MINI PROJECT REPORT

ON

HOTEL REVIEW CLASSIFICATION

SELVI V

RA2332241020020

Submitted to the

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS (MCA)

Under the guidance of

Dr. AGUSTHIYAR R, MCA., M. Phil., Ph. D.,

(Professor, Department of Computer Science and Applications)

MASTER OF COMPUTER APPLICATIONS

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Ramapuram, Chennai - 89

OCTOBER 2024

FACULTY OF SCIENCE AND HUMANITIES

Ramapuram, Chennai

Department of Computer Science and Applications (MCA)

BONAFIDE CERTIFICATE

Certified that this Mini Project report titled **HOTEL REVIEW CLASSIFICATION** is the bonafide work of **SELVI V (RA2332241020020)** who carried out the Mini Project work done under my supervision.

Signature of Internal Guide **Signature of Head of the Department**

Signature of Internal Examiner **Signature of External Examiner**

TABLE OF CONTENTS

ABSTRACT

In the era of digital information, customer feedback in the form of online reviews has become a crucial resource for businesses, especially in the hospitality industry. This project presents a comprehensive sentiment analysis of hotel reviews using advanced Natural Language Processing (NLP) and machine learning techniques. By processing a dataset of hotel reviews, we aim to uncover hidden sentiments and trends that can guide hotel management in enhancing customer satisfaction. The methodology encompasses several steps, including data cleaning, lemmatization, and stop word removal, to ensure the text data is appropriately prepared for analysis. We employ various vectorization techniques, such as TF-IDF and Count Vectorization, to convert textual data into numerical format suitable for machine learning models. Multiple classification algorithms, including Naive Bayes, Support Vector Machines, and Long Short-Term Memory (LSTM) networks, are utilized to predict sentiment polarity—classifying reviews as positive, negative, or neutral. Additionally, we leverage data visualization techniques, such as word clouds and bar plots, to illustrate the frequency of key terms and the overall sentiment distribution. These visualizations provide hotel managers with immediate insights into customer opinions and areas needing attention. The project ultimately demonstrates the potential of automated sentiment analysis in transforming unstructured text into actionable insights, allowing hotels to proactively address customer concerns and improve service quality. By integrating these advanced analytical methods, this study highlights the significant benefits that technology can offer in the realm of customer experience management within the hospitality sector.

ACKNOWLEDGEMENT

I extend my sincere gratitude to the Chancellor ,**Dr.T.R. PACHAMUTHU** and to Chairman **Dr. R. SHIVAKUMAR** of SRM Institute of Science and Technology, Ramapuram and Trichy campuses for providing me the opportunity to pursue the MCA degree at this University.

I express my sincere gratitude to **Dr. C. SUNDAR, Dean(S&H),** SRM IST, Ramapuram for his support and encouragement for the successful completion of the project.

I record my sincere thanks to **Dr. J. DHILIPAN, M.Sc., MBA., M.Phil., Ph.D., Vice Principal-Admin(S&H)** and **Head of the Department** of Computer Science and Applications, SRM IST, Ramapuram for his continuous support and keen interest to make this project a successful one.

I would like to convey my heartfelt gratitude to **Dr. D. KANCHANA, Program Coordinator** for her useful advice and suggestions which really helpful to me during the project completion.

I find no word to express profound gratitude to my **Dr. R. AGUSTHIYAR, MCA., M.Phil., Ph.D., Professor , Department of Computer Science and Applications (MCA),** SRM IST Ramapuram.

I thank the almighty who has made this possible. Finally, I thank my beloved family members and friends for their motivation, encouragement and cooperation in all aspect which led me to the completion of this project.

SELVI V

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 PROJECT INTRODUCTION

In recent years, the explosion of user-generated content has dramatically reshaped industries worldwide, and the hospitality sector is no exception. Hotels, resorts, and other accommodation providers now rely heavily on online reviews posted by guests to gauge their performance, customer satisfaction, and areas for improvement. These reviews, often written informally, contain a wealth of information, but extracting actionable insights from them is challenging due to the sheer volume of data. This is where automated sentiment analysis and Natural Language Processing (NLP) come into play, offering a solution for efficiently analyzing vast amounts of textual data.

This project focuses on analyzing hotel reviews using NLP techniques, transforming unstructured review data into structured, meaningful insights. The primary goal is to identify the sentiments expressed in these reviews—whether they reflect positive, negative, or neutral experiences. Sentiment analysis allows hotel management to make data-driven decisions, helping them to address common concerns, enhance guest experiences, and improve their services in a competitive market.

The data for this project consists of hotel reviews stored in an Excel file. These reviews serve as input to the NLP pipeline, which begins with preprocessing the text data. Preprocessing includes several critical steps: text cleaning to remove irrelevant characters or noise, lemmatization to reduce words to their base forms, and the removal of stopwords—common words like “the” and “is”—that do not contribute significantly to the sentiment analysis. Additionally, domain-specific stopwords, such as “hotel,” “stay,” and “room,” are removed to ensure the analysis focuses on the content most relevant to understanding customer experiences. Once the text data has been cleaned and processed, sentiment analysis is performed using several machine learning algorithms. These algorithms range from traditional models such as Naive Bayes and Support Vector Machines (SVM) to more advanced models like Long Short-Term Memory (LSTM) networks. Each model is trained to classify reviews into sentiment categories, allowing for a comprehensive understanding of guest feedback.

Beyond sentiment classification, the project includes various data visualization techniques to represent the results in a clear and actionable way. For instance, word clouds are used to visually depict the most frequently mentioned words across all reviews. This gives hotel managers a quick overview of the topics and themes most often discussed by guests. Furthermore, bar plots are generated to show the frequency of different sentiment categories, offering a snapshot of the overall customer satisfaction levels.

The insights gained from this analysis are invaluable for hotel managers. By understanding not only the overall sentiment but also the specific elements that guests appreciate or criticize, hotels can take targeted actions to improve their services. For example, if reviews frequently mention issues with room cleanliness or staff behavior, these can be immediately addressed to prevent future complaints. Similarly, positive feedback about amenities or customer service can help reinforce and promote those aspects of the hotel's offerings.

This project also explores methods for dealing with class imbalances in the dataset, such as oversampling and under sampling techniques. Reviews are often skewed towards positive or negative extremes, so these techniques ensure that the machine learning models are trained on a balanced set of examples, improving the accuracy of the sentiment predictions.

Overall, the project demonstrates the significant role NLP can play in transforming raw textual data into meaningful insights that can drive business improvements. By automating the sentiment analysis of hotel reviews, this project not only reduces the time and effort required for manual review but also provides hotels with powerful tools to better understand and respond to their guests' needs. This, in turn, leads to enhanced guest satisfaction, better reviews, and ultimately, a stronger competitive position in the hospitality market.

CHAPTER 2

2. Working Environment

2.1 Hardware Requirement

a. Server Hardware Requirement.

Processor: Intel Xeon Gold 5120 or equivalent with at least 8 cores.

RAM: Minimum 32 GB DDR4.

Storage: At least 1 TB SSD for fast read/write operations

Network: Gigabit Ethernet or higher for optimal data transfer rates.

b. Developer Workstation Requirements.

Processor: Intel i7 or AMD Ryzen 5 or equivalent

RAM: Minimum 16 GB DDR4.

Display: 24-inch Full HD (1920 x 1080) or higher.

Network: Stable high-speed internet connection (minimum 50 Mbps).

c. Continuous Integration Server

OS : Ubuntu/CentOS for servers, Windows/macOS/Linux for workstations.

IDE : Jupyter NoteBook.

Programming Language : Python,Mechine Learning.

CI/CD : Jenkins/GitLab CI/CircleCI, Git

2.2 SYSTEM SOFTWARE

1. Operating System

Windows 10/11 (recommended), or **macOS/Linux** (with .NET Core).

2. Python

- Python 3.7 or later for executing scripts and analysis
- Download: [Python Downloads](https://www.python.org/downloads/)

3. IDE

- Visual Studio Code or PyCharm for efficient Python development.

4. Natural Language Processing Libraries

- Install via pip:
- pip install numpy pandas matplotlib seaborn spacy nltk
- pip install wordcloud scikit-learn
- pip install keras tensorflow

5. Machine Learning Frameworks

- TensorFlow or Keras for building and training models for sentiment analysis.

- Install via pip:

- pip install tensorflow keras

6. Version Control

- Git for version control with repositories hosted on GitHub, Bitbucket, or GitLab.

7. Additional Libraries

- Install additional libraries for enhanced functionality:

- pip install imbalanced-learn for handling class imbalance.

- pip install textblob for text processing and sentiment analysis.

CHAPTER 3

3. SYSTEM ANALYSIS

System analysis is a crucial phase in any project development lifecycle. It involves evaluating the current system, identifying its limitations, and proposing solutions that meet organizational needs. This section outlines a detailed analysis of the current approach to hotel review management and the proposed enhancements through the implementation of Natural Language Processing (NLP) and machine learning techniques.

3.1 Feasibility Study

A feasibility study is an essential preliminary step in project planning, used to evaluate whether a proposed project is viable across three main dimensions: **technical feasibility**, **operational feasibility**, and **financial feasibility**. Let’s expand on each of these:

1. Technical Feasibility:

Technical feasibility involves evaluating whether the current technological environment and infrastructure are capable of supporting the project. This includes reviewing the hardware, software, and expertise available to determine if they can accommodate the demands of the new system.

- **Technological Requirements**: The project might need access to specialized tools such as **Natural Language Processing (NLP) libraries**, **machine learning frameworks**, and **data visualization tools**. Commonly used NLP libraries like **spaCy**, **NLTK**, and **Transformers** (from Hugging Face), as well as machine learning frameworks like **TensorFlow**, **PyTorch**, or **scikit-learn**, must be compatible with the current system. If these tools can integrate smoothly with the existing infrastructure, the project can proceed.
- **Hardware Capacity**: The technical feasibility study also assesses if the organization's existing infrastructure can handle the project's data processing, storage, and analytical needs. The system must have sufficient computing power (CPU, GPU, and memory), storage capacity, and networking capabilities to process large datasets, train machine learning models, and provide real-time insights.

- **Scalability**: It's important to ensure that the technological environment can scale as the project grows. If the project involves a large volume of data, it's crucial that the infrastructure can be expanded to handle future requirements without performance degradation.

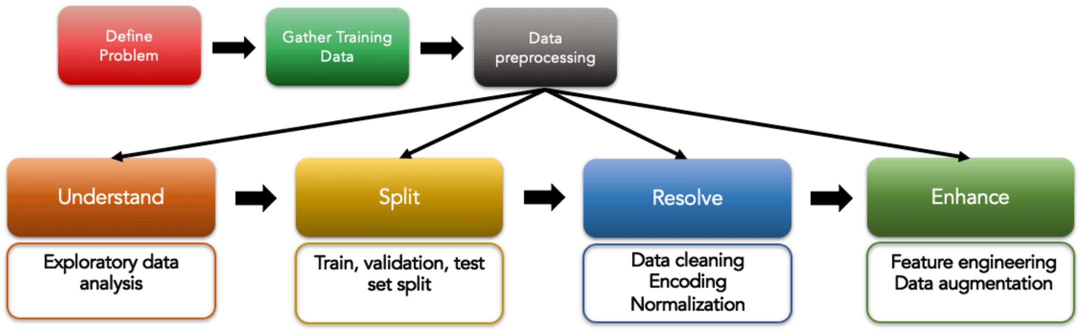


Fig 1

2. Operational Feasibility:

Operational feasibility focuses on the organization's ability to implement, manage, and support the new system in day-to-day operations. It assesses the readiness of the staff, workflows, and internal processes for the project.

- **Staff Expertise**: The team's familiarity with **NLP**, **machine learning**, and other advanced technologies needs to be evaluated. If the team lacks the necessary knowledge, **training programs** will be essential. Training could cover topics such as data processing, model development, and how to use the selected NLP and machine learning tools effectively.
- **Support and Maintenance**: Operational feasibility also involves evaluating whether the organization can provide ongoing support and maintenance for the system
-

3. Financial Feasibility:

Financial feasibility evaluates the cost-effectiveness of the project. This involves estimating the financial resources needed and comparing them to the expected benefits, to determine whether the investment is justified.

- **Cost Estimation**: Financial feasibility includes a detailed analysis of the costs involved in acquiring the necessary hardware and software, as well as costs for staff training. **Software costs** may involve purchasing licenses or subscriptions for machine learning tools or cloud services, while **hardware costs** could include upgrading servers or adding new computing resources for large-scale data processing.
- **Training Costs**: In addition to technical costs, there may be significant **training expenses** to ensure that staff can efficiently use the new system. This might include external training programs, online courses, or in-house workshops led by experts.
- **Expected Benefits and ROI**: The financial feasibility study will compare these costs against the expected benefits of the project. The key expected benefits are **improved customer insights** through better data analysis and **increased operational efficiencies**. For example, more accurate customer data from NLP and machine learning could lead to improved decision-making and personalized services, ultimately enhancing customer satisfaction and increasing revenue. The expected return on investment (ROI) helps determine whether the project's long-term financial gains will outweigh the upfront costs.

