33
34  # Track progress
35  total_rows = cleaned_reviews_df.shape[0]
36  progress_interval = 100  # Print progress every 100 rows
37
38  # Apply the spelling correction function to the 'comments' column with progress tracking
39  for i, row in cleaned_reviews_df.iterrows():
40      # Safeguard against None values
41      comment = row['comments'] if pd.notnull(row['comments']) else 'no comment'
42      cleaned_reviews_df.at[i, 'comments'] = correct_spelling(comment)
43
44      if (i + 1) % progress_interval == 0 or (i + 1) == total_rows:
45          print(f"Processed {i + 1} / {total_rows} rows")
46
47  end_time = time.time()
48  total_elapsed_time = end_time - start_time
49
50  # Save the updated dataframe to a new CSV file
51  spelling_corrected_file_path = 'spelling_corrected_reviews.csv'
52  cleaned_reviews_df.to_csv(spelling_corrected_file_path, index=False)
53
54  print(f"Spelling corrected dataset has been saved to {spelling_corrected_file_path}")
55  print(f"Total processing time: {total_elapsed_time / 60:.2f} minutes")
```