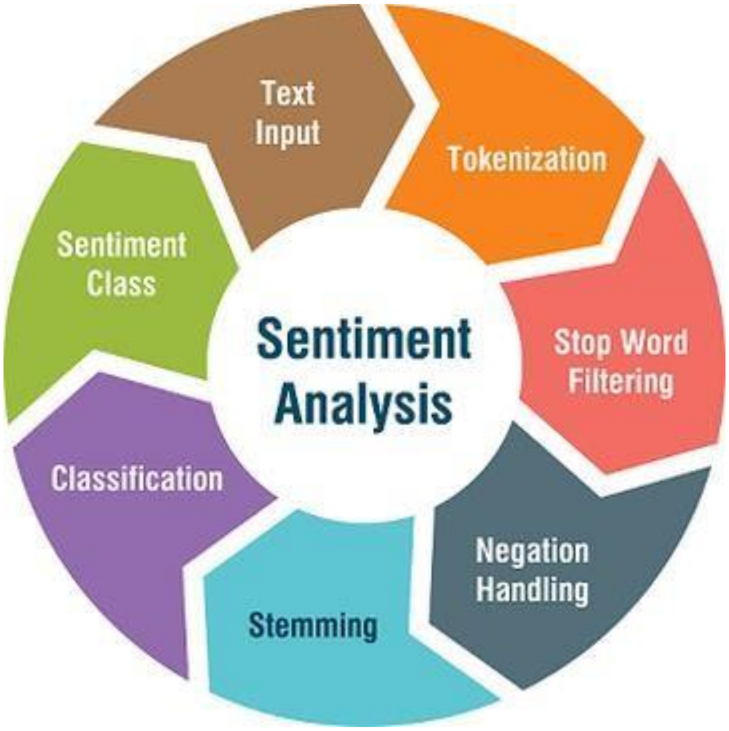


Fig 2

3.2 Existing System

The current system for analyzing hotel reviews is a **manual, labor-intensive process**, heavily reliant on human interpretation. This approach poses several limitations, particularly in terms of efficiency, scalability, and the ability to derive deep, meaningful insights from customer feedback. Let’s expand on the key characteristics of the existing system:

1. Manual Analysis:

The existing system requires hotel staff or managers to manually sift through individual reviews to understand customer sentiments and identify trends. This method involves reading, categorizing, and interpreting the textual feedback provided by customers. While it allows for personalized, human-driven insights, it presents several significant drawbacks:

- **Time-Consuming**: Reading through large volumes of reviews is extremely time-consuming, especially for hotels that receive hundreds or thousands of reviews. Managers must dedicate a substantial amount of time to this task, which detracts from their ability to focus on other operational priorities.
- **Limited Sample Size**: Due to the time constraints, manual analysis usually focuses on a small subset of reviews, often prioritizing the most recent or most visible feedback. As a result, valuable insights from older or less prominent reviews may be missed, leading to an incomplete understanding of customer sentiments.
- **Potential for Oversight**: The human factor introduces the risk of oversight. Important trends or patterns may go unnoticed, especially if they are subtle or scattered across numerous reviews. This could lead to missed opportunities for improvement in areas that are frequently mentioned but not immediately obvious.
- **Subjective Interpretation**: Human interpretation of reviews can be highly subjective. Different managers may interpret the same review differently based on their personal biases, experiences, or mood. This inconsistency makes it difficult to maintain a standardized approach to sentiment analysis and may lead to skewed conclusions.

2. Lack of Automation:

One of the key weaknesses of the existing system is the **absence of automation**, which hampers the hotel's ability to manage large volumes of reviews efficiently. Without automated processes, the system cannot keep up with the increasing volume of customer feedback, which is often scattered across various platforms (e.g., TripAdvisor, Google, Yelp, social media).

- **Inability to Scale**: As the number of reviews grows, the manual approach becomes less feasible. Large hotel chains or popular establishments with extensive customer feedback will struggle to keep up, leading to significant delays in analysis. Automation could help scale the review analysis process by quickly processing thousands of reviews, allowing for real-time insights and responses.
- **Limited Analysis Scope**: Manual methods restrict the scope of analysis to a small sample size, often focusing only on the most recent feedback or the most vocal customer complaints. This limited scope means that the hotel might overlook emerging trends, recurring issues, or positive feedback that could be leveraged for marketing purposes. Automation can handle much larger datasets, enabling comprehensive analysis of all available reviews, regardless of volume.
- **Inefficient Resource Use**: Without automation, hotel staff must dedicate significant time and effort to tasks that could otherwise be automated. By implementing automated review analysis systems, hotels could free up their staff to focus on more strategic and customer-facing activities, such as personalizing services based on insights or improving operational processes based on feedback.

3. No Use of Advanced Analytics:

The existing system does not leverage **modern analytical techniques**, such as **Natural Language Processing (NLP)** or **machine learning**, which are powerful tools for extracting meaningful insights from unstructured textual data like customer reviews.

Lack of Sentiment Analysis: Advanced sentiment analysis, powered by NLP, can automatically detect and quantify the emotional tone of customer reviews.

- By identifying positive, negative, or neutral sentiments, hotels can gain a clearer understanding of how their services are perceived without needing to manually interpret each review.
- **Inability to Identify Key Trends**: Without machine learning and NLP, the current system lacks the ability to identify recurring themes, keywords, or patterns across large datasets. Advanced analytics can detect hidden trends, such as frequent complaints about specific services (e.g., cleanliness, staff behavior, food quality) or praise for certain amenities, enabling data-driven decisions for improving guest experiences.
- **Missed Opportunities for Predictive Analysis**: Machine learning can go beyond simple sentiment analysis to predict future customer behavior or preferences based on historical review data. For example, it could identify which aspects of the hotel experience are most likely to lead to positive reviews and repeat bookings
- **No Data Visualization**: In addition, modern analytics tools can transform complex review data into visual insights through data visualization techniques. This allows hotel managers to quickly grasp trends and patterns through dashboards or reports, rather than reading through raw review text. The current system does not provide these capabilities, making it harder for decision-makers to interpret the data efficiently.

3.3 Drawbacks of Existing System

The current system for analyzing hotel reviews, which is primarily manual, has several critical limitations. These drawbacks hinder the effectiveness of the system in delivering timely, accurate, and comprehensive insights into customer feedback. The following are the major drawbacks:

1. Time-Consuming Processes:

One of the most significant drawbacks of the existing system is that it is highly **time-consuming**. Since hotel managers or staff have to manually read, process, and interpret each review, the process is slow and labor-intensive. This has multiple implications:

Delays in Decision-Making: Manual analysis requires considerable time to gather and interpret feedback, resulting in delays in responding to customer concerns or addressing

- operational inefficiencies. If reviews point to urgent issues (e.g., cleanliness problems or rude staff), delays in identifying and responding to these concerns can damage the hotel's reputation and customer satisfaction.
- **Lower Productivity**: Hotel staff must allocate a significant portion of their time to tasks that could otherwise be automated. This detracts from their ability to focus on other responsibilities such as improving customer experience, developing strategic initiatives, or even responding to reviews in real time. Automating the review analysis process could improve overall operational efficiency.
- **Risk of Overlooking Important Feedback**: Due to the labor-intensive nature of manual review analysis, there is a risk of overlooking important feedback. Staff may rush through the process or unintentionally neglect certain reviews, especially when facing a large volume of comments. As a result, critical customer insights might be missed.

2. Bias in Interpretation:

Another major drawback is the **subjective nature of manual interpretation**. Human analysis is inherently prone to bias, and different individuals may interpret the same review differently based on their perspectives, experiences, or emotions at the time of reading. This introduces several issues:

- **Inconsistent Sentiment Analysis**: Since hotel staff members interpret reviews differently, there may be inconsistencies in how feedback is categorized and understood. For example, one staff member might consider a certain comment as "negative" while another might interpret it as "neutral." These inconsistencies can lead to flawed analysis and decision-making.
- **Personal Bias**: Personal biases, whether conscious or unconscious, can affect how reviews are interpreted. Staff members might focus more on reviews that align with their pre-existing beliefs or overlook feedback that challenges their perspectives. This selective interpretation can result in a skewed understanding of customer sentiment.

  Emotional Influence: The emotional state of the reviewer and the person analyzing the review can also impact interpretation. For example, a negative review written in a sarcastic tone may be misinterpreted as humorous or not serious enough, even if it highlights a genuine problem

3. Inability to Scale:

The **manual approach** to review analysis becomes unsustainable as the volume of reviews grows, particularly for larger hotels or chains that receive hundreds or thousands of reviews across multiple platforms. This lack of scalability poses several significant challenges:

- **Limited Scope of Analysis**: Due to time and resource constraints, staff may only be able to analyze a small subset of reviews, typically focusing on the most recent or most prominent ones. This limited sample size may lead to an incomplete understanding of customer sentiment, as reviews that reflect key trends or recurring issues may go unnoticed. Without the ability to process large datasets, important patterns and insights are likely to be missed.
- **Missed Opportunities for Comprehensive Insights**: As review volume increases, the manual system fails to capture a holistic view of customer feedback. Automating the process would allow for the analysis of large datasets, which could uncover valuable insights such as frequent customer complaints, praise for specific services, or emerging trends in guest preferences. This limitation prevents hotel management from gaining comprehensive insights that could guide operational improvements or enhance the guest experience.
- **Difficulty in Handling Multichannel Feedback**: Modern hotels receive reviews across multiple platforms, including websites like TripAdvisor, Google Reviews, social media, and booking platforms like Booking.com or Expedia. The existing manual system struggles to gather and analyze feedback from all these sources, further limiting the analysis to a small fraction of available data. Automating the process would enable seamless integration of multichannel feedback into a unified analysis framework.

4. Lack of Real-Time Insights:

- Due to the slow, manual nature of the process, the existing system **cannot provide real-time insights** into customer feedback. By the time reviews are read and analyzed, the issues highlighted in the reviews may already have affected multiple customers. This delay prevents hotel management from responding to problems in a timely manner, which could result in further dissatisfaction and a potential decrease in overall guest satisfaction scores.

- Real-time insights are crucial in today's competitive market, where quick responses and proactive actions can significantly improve customer experiences. Without the ability to generate real-time reports or alerts, hotels miss the opportunity to address critical concerns immediately.

5. Inability to Perform Trend Analysis:

The existing system lacks the ability to effectively track and analyze trends over time. Without advanced analytical tools like **NLP** (Natural Language Processing) or **machine learning**, the hotel cannot identify recurring themes, emerging trends, or long-term patterns in customer reviews.

- **No Identification of Common Complaints**: Manually reviewing each comment does not allow the system to group similar reviews together or identify frequent complaints. For instance, if multiple guests consistently mention issues with cleanliness or noise, the current system is unlikely to pick up on these trends until they become widespread.
- **No Predictive Analysis**: Trend analysis and predictive insights could help hotels anticipate customer needs or preferences based on historical review data. For example, during certain seasons, guests might frequently mention specific issues like heating or air conditioning. Without the ability to analyze reviews in bulk and over time, the current system cannot make these connections, resulting in missed opportunities for proactive service adjustments.
- The drawbacks of the existing system—**time-consuming manual processes**, **bias in interpretation**, **inability to scale**, and the **lack of real-time insights and trend analysis**—make it ineffective in today's fast-paced and data-driven hospitality industry. These limitations prevent hotel management from fully understanding customer feedback and taking timely, informed actions to improve guest experiences. An automated review analysis system that leverages advanced technologies like NLP and machine learning would greatly enhance the efficiency, accuracy, and depth of the analysis, allowing hotels to make data-driven decisions and respond to customer needs more effectively.

3.4 Proposed System

The proposed system seeks to overcome the limitations of the current manual approach to hotel review analysis by integrating **Natural Language Processing (NLP)** and

machine learning techniques. These advanced technologies will enable more efficient, scalable, and accurate analysis of customer feedback, transforming how hotels interpret and respond to reviews. Let's expand on the key features of this system:

1. Automation of Review Analysis:

One of the primary improvements in the proposed system is the **automation of review analysis**, which eliminates the need for manual reading and interpretation of reviews. This automation offers several significant advantages:

- **Efficient Processing of Large Datasets**: The system can quickly process vast amounts of review data from various platforms (such as TripAdvisor, Google Reviews, and Booking.com), regardless of the volume. This scalability ensures that all reviews, not just a limited sample, are analyzed, providing a more comprehensive understanding of customer feedback.
- **Real-Time Analysis**: Automation enables real-time analysis of reviews as they are posted, allowing hotel managers to receive up-to-date insights into customer sentiment. This means that issues can be identified and addressed immediately, improving customer satisfaction and preventing potential problems from escalating.
- **Reduction of Human Error**: By removing manual intervention, the system eliminates the risk of human error, such as misinterpretation of reviews or overlooking key feedback. This ensures that all reviews are treated equally, and valuable insights are not missed due to time constraints or oversight.
- **Improved Resource Utilization**: Automating the review analysis process frees up hotel staff to focus on higher-value tasks, such as improving customer experience, refining services based on insights, or implementing strategic initiatives. Hotel managers can prioritize actions based on data-driven insights rather than spending hours manually sorting through reviews.

2. NLP Techniques for Sentiment Extraction:

The proposed system leverages **Natural Language Processing (NLP)** techniques to extract and interpret sentiments from textual reviews. NLP allows for a much deeper and more nuanced understanding of customer opinions, beyond just simple positive or negative categorizations.

- **Sentiment Analysis with Contextual Understanding**: Traditional methods of sentiment analysis often classify feedback into binary categories like "positive" or "negative." However, NLP techniques enable the system to capture **nuanced**

- **sentiments**, including **neutral** sentiments or **mixed opinions** where customers mention both positive and negative aspects in a single review. This allows hotel managers to gain a more sophisticated understanding of how customers feel about specific services or experiences.
- **Identification of Key Themes and Topics**: NLP can automatically identify common themes, topics, or keywords in customer reviews, helping managers to pinpoint areas that are frequently mentioned (e.g., service quality, cleanliness, food, or room comfort). For example, if multiple reviews mention problems with Wi-Fi, the system can detect this pattern and flag it as a recurring issue, even if the comments are spread across different platforms and phrased differently. This capability helps managers prioritize their focus on the most critical issues.
- **Handling of Complex Language Structures**: NLP is equipped to handle various language complexities, including **sarcasm**, **idiomatic expressions**, and **contextual subtleties** that would otherwise be challenging to interpret manually. By understanding these complexities, the system can provide a more accurate analysis of customer sentiments. For instance, it can detect sarcasm in a review like "The room was amazing, if you love sleeping next to a construction site" and correctly classify it as negative.
- **Multilingual Support**: Many hotels cater to international guests who leave reviews in different languages. NLP techniques, combined with language processing tools, enable the system to analyze reviews written in multiple languages, further broadening the scope of analysis. This allows hotels to understand customer sentiment globally without requiring translations to be done manually.

3. Machine Learning Models for Sentiment Classification:

The proposed system incorporates various **machine learning algorithms** to classify sentiments accurately and reliably. Machine learning models can learn from large datasets, continuously improving the accuracy and efficiency of sentiment classification over time.

- **Supervised and Unsupervised Learning Models**: The system can use **supervised learning** techniques, where the model is trained on a labeled dataset (where reviews are already categorized as positive, negative, or neutral), enabling it to accurately classify new reviews.

- **Scalability and Flexibility**: Machine learning models are highly scalable, meaning they can handle increasing volumes of data without significant degradation in performance. This makes the system suitable for both small boutique hotels and large international chains with massive amounts of customer feedback. As new reviews are continuously added, the machine learning algorithms can update and adjust their analysis, ensuring that the system stays relevant and accurate.
- **Improved Accuracy Over Time**: One of the key benefits of machine learning models is that they improve with use. As more data is fed into the system, the models refine their predictions and become better at classifying sentiments accurately. For instance, the system might initially struggle with ambiguous or contradictory reviews but will learn to handle these complexities over time, increasing the precision of its sentiment analysis.
    - **Predictive Analytics**: In addition to classifying sentiments, machine learning models can be used to make **predictions** about future customer behavior. By analyzing historical review data, the system can predict which services or experiences are most likely to lead to positive reviews or customer satisfaction.

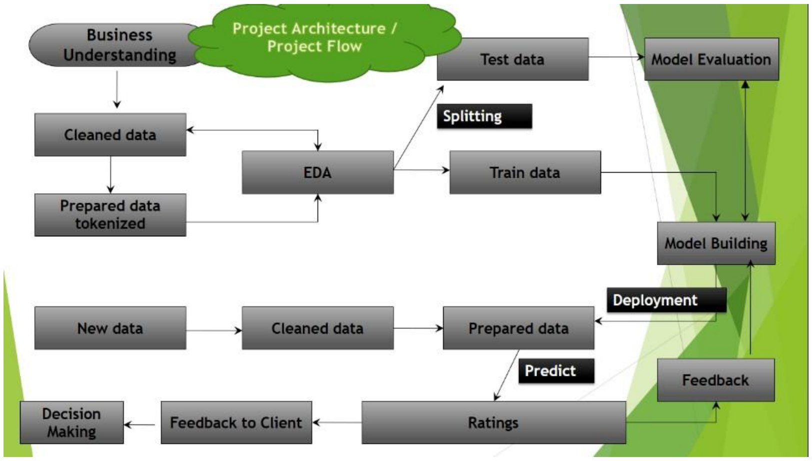


Fig 3

4. Enhanced Speed and Accuracy:

The integration of NLP and machine learning not only significantly **speeds up the review analysis process** but also **improves the accuracy** of the insights derived from the data. This combination of speed and accuracy provides hotel managers with actionable intelligence, empowering them to make data-driven decisions.

- **Speed of Analysis**: With automation, reviews can be processed within seconds or minutes of being posted, providing hotel managers with timely insights. This allows for quick identification of issues and more rapid responses to customer concerns. For instance, if a recurring complaint is identified, management can take immediate action to resolve the issue before it affects more customers.
- **Data-Driven Decision-Making**: Accurate sentiment analysis enables managers to make informed decisions based on actual customer feedback rather than relying on intuition or manual interpretations. This helps to ensure that operational improvements are aligned with customer needs and preferences. For example, if the system detects that cleanliness is frequently praised, but staff interactions are often criticized, hotel management can prioritize staff training programs to improve customer service.

3.5 Benefits of Proposed System

The proposed system offers a wide range of significant benefits that enhance the hotel's ability to efficiently analyze customer reviews and gain valuable insights. These advantages include improved efficiency, the ability to make data-driven decisions, and scalability, making it a transformative tool for improving service quality and guest satisfaction. Let's explore these benefits in more detail:

1. Enhanced Efficiency:

One of the primary advantages of the proposed system is the **dramatic increase in efficiency** achieved through the automation of the review analysis process. This automation delivers numerous benefits that address the time-consuming nature of the current manual system:

- **Reduction in Time**: The system can process large volumes of reviews in a fraction of the time it takes for human staff to manually read and analyze them. Instead of hours or even days spent manually sorting through reviews, the system performs the analysis in real time or near real time. This enables hotel managers to receive insights as soon as reviews are posted, leading to quicker responses to customer concerns.
- **Real-Time Insights**: With automated analysis, reviews can be evaluated as they are posted, providing **real-time insights** into customer sentiments. This gives hotel management the ability to respond to both positive and negative feedback much faster than with manual processes.

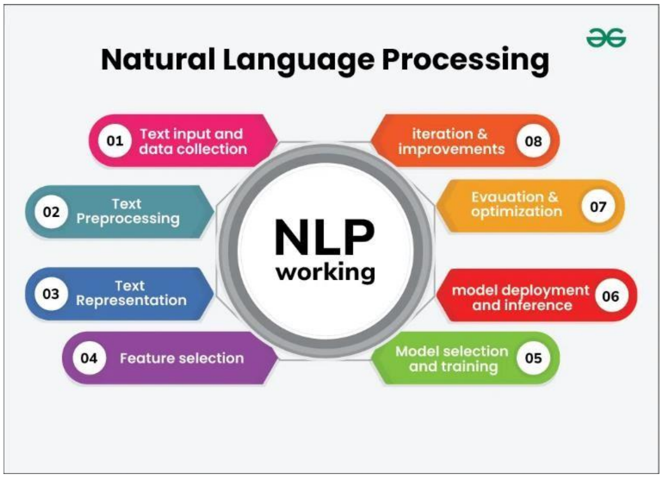

Natural Language Processing
NLP working
01 Text input and data collection
02 Text Preprocessing
03 Text Representation
04 Feature selection
05 Model selection and training
06 model deployment and inference
07 Evauation & optimization
08 iteration & improvements

Fig 4

- **Increased Staff Productivity**: By automating the review analysis process, hotel staff no longer need to dedicate their time to manually reviewing and interpreting feedback. This frees them up to focus on more value-added tasks, such as directly improving the guest experience, managing daily operations, or developing strategic initiatives.
- **2. Data-Driven Decision-Making:**

The system’s ability to accurately classify sentiments and extract insights from large datasets provides hotel managers with the tools they need to make **data-driven decisions**. This marks a significant improvement over the existing system, where decisions may be based on subjective or incomplete interpretations of customer feedback. The following benefits arise from this enhanced capability:

- **Accurate Sentiment Classification**: The use of **Natural Language Processing (NLP)** and **machine learning** ensures that customer sentiments are classified accurately, even when reviews contain nuanced, complex, or contradictory opinions. For example, a guest might leave a review that contains both praise for the hotel’s location and criticism of the cleanliness. The system can accurately classify this review as containing both positive and negative sentiments, providing a more precise picture of the customer’s experience.
- **Actionable Insights**: Rather than relying on intuition or partial information, hotel managers can make decisions based on concrete data and trends. The system can automatically highlight key insights, such as frequent complaints or recurring praise, helping managers focus their efforts on areas that will have the greatest impact on guest satisfaction

- **Objective Analysis of Customer Feedback**: Data-driven decision-making eliminates the risk of **subjective bias** inherent in manual review analysis. Because the system relies on objective data, all reviews are treated consistently, ensuring that feedback is not misinterpreted or overlooked. This allows for a fair and balanced assessment of customer experiences, providing a more accurate basis for decision-making.

3. Scalability:

Another key benefit of the proposed system is its **scalability**, which makes it suitable for hotels of all sizes, from small boutique establishments to large chains with thousands of reviews across multiple platforms. Scalability ensures that the system can adapt to growing volumes of customer feedback while maintaining accuracy and efficiency.

- **Handling Large Datasets**: The system is designed to handle vast amounts of data, enabling it to process reviews from multiple platforms (such as TripAdvisor, Google Reviews, Booking.com, and social media) without any loss of performance. This is particularly important for larger hotels or chains that receive thousands of reviews per month. Instead of manually analyzing a small sample of these reviews, the system can process **all reviews**, providing a complete picture of customer feedback.
- **Comprehensive Understanding of Guest Satisfaction**: By analyzing large datasets, the system offers a more **comprehensive understanding of guest satisfaction**. It can detect trends, patterns, and recurring themes that would be difficult to identify manually. For example, the system might uncover that guests frequently mention the excellent quality of the food but also express dissatisfaction with wait times. By analyzing all available data, hotel managers can prioritize areas for improvement or investment based on actual customer preferences and concerns.
- **Adaptability to Multichannel Feedback**: Modern hotels receive feedback from a variety of platforms, including online reviews, social media posts, and customer surveys. The system is capable of aggregating feedback from multiple sources, ensuring that all guest opinions are taken into account, regardless of where they are posted. This multichannel approach ensures that no important feedback is missed, further enhancing the hotel’s ability to respond to guest needs.
- **Support for Multilingual Reviews**: With the increasing globalization of the hospitality industry, many hotels cater to international guests who leave reviews in multiple

languages. The proposed system’s NLP capabilities allow it to process reviews in different languages, ensuring that feedback from international guests is fully considered in the analysis. This feature broadens the scope of the system, making it useful for hotels with a diverse, global clientele.

4. Proactive Response to Customer Needs:

The system’s ability to generate **real-time insights** and process large amounts of data allows hotel managers to be more **proactive** in addressing customer needs and concerns. This proactivity offers several advantages:

- **Faster Issue Resolution**: The real-time nature of the system means that hotel staff can identify and respond to problems almost as soon as they arise. If multiple guests report an issue with room service or cleanliness, management can take immediate steps to rectify the problem, improving the overall guest experience and preventing further negative reviews.
- **Improving Guest Satisfaction and Loyalty**: By responding quickly to customer feedback, hotels can demonstrate that they are actively listening to their guests and are committed to improving their services. This responsiveness can enhance guest satisfaction, leading to increased loyalty and positive word-of-mouth. In today’s competitive hospitality market, the ability to quickly resolve issues and address customer needs is a key differentiator.
- **Enhancing Service Quality**: The system’s insights allow hotels to continuously improve the quality of their services. For example, if the system detects frequent praise for certain aspects of the guest experience, such as friendly staff or comfortable rooms, hotel managers can focus on reinforcing these strengths while addressing areas of concern. This iterative process of improvement helps ensure that service quality is consistently high.

3.6 Scope of the Project

The **project scope** defines the key areas of focus for developing a system to analyze hotel reviews, aiming to deliver a comprehensive solution for understanding customer sentiments. The scope outlines several core components that are essential to the functioning of the system and its ability to provide actionable insights for hotel management. Let’s expand on these elements in more detail:

1. Analysis of Hotel Reviews:

The central aspect of the project is the **processing and analysis of textual hotel reviews**. The system will automate the extraction of sentiments from customer feedback, enabling a much more efficient and scalable approach compared to traditional manual review analysis.

- **Automated Text Processing**: The system will leverage **Natural Language Processing (NLP)** techniques to analyze customer reviews, parsing the text to understand the emotions, opinions, and experiences expressed by guests. This involves processing large volumes of data, including identifying relevant keywords, extracting meaningful patterns, and interpreting the context of the reviews.
- **Sentiment Extraction**: The primary goal of this analysis is to extract sentiment information from the reviews. This involves determining whether a review expresses a positive, negative, or neutral sentiment towards various aspects of the hotel experience, such as room quality, service, cleanliness, or amenities. For instance, a review stating "The staff was friendly, but the room was dirty" would be recognized as containing both positive and negative sentiments.
- **Multilingual Support**: Since many hotels cater to international guests, the analysis will support reviews written in multiple languages. This ensures that feedback from diverse customer segments is fully analyzed, providing hotels with insights from a global audience.

2. Sentiment Classification:

A key feature of the project is the **classification of sentiments** within the reviews. Sentiment classification will allow hotel management to quickly and easily interpret the overall tone of customer feedback, providing clear and actionable insights.

- **Categorization of Sentiments**: The system will categorize reviews into sentiment categories such as **positive**, **negative**, and **neutral**. By doing so, hotel managers can gauge the overall customer satisfaction levels based on aggregated sentiment scores. For instance, reviews praising the service but criticizing the food will be classified accordingly, giving management a clear understanding of what areas require attention.
- **Granular Sentiment Analysis**: Beyond overall sentiment classification, the system will offer more granular analysis by identifying specific aspects or themes mentioned in reviews (e.g., "food quality," "cleanliness," or "staff behavior").

- **Improvement over Manual Sentiment Classification**: Traditional manual classification of reviews is time-consuming and prone to human bias. The automated system eliminates these inefficiencies by providing an objective and consistent method for classifying sentiments, ensuring a more reliable analysis of customer feedback.

3. Data Visualization:

One of the major components of the project is the use of **data visualization** tools to present the results of the sentiment analysis in a clear and easily interpretable manner. Data visualization enhances the usability of the system by allowing hotel managers to quickly understand complex data insights.

- **Word Clouds**: The system will generate **word clouds**, a visual representation of the most frequently mentioned words in customer reviews. The size of each word in the cloud represents its frequency or importance. This feature allows managers to quickly see what topics are most commonly discussed by guests. For instance, if "cleanliness" appears as a large word in the cloud, it indicates that cleanliness is a recurring theme in customer feedback, either positively or negatively.
- **Bar Graphs and Charts**: In addition to word clouds, the system will utilize **bar graphs**, pie charts, and other visual formats to display sentiment distribution, key themes, and trends over time. These charts will help managers quickly identify patterns, such as the percentage of reviews that are positive versus negative or how sentiment regarding specific services (e.g., room service or check-in experience) evolves over time. For example, a bar graph showing an increase in negative reviews related to dining services over the past few months can signal a need for intervention.
- **Interactive Dashboards**: The system may also feature **interactive dashboards**, allowing hotel managers to filter reviews by date, sentiment, or service area, offering a more personalized exploration of the data. These dashboards can be customized to focus on specific issues or areas of interest, providing detailed insights at a glance.

4. Potential for Future Extensions:

While the project’s initial scope focuses on hotel reviews, there is substantial room for future **extensions** that would broaden the system’s capabilities. These future enhancements aim to incorporate a wider range of data sources and provide even richer insights into customer behavior and preferences.

- **Incorporation of Social Media Platforms**: The system could be expanded to include the analysis of customer feedback from **social media platforms** such as Facebook, Twitter, and Instagram. Social media often serves as an informal channel for guests to share their experiences, whether positive or negative. By analyzing these posts, the system could provide real-time insights into guest sentiment, allowing hotel managers to monitor their online reputation and respond quickly to emerging issues.

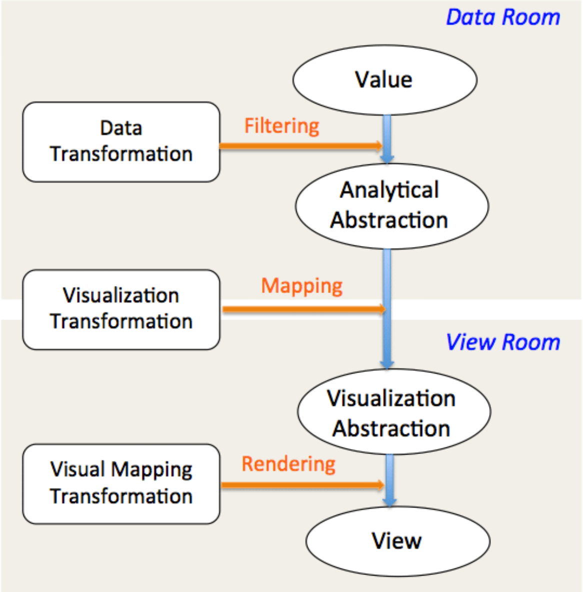


Fig 5

- **Integration with Customer Surveys**: The system could also be integrated with data from **customer surveys**, allowing hotels to compare survey responses with publicly posted reviews. This integration would provide a more comprehensive view of customer feedback by combining structured survey data with unstructured text from reviews and social media.
- **Predictive Analytics**: As the system collects more data over time, it could incorporate **predictive analytics** to forecast future trends in guest satisfaction or predict the impact of certain changes (such as pricing adjustments or service upgrades) on guest sentiment.
- **Sentiment Analysis for Competitors**: Another potential extension could involve analyzing reviews of **competitor hotels**, providing valuable market insights. By understanding how guests feel about competing properties, hotels can benchmark their performance and identify areas where they have a competitive advantage or need to improve.

CHAPTER 4

4. System Design

1. **Data Ingestion**:
   - **Source**: Excel file (e.g., hotel_reviews.xlsx) containing hotel review data.
   - **Data Handling**: Read data with pandas, handle missing data, blank entries, and categorical conversion.
2. **Data Preprocessing**:
   - **Text Cleaning**: Removing stopwords, punctuation, and non-alphabetic characters.
   - **Tokenization and Vectorization**: Using TfidfVectorizer for transforming text data into numerical features for modeling.
3. **Natural Language Processing (NLP)**:
   - **Text Features**: Generate word-level and sentence-level features using spaCy (POS tagging, named entity recognition) and nltk (sentiment analysis, word clouds).
   - **Sentiment Analysis**: Use tools like TextBlob and VADER to extract sentiment features.
4. **Machine Learning Models**:
   - **Classification Algorithms**: Implement several models like Logistic Regression, Decision Trees, and Random Forests to predict review ratings.
   - **Resampling Techniques**: Use RandomOverSampler or RandomUnderSampler from imblearn to handle imbalanced data.
5. **Evaluation and Comparison**:
   - **Cross-Validation**: Perform k-fold cross-validation to evaluate models.
   - **Performance Metrics**: Track accuracy, precision, recall, F1 scores, and confusion matrices for comparison.
   - **Model Comparison**: Store evaluation metrics in a dataframe for model comparison.
6. **Visualization**:
   - **Data Visualization**: Use matplotlib and seaborn for visualizing the distribution of review ratings, word clouds, and model performance.

CHAPTER 5

5. PROJECT DESCRIPTION

5.1 OBJECTIVE

- Develop a robust system to **classify hotel reviews** based on sentiment (positive, negative, neutral).
- **Leverage NLP** and **machine learning** to automate review analysis, reducing manual effort.
- **Visualize sentiment** results using charts and word clouds to provide comprehensive insights for hotel managers.
- **Facilitate decision-making** by identifying service strengths and areas for improvement.

5.2 Module Description

The project is divided into three main modules:

1. **Data Preprocessing Module**:
   - **Text Normalization**: Standardizing text format (e.g., lowercasing).
   - **Tokenization**: Splitting text into words or phrases.
   - **Stopword Removal**: Removing common, irrelevant words.
   - **Lemmatization**: Reducing words to their base form.
2. **Sentiment Analysis Module**:
   - **Feature Extraction**: Using techniques like TF-IDF to numerically represent text.
   - **Model Training**: Training classifiers (Naive Bayes, SVM, LSTM) on labeled data.
   - **Sentiment Classification**: Applying models to classify sentiment in new reviews.
3. **Visualization Module**:
   - **Data Visualization**: Creating **bar graphs, pie charts, word clouds** to show sentiment and frequently mentioned words.
   - **Reporting**: Summarizing findings in reports for hotel management.

5.3 Implementation

- **Coding the Modules**: Using Python libraries (Pandas, NumPy, NLTK, spaCy, Scikit-learn) for modular, testable code.

- **Testing the Modules**:
  - **Unit Testing**: Ensuring functions and classes are correct.
  - **Integration Testing**: Confirming smooth module interaction.
  - **Performance Testing**: Evaluating the system's speed with large datasets.
- **Integration into a Cohesive Application**: Unifying modules into one application to ensure seamless functionality across preprocessing, analysis, and visualization.

5.4 Maintenance

- **Model Updates**: Retrain models with fresh data for accuracy and relevance.
- **Algorithm Refinement**: Continuously improve models via **hyperparameter tuning**, new algorithms, or ensemble methods.
- **Software Compatibility**: Ensure compatibility with updated Python libraries and dependencies.
- **User Feedback Integration**: Collect feedback from hotel managers for improvements and new features.

Conclusion

The structured development approach focuses on modular design, implementation, and maintenance to create an efficient sentiment analysis system. Continuous updates and improvements will ensure that the system remains a valuable tool for analyzing customer feedback and aiding decision-making in the hospitality industry.

CHAPTER 6

6.System Testing

System testing is a critical phase in the development of the hotel reviews analysis system, focusing on verifying that the system functions as intended and meets the specified requirements. It evaluates the system’s performance, reliability, and overall effectiveness in processing hotel reviews, extracting sentiments, and visualizing data for end users.

6.1 Testing Definition

Testing, in the context of the hotel reviews system, involves assessing the entire application to ensure that it accurately analyzes hotel reviews and provides meaningful insights. This includes evaluating the system's ability to preprocess review data, perform sentiment analysis, and generate visualizations. The objective is to validate that the software meets its intended functionality and delivers accurate results to stakeholders.

6.2 Testing Objective

The primary objective of system testing for the hotel reviews system is to ensure that the application meets the defined requirements and performs accurately across a range of scenarios. This includes validating that:

- The system correctly preprocesses raw hotel reviews by removing noise and normalizing the text.
- Sentiment analysis accurately classifies reviews as positive, negative, or neutral.
- Visualization components (like word clouds and sentiment graphs) accurately reflect the analyzed data.
- The system can handle a large volume of reviews efficiently without performance degradation.

- All functionalities are user-friendly, enabling hotel managers to easily interpret results and make data-driven decisions.

6.3 Types of Testing

For the hotel reviews system, several types of testing are essential:

1. **Unit Testing**: Each module (e.g., data preprocessing, sentiment analysis, and visualization) will be tested individually. Developers will create unit tests to ensure that specific functions, such as text normalization and sentiment scoring, work correctly.

2. **Integration Testing**: This phase focuses on verifying the interaction between different modules. For instance, it will test whether the cleaned reviews from the preprocessing module are correctly passed to the sentiment analysis module and that the output is successfully integrated with the visualization components.

3. **System Testing**: The complete system will be tested as a whole. This includes running end-to-end scenarios that mimic real-world usage, such as importing hotel reviews, processing them, generating sentiment scores, and displaying results through visualizations. Performance testing will also be conducted to evaluate the system's ability to handle large datasets efficiently.

4. **User Acceptance Testing (UAT):** This final phase involves hotel managers or end users testing the system to ensure it meets their expectations and is user-friendly. Feedback gathered during UAT will be crucial for making final adjustments before deployment.

6.4 Test Cases

Developing detailed test cases is essential for assessing each module's performance in the hotel reviews system.

Each test case will cover key functionalities, focusing on expected outcomes. Examples of test cases may include:

Test Case ID: TC001

Description: Test the data preprocessing module.

Preconditions: A set of raw hotel reviews is available for processing.

Test Steps: Input raw reviews into the preprocessing module and execute.

Expected Results: The output should be cleaned reviews with punctuation removed, stopwords eliminated, and lemmatization applied.

Actual Results: To be filled after execution.

Status: Pass/Fail.

Test Case ID: TC002

Description: Test sentiment analysis accuracy.

Preconditions: Cleaned reviews are available for analysis.

Test Steps: Input cleaned reviews into the sentiment analysis module and execute.

Expected Results: Each review should be assigned a sentiment score (positive, negative, or neutral) based on predefined criteria.

Actual Results: To be filled after execution.

Status: Pass/Fail.

Test Case ID: TC003

Description: Test the visualization output.

Preconditions: Sentiment scores are generated.

Test Steps: Execute the visualization module to generate word clouds and sentiment graphs.

Expected Results: Visualizations should accurately represent the sentiment distribution and most common words in the reviews.

Actual Results: To be filled after execution.

Status: Pass/Fail.

These test cases will ensure that all critical functionalities of the hotel reviews system are thoroughly evaluated and any issues are addressed before deployment.

CHAPTER 7

7.Conclusion

7.1 Summary

This project seems to focus on analyzing hotel reviews to extract meaningful insights using various data science and NLP techniques. It involves several key steps:

1. **Data Importing and Preprocessing**:
   - The project starts with loading the hotel reviews dataset from an Excel file (hotel_reviews.xlsx).
   - Checks for missing data (isna) and blanks in the reviews.
   - Performs some categorical conversions (e.g., on the 'Rating' column).
2. **Natural Language Processing (NLP):**
   - Use of libraries like SpaCy, NLTK, and TextBlob to clean and process textual data.
   - Conducting sentiment analysis using Vader SentimentIntensityAnalyzer and TextBlob.
   - Tokenization and vectorization using TfidfVectorizer and CountVectorizer.
3. **Visualization:**
   - Generates visualizations using libraries like Matplotlib and Seaborn.
   - Part-of-Speech (POS) visualization using SpaCy's displacy.render.
4. **Modeling:**
   - Logistic regression for classification tasks, possibly predicting ratings or sentiments.
   - Addressing imbalanced data using oversampling and undersampling techniques (RandomOverSampler, RandomUnderSampler).
   - Performance evaluation with accuracy scores, confusion matrices, and classification reports.

7.2 Future Enhancements:

1. Advanced NLP Techniques:

- Implement deep learning models such as LSTM or Transformer models for more complex sentiment analysis and classification.

- Use pre-trained language models like BERT for improved accuracy in text **understanding.**

2. Feature Engineering:

- Incorporate additional features like review length, time of review, or user profile data to enhance model accuracy.

3. Topic Modeling:

- Use Latent Dirichlet Allocation (LDA) to identify topics in reviews and gain deeper insights into customer sentiments.

4.Interactive Visualizations:

- Implement interactive visualizations (using Plotly or Dash) to explore data trends more dynamically.

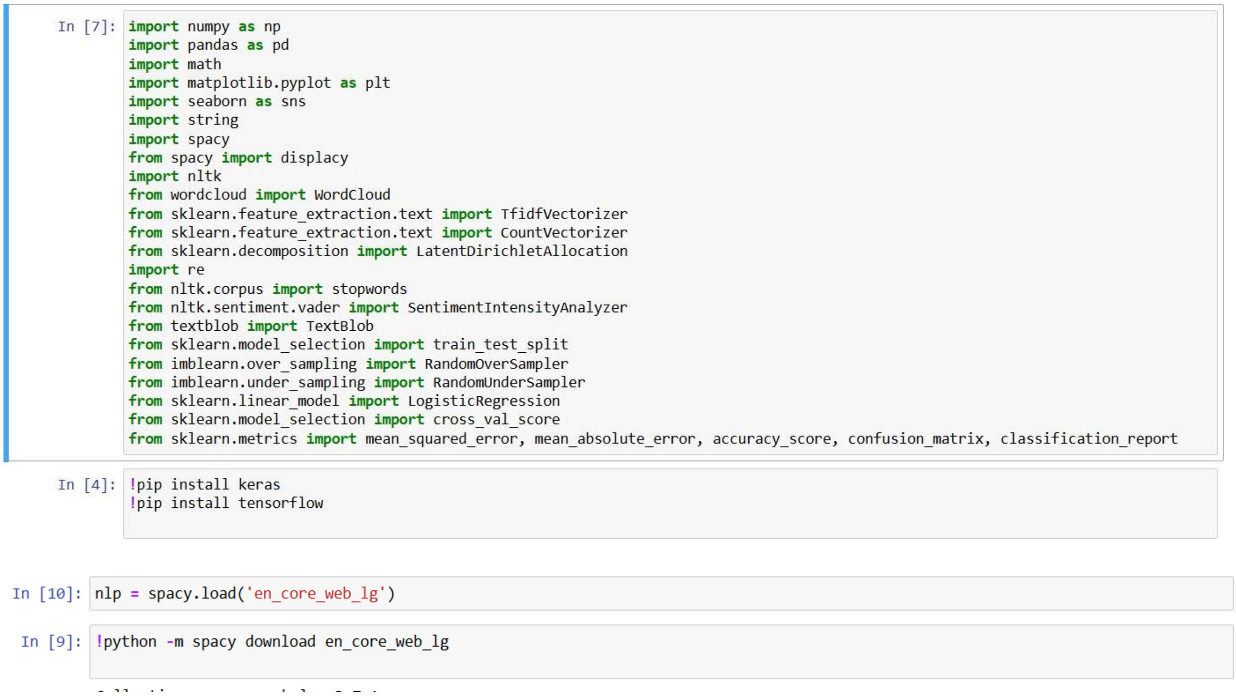
5. Model Optimization:

- Experiment with other machine learning algorithms such as Random Forest, XGBoost, or SVM for better performance.
- Implement hyperparameter tuning techniques like GridSearchCV or RandomizedSearchCV for model optimization**.**

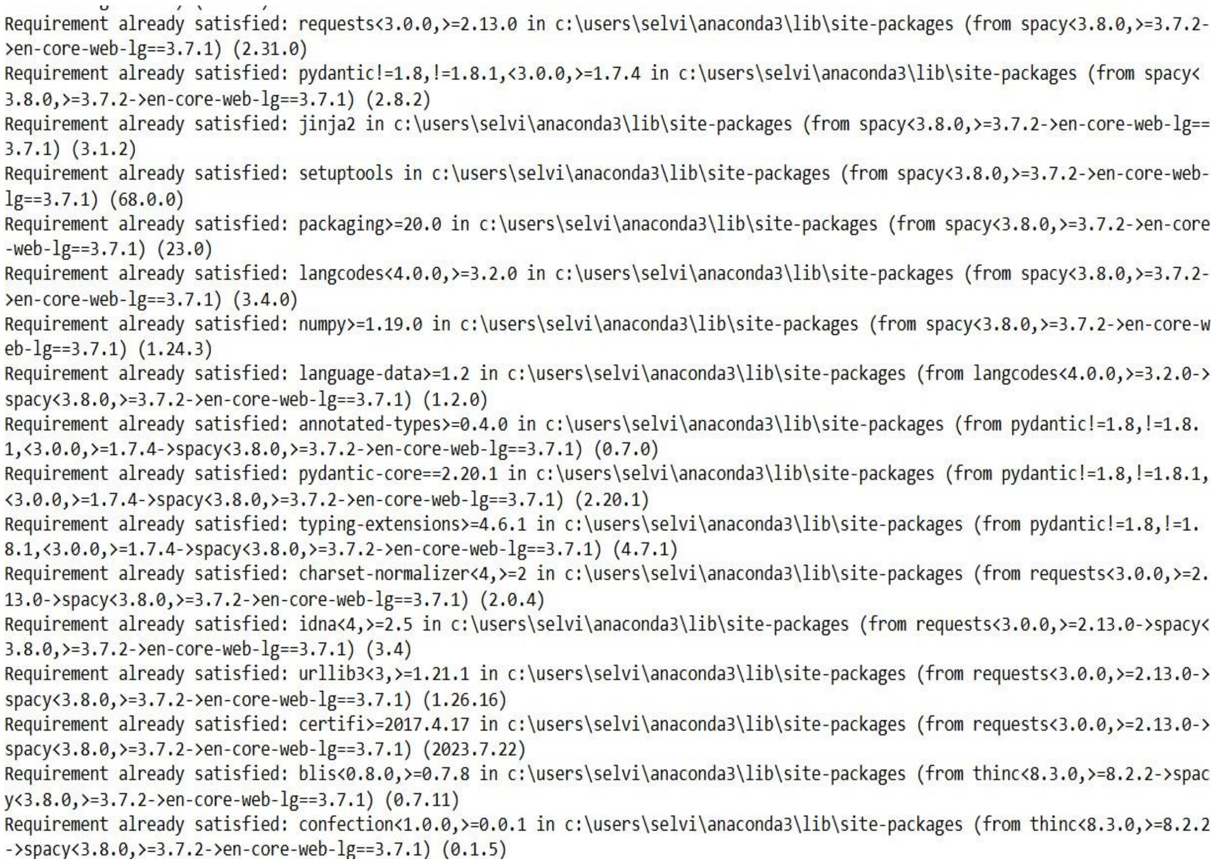
CHAPTER 8

8. Appendix

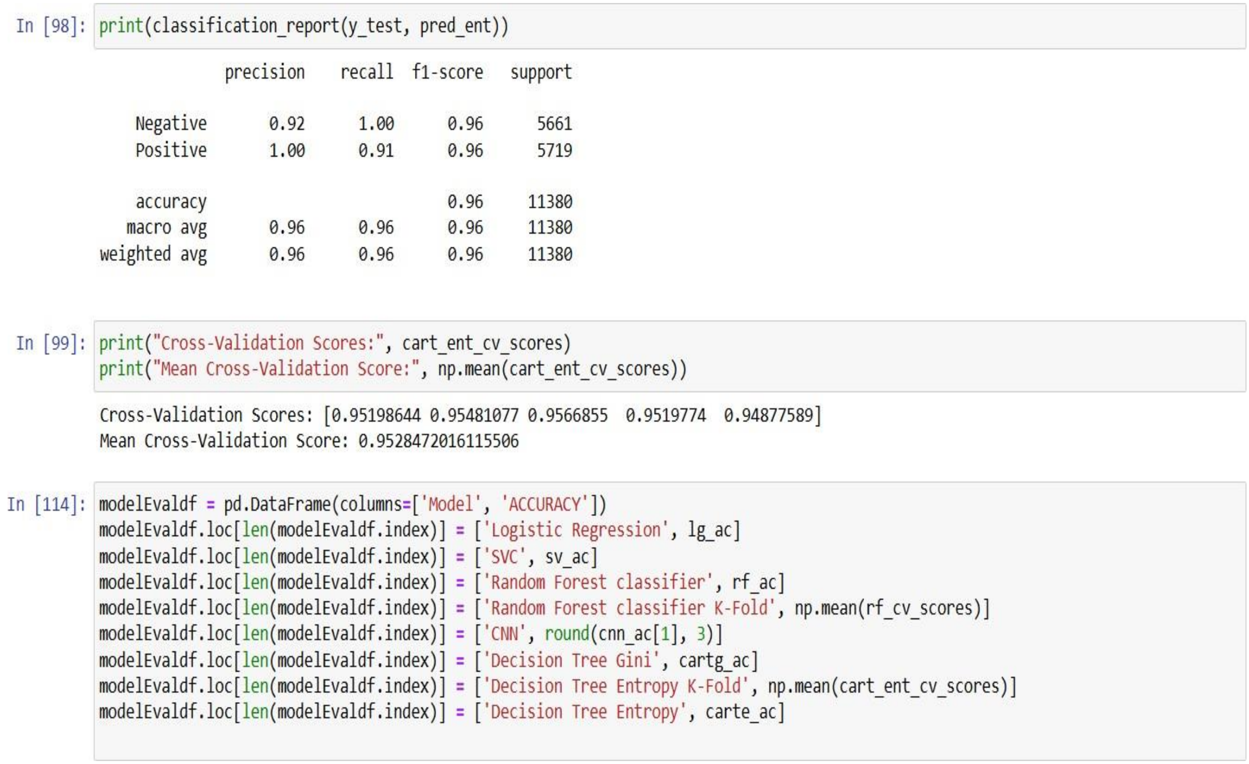
8.1 Coding



```
In [7]: import numpy as np
        import pandas as pd
        import math
        import matplotlib.pyplot as plt
        import seaborn as sns
        import string
        import spacy
        from spacy import displacy
        import nltk
        from wordcloud import WordCloud
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.decomposition import LatentDirichletAllocation
        import re
        from nltk.corpus import stopwords
        from nltk.sentiment.vader import SentimentIntensityAnalyzer
        from textblob import TextBlob
        from sklearn.model_selection import train_test_split
        from imblearn.over_sampling import RandomOverSampler
        from imblearn.under_sampling import RandomUnderSampler
        from sklearn.linear_model import LogisticRegression
        from sklearn.model_selection import cross_val_score
        from sklearn.metrics import mean_squared_error, mean_absolute_error, accuracy_score, confusion_matrix, classification_report

In [4]: !pip install keras
        !pip install tensorflow

In [10]: nlp = spacy.load('en_core_web_lg')

In [9]: !python -m spacy download en_core_web_lg
```



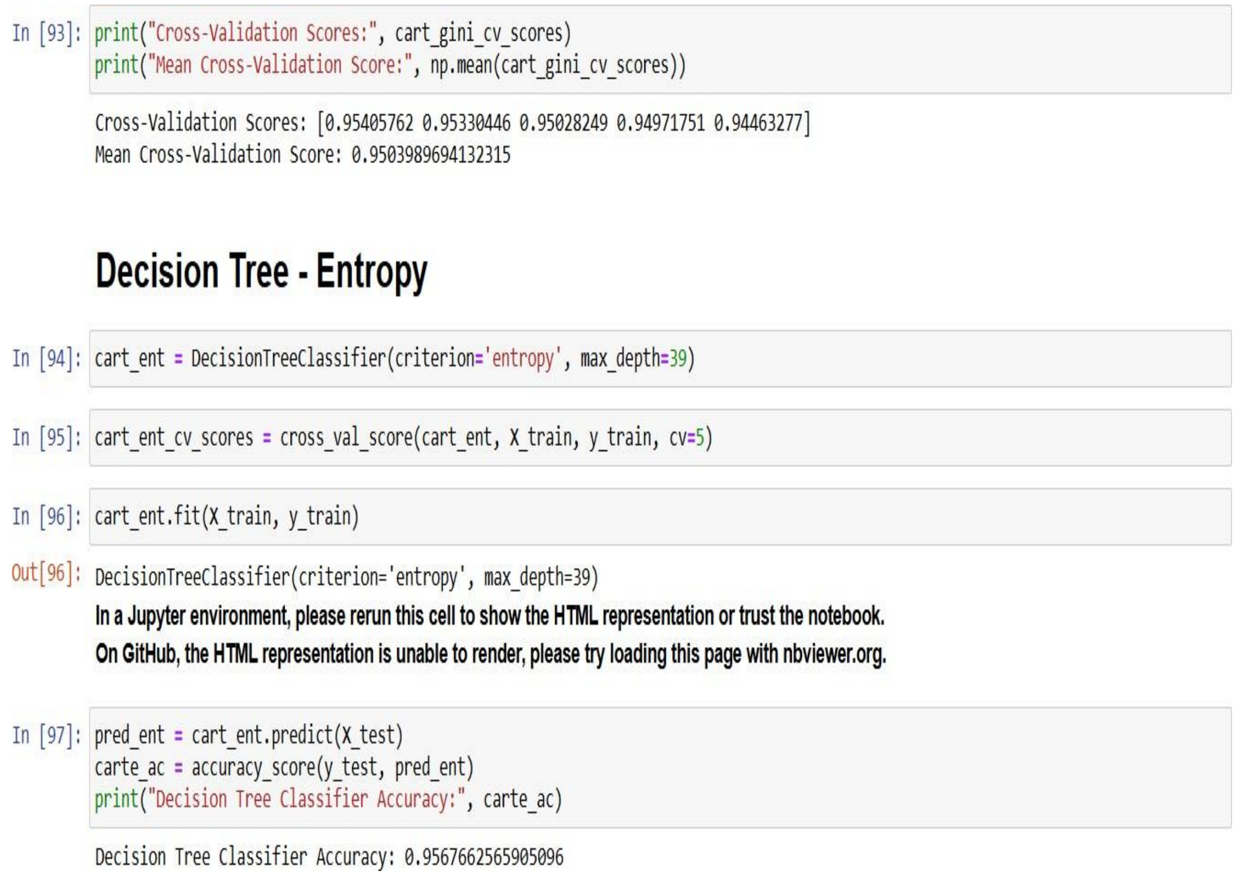
```
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\selvi\anaconda3\lib\site-packages (from spacy<3.8.0,>=3.7.2-
>en-core-web-lg==3.7.1) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in c:\users\selvi\anaconda3\lib\site-packages (from spacy<
3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (2.8.2)
Requirement already satisfied: jinja2 in c:\users\selvi\anaconda3\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-lg==
3.7.1) (3.1.2)
Requirement already satisfied: setuptools in c:\users\selvi\anaconda3\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-
lg==3.7.1) (68.0.0)
Requirement already satisfied: packaging>=20.0 in c:\users\selvi\anaconda3\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core
-web-lg==3.7.1) (23.0)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\selvi\anaconda3\lib\site-packages (from spacy<3.8.0,>=3.7.2-
>en-core-web-lg==3.7.1) (3.4.0)
Requirement already satisfied: numpy>=1.19.0 in c:\users\selvi\anaconda3\lib\site-packages (from spacy<3.8.0,>=3.7.2->en-core-w
eb-lg==3.7.1) (1.24.3)
Requirement already satisfied: language-data>=1.2 in c:\users\selvi\anaconda3\lib\site-packages (from langcodes<4.0.0,>=3.2.0->
spacy<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (1.2.0)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\selvi\anaconda3\lib\site-packages (from pydantic!=1.8,!=1.8.
1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (0.7.0)
Requirement already satisfied: pydantic-core==2.20.1 in c:\users\selvi\anaconda3\lib\site-packages (from pydantic!=1.8,!=1.8.1,
<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (2.20.1)
Requirement already satisfied: typing-extensions>=4.6.1 in c:\users\selvi\anaconda3\lib\site-packages (from pydantic!=1.8,!=1.
8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (4.7.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\selvi\anaconda3\lib\site-packages (from requests<3.0.0,>=2.
13.0->spacy<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\selvi\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy<
3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\selvi\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->
spacy<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\selvi\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->
spacy<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (2023.7.22)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in c:\users\selvi\anaconda3\lib\site-packages (from thinc<8.3.0,>=8.2.2->spac
y<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\selvi\anaconda3\lib\site-packages (from thinc<8.3.0,>=8.2.2
->spacy<3.8.0,>=3.7.2->en-core-web-lg==3.7.1) (0.1.5)
```



```
In [98]: print(classification_report(y_test, pred_ent))

              precision    recall  f1-score   support

    Negative       0.92      1.00      0.96      5661
    Positive       1.00      0.91      0.96      5719

    accuracy                           0.96     11380
   macro avg       0.96      0.96      0.96     11380
weighted avg       0.96      0.96      0.96     11380
```

```
In [99]: print("Cross-Validation Scores:", cart_ent_cv_scores)
         print("Mean Cross-Validation Score:", np.mean(cart_ent_cv_scores))

Cross-Validation Scores: [0.95198644 0.95481077 0.9566855  0.9519774  0.94877589]
Mean Cross-Validation Score: 0.9528472016115506
```

```
In [114]: modelEvaldf = pd.DataFrame(columns=['Model', 'ACCURACY'])
          modelEvaldf.loc[len(modelEvaldf.index)] = ['Logistic Regression', lg_ac]
          modelEvaldf.loc[len(modelEvaldf.index)] = ['SVC', sv_ac]
          modelEvaldf.loc[len(modelEvaldf.index)] = ['Random Forest classifier', rf_ac]
          modelEvaldf.loc[len(modelEvaldf.index)] = ['Random Forest classifier K-Fold', np.mean(rf_cv_scores)]
          modelEvaldf.loc[len(modelEvaldf.index)] = ['CNN', round(cnn_ac[1], 3)]
          modelEvaldf.loc[len(modelEvaldf.index)] = ['Decision Tree Gini', cartg_ac]
          modelEvaldf.loc[len(modelEvaldf.index)] = ['Decision Tree Entropy K-Fold', np.mean(cart_ent_cv_scores)]
          modelEvaldf.loc[len(modelEvaldf.index)] = ['Decision Tree Entropy', carte_ac]
```



```
In [93]: print("Cross-Validation Scores:", cart_gini_cv_scores)
         print("Mean Cross-Validation Score:", np.mean(cart_gini_cv_scores))

Cross-Validation Scores: [0.95405762 0.95330446 0.95028249 0.94971751 0.94463277]
Mean Cross-Validation Score: 0.9503989694132315
```

Decision Tree - Entropy

```
In [94]: cart_ent = DecisionTreeClassifier(criterion='entropy', max_depth=39)
```

```
In [95]: cart_ent_cv_scores = cross_val_score(cart_ent, X_train, y_train, cv=5)
```

```
In [96]: cart_ent.fit(X_train, y_train)

Out[96]: DecisionTreeClassifier(criterion='entropy', max_depth=39)
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
In [97]: pred_ent = cart_ent.predict(X_test)
         carte_ac = accuracy_score(y_test, pred_ent)
         print("Decision Tree Classifier Accuracy:", carte_ac)

Decision Tree Classifier Accuracy: 0.9567662565905096
```

Decision Tree - Gini

```
In [88]: cart_gini = DecisionTreeClassifier(criterion='gini', max_depth=39)
```

```
In [89]: cart_gini_cv_scores = cross_val_score(cart_gini, X_train, y_train, cv=5)
```

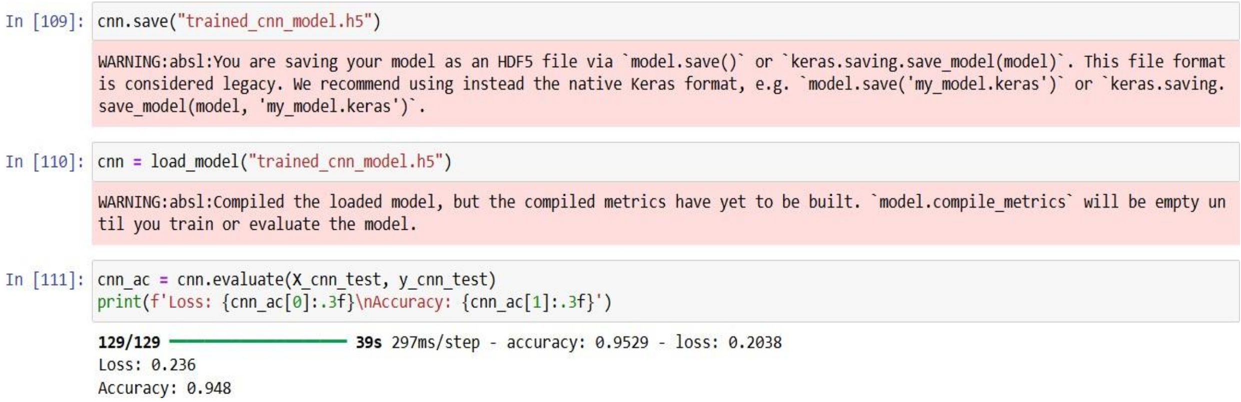
```
In [90]: cart_gini.fit(X_train, y_train)
```

```
Out[90]: DecisionTreeClassifier(max_depth=39)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [91]: pred_gini = cart_gini.predict(X_test)
         cartg_ac = accuracy_score(y_test, pred_gini)
         print("Decision Tree Classifier Accuracy:", cartg_ac)

         Decision Tree Classifier Accuracy: 0.9493848857644991
```

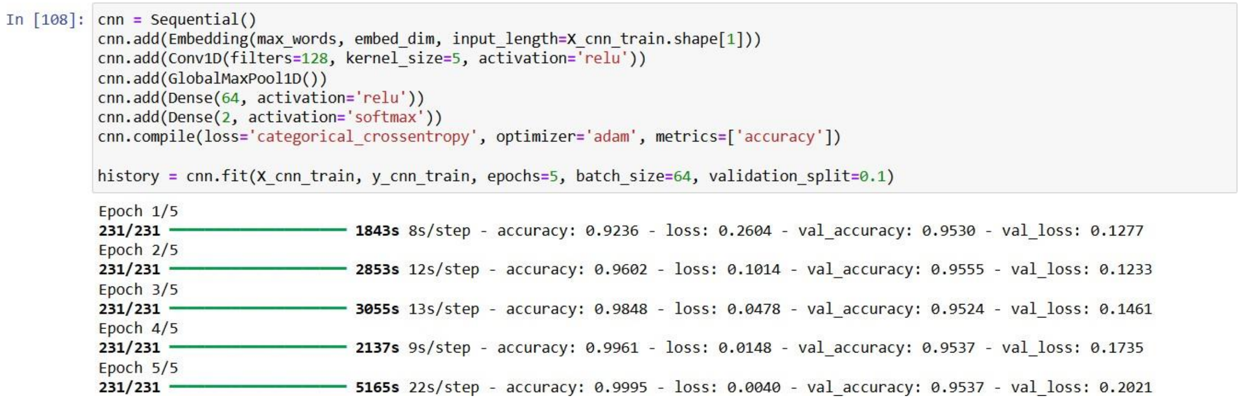
```
In [92]: print(classification_report(y_test, pred_gini))

                       precision    recall  f1-score   support

             Negative       0.91      1.00      0.95      5661
             Positive       1.00      0.90      0.95      5719

             accuracy                           0.95     11380
            macro avg       0.95      0.95      0.95     11380
         weighted avg       0.95      0.95      0.95     11380
```



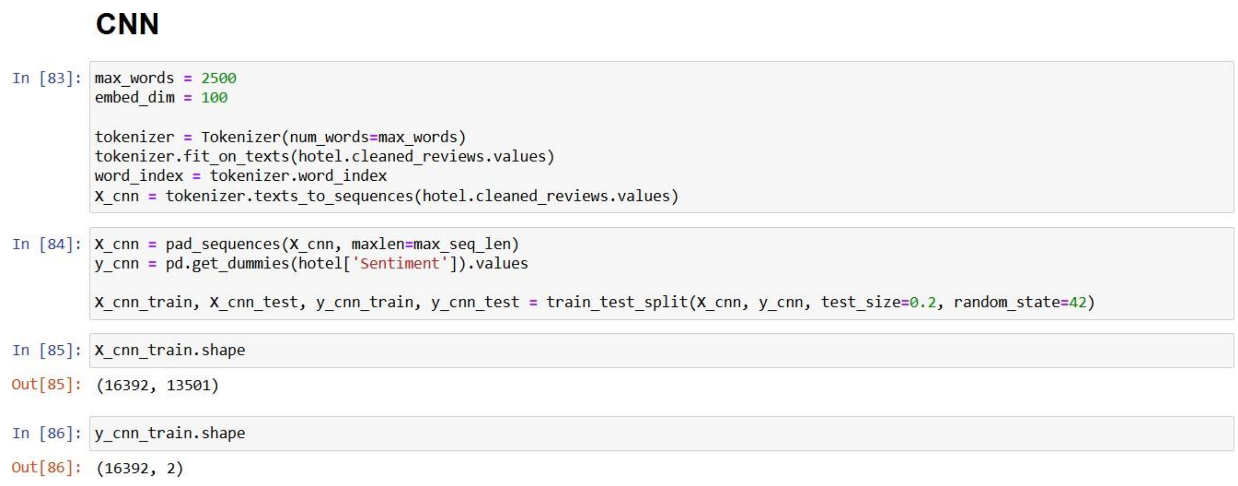
```
In [109]: cnn.save("trained_cnn_model.h5")

          WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format
          is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.
          save_model(model, 'my_model.keras')`.
```

```
In [110]: cnn = load_model("trained_cnn_model.h5")

          WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty un
          til you train or evaluate the model.
```

```
In [111]: cnn_ac = cnn.evaluate(X_cnn_test, y_cnn_test)
          print(f'Loss: {cnn_ac[0]:.3f}\nAccuracy: {cnn_ac[1]:.3f}')

          129/129 ━━━━━━━━━━━━━━━━━━━━ 39s 297ms/step - accuracy: 0.9529 - loss: 0.2038
          Loss: 0.236
          Accuracy: 0.948
```

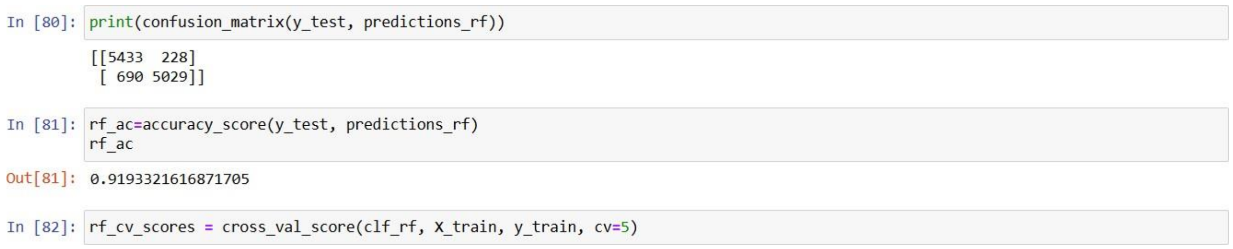


```
In [108]: cnn = Sequential()
          cnn.add(Embedding(max_words, embed_dim, input_length=X_cnn_train.shape[1]))
          cnn.add(Conv1D(filters=128, kernel_size=5, activation='relu'))
          cnn.add(GlobalMaxPool1D())
          cnn.add(Dense(64, activation='relu'))
          cnn.add(Dense(2, activation='softmax'))
          cnn.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

          history = cnn.fit(X_cnn_train, y_cnn_train, epochs=5, batch_size=64, validation_split=0.1)

          Epoch 1/5
          231/231 ━━━━━━━━━━━━━━━━━━━━ 1843s 8s/step - accuracy: 0.9236 - loss: 0.2604 - val_accuracy: 0.9530 - val_loss: 0.1277
          Epoch 2/5
          231/231 ━━━━━━━━━━━━━━━━━━━━ 2853s 12s/step - accuracy: 0.9602 - loss: 0.1014 - val_accuracy: 0.9555 - val_loss: 0.1233
          Epoch 3/5
          231/231 ━━━━━━━━━━━━━━━━━━━━ 3055s 13s/step - accuracy: 0.9848 - loss: 0.0478 - val_accuracy: 0.9524 - val_loss: 0.1461
          Epoch 4/5
          231/231 ━━━━━━━━━━━━━━━━━━━━ 2137s 9s/step - accuracy: 0.9961 - loss: 0.0148 - val_accuracy: 0.9537 - val_loss: 0.1735
          Epoch 5/5
          231/231 ━━━━━━━━━━━━━━━━━━━━ 5165s 22s/step - accuracy: 0.9995 - loss: 0.0040 - val_accuracy: 0.9537 - val_loss: 0.2021
```

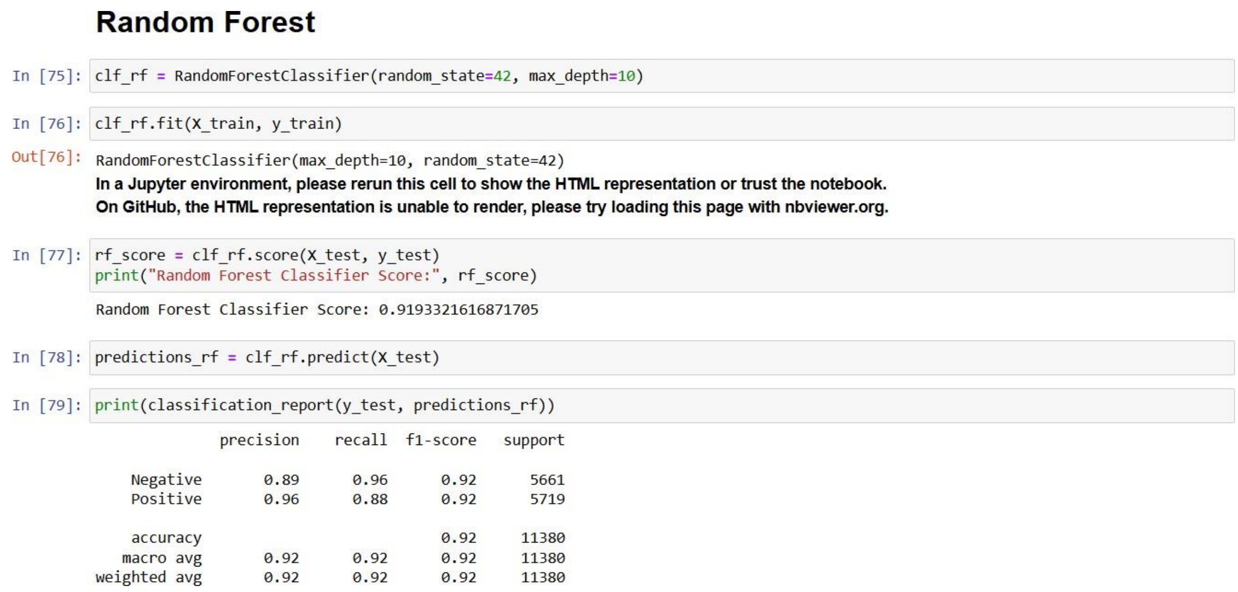


CNN

```
In [83]: max_words = 2500
         embed_dim = 100

         tokenizer = Tokenizer(num_words=max_words)
         tokenizer.fit_on_texts(hotel.cleaned_reviews.values)
         word_index = tokenizer.word_index
         X_cnn = tokenizer.texts_to_sequences(hotel.cleaned_reviews.values)

In [84]: X_cnn = pad_sequences(X_cnn, maxlen=max_seq_len)
         y_cnn = pd.get_dummies(hotel['Sentiment']).values

         X_cnn_train, X_cnn_test, y_cnn_train, y_cnn_test = train_test_split(X_cnn, y_cnn, test_size=0.2, random_state=42)

In [85]: X_cnn_train.shape

Out[85]: (16392, 13501)

In [86]: y_cnn_train.shape

Out[86]: (16392, 2)
```



```
In [80]: print(confusion_matrix(y_test, predictions_rf))

         [[5433  228]
          [ 690 5029]]

In [81]: rf_ac=accuracy_score(y_test, predictions_rf)
         rf_ac

Out[81]: 0.9193321616871705

In [82]: rf_cv_scores = cross_val_score(clf_rf, X_train, y_train, cv=5)
```

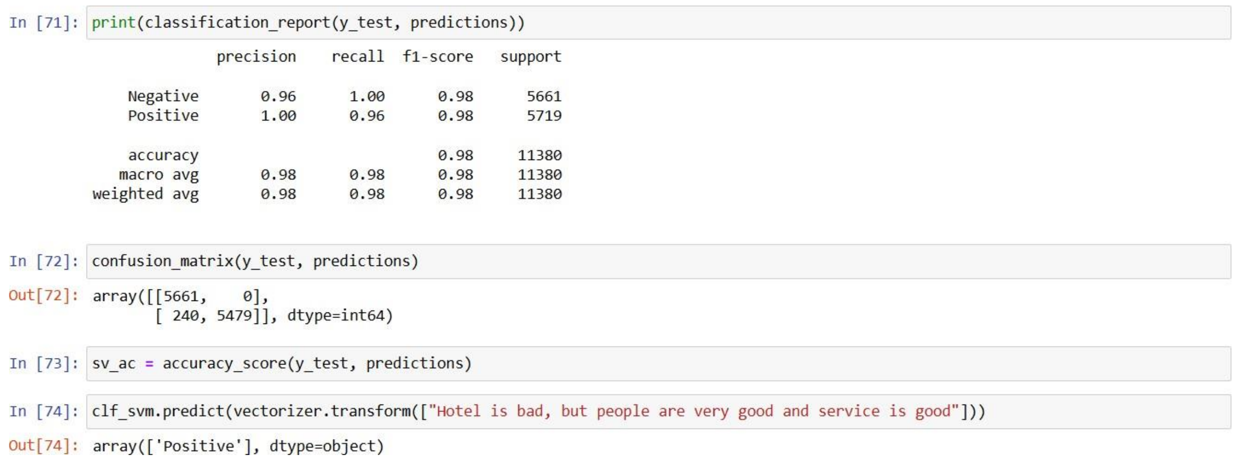


Random Forest

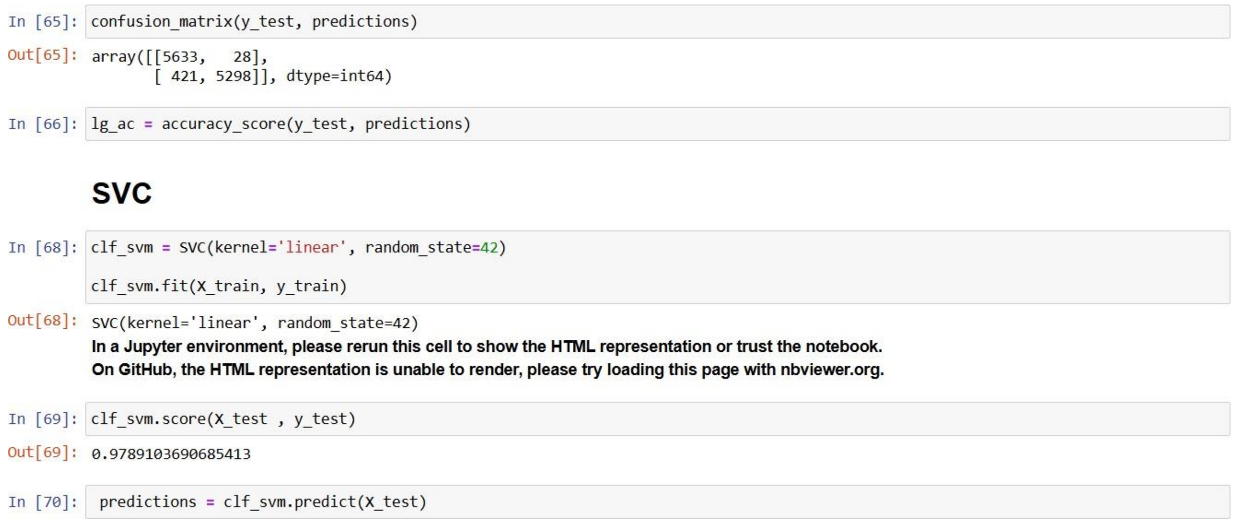
```
In [75]: clf_rf = RandomForestClassifier(random_state=42, max_depth=10)

In [76]: clf_rf.fit(X_train, y_train)

Out[76]: RandomForestClassifier(max_depth=10, random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [77]: rf_score = clf_rf.score(X_test, y_test)
         print("Random Forest Classifier Score:", rf_score)

         Random Forest Classifier Score: 0.9193321616871705

In [78]: predictions_rf = clf_rf.predict(X_test)

In [79]: print(classification_report(y_test, predictions_rf))

                       precision    recall  f1-score   support

             Negative       0.89      0.96      0.92      5661
             Positive       0.96      0.88      0.92      5719

             accuracy                           0.92     11380
            macro avg       0.92      0.92      0.92     11380
         weighted avg       0.92      0.92      0.92     11380
```



```
In [71]: print(classification_report(y_test, predictions))

                  precision    recall  f1-score   support

        Negative       0.96      1.00      0.98      5661
        Positive       1.00      0.96      0.98      5719

        accuracy                           0.98     11380
       macro avg       0.98      0.98      0.98     11380
    weighted avg       0.98      0.98      0.98     11380

In [72]: confusion_matrix(y_test, predictions)

Out[72]: array([[5661,    0],
                [ 240, 5479]], dtype=int64)

In [73]: sv_ac = accuracy_score(y_test, predictions)

In [74]: clf_svm.predict(vectorizer.transform(["Hotel is bad, but people are very good and service is good"]))

Out[74]: array(['Positive'], dtype=object)
```



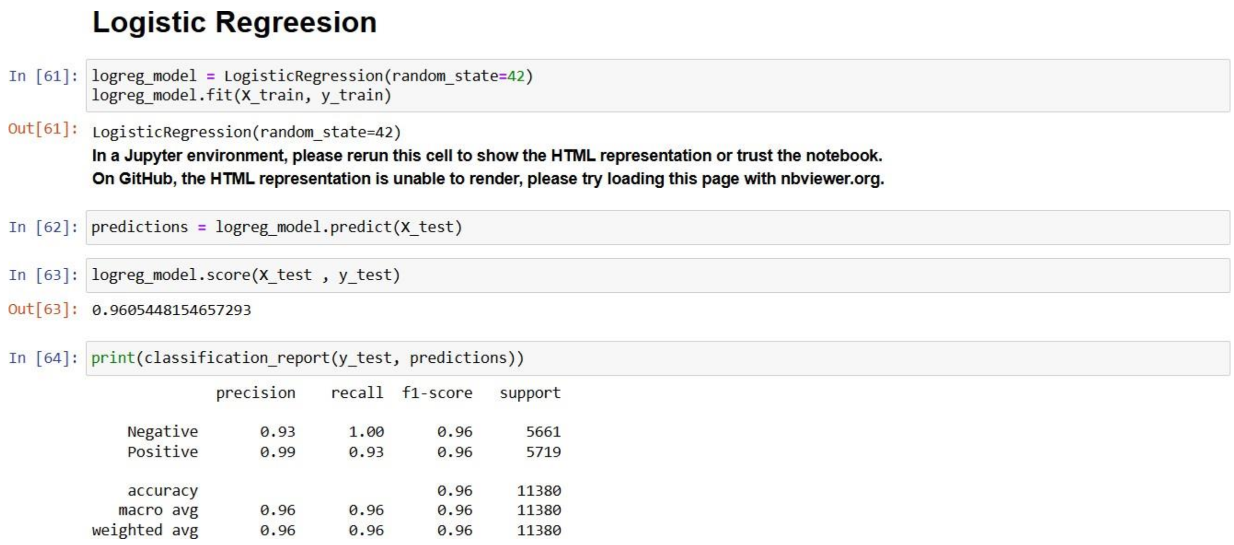
```
In [65]: confusion_matrix(y_test, predictions)

Out[65]: array([[5633,   28],
                [ 421, 5298]], dtype=int64)

In [66]: lg_ac = accuracy_score(y_test, predictions)
```

SVC

```
In [68]: clf_svm = SVC(kernel='linear', random_state=42)

         clf_svm.fit(X_train, y_train)

Out[68]: SVC(kernel='linear', random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [69]: clf_svm.score(X_test , y_test)

Out[69]: 0.9789103690685413

In [70]:  predictions = clf_svm.predict(X_test)
```



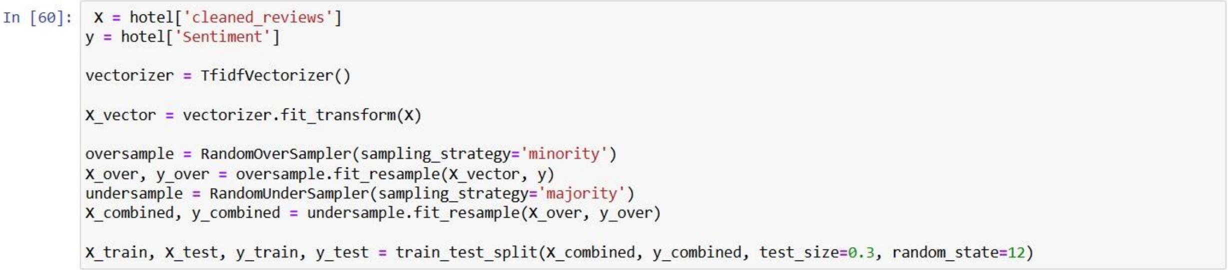
Logistic Regreesion

```
In [61]: logreg_model = LogisticRegression(random_state=42)
         logreg_model.fit(X_train, y_train)

Out[61]: LogisticRegression(random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

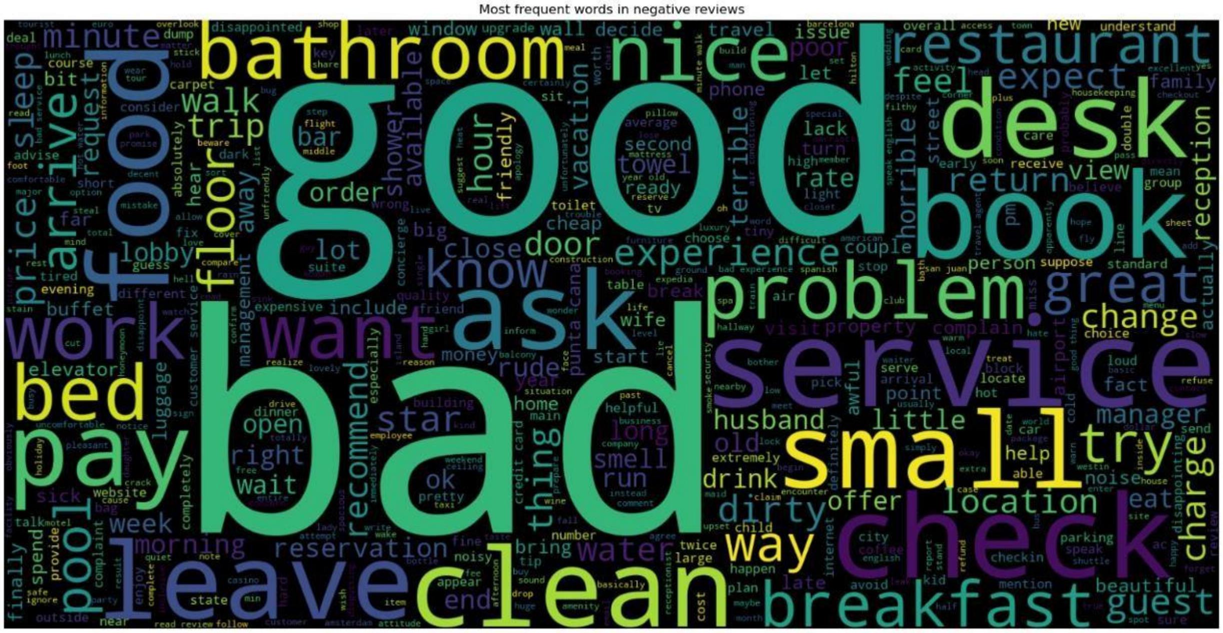
```
In [62]: predictions = logreg_model.predict(X_test)

In [63]: logreg_model.score(X_test , y_test)

Out[63]: 0.9605448154657293

In [64]: print(classification_report(y_test, predictions))

                  precision    recall  f1-score   support

        Negative       0.93      1.00      0.96      5661
        Positive       0.99      0.93      0.96      5719

        accuracy                           0.96     11380
       macro avg       0.96      0.96      0.96     11380
    weighted avg       0.96      0.96      0.96     11380
```

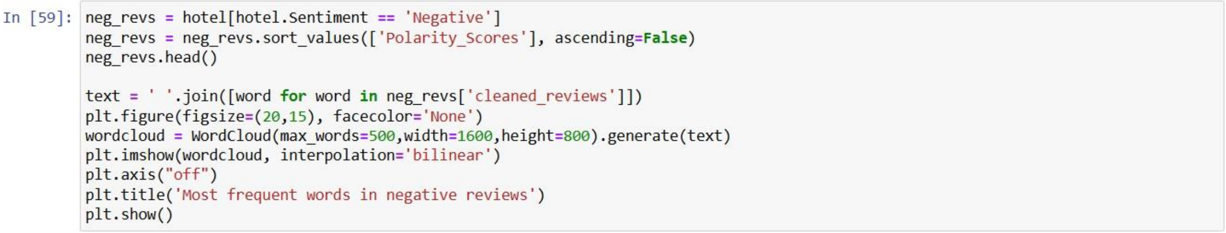
Vectorization



In [60]:
```
 X = hotel['cleaned_reviews']
y = hotel['Sentiment']

vectorizer = TfidfVectorizer()

X_vector = vectorizer.fit_transform(X)

oversample = RandomOverSampler(sampling_strategy='minority')
X_over, y_over = oversample.fit_resample(X_vector, y)
undersample = RandomUnderSampler(sampling_strategy='majority')
X_combined, y_combined = undersample.fit_resample(X_over, y_over)

X_train, X_test, y_train, y_test = train_test_split(X_combined, y_combined, test_size=0.3, random_state=12)
```

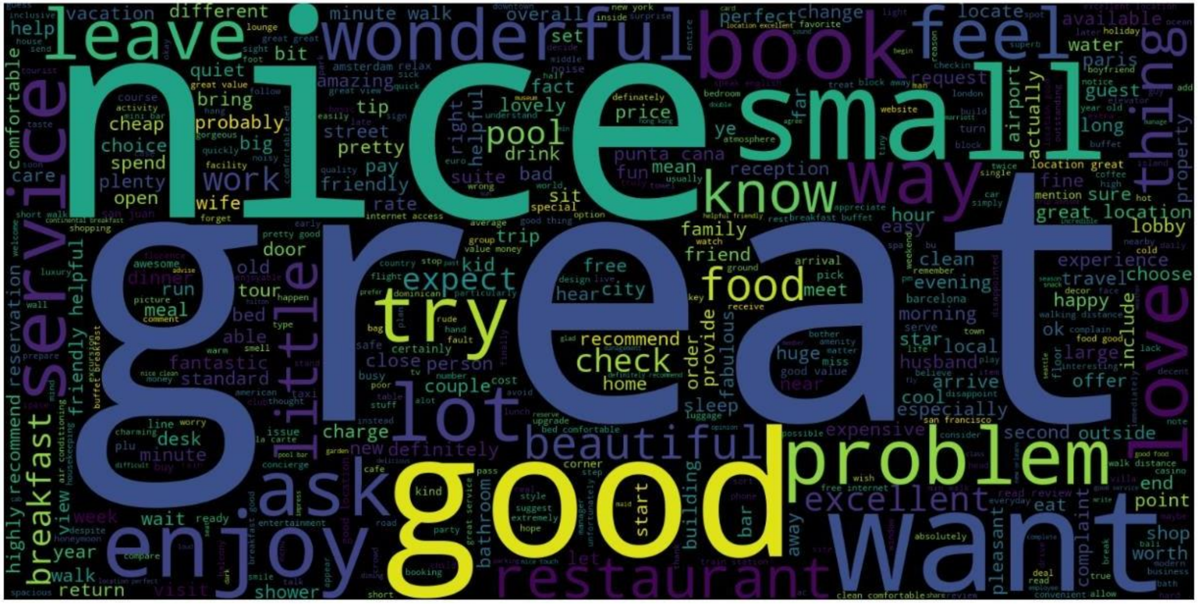
Most frequent words in negative reviews





In [59]:
```
neg_revs = hotel[hotel.Sentiment == 'Negative']
neg_revs = neg_revs.sort_values(['Polarity_Scores'], ascending=False)
neg_revs.head()

text = ' '.join([word for word in neg_revs['cleaned_reviews']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud = WordCloud(max_words=500,width=1600,height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Most frequent words in negative reviews')
plt.show()
```

Most frequent words in positive reviews



```
In [58]: text = ' '.join([word for word in pos_revs['cleaned_reviews']])
         plt.figure(figsize=(20,15), facecolor='None')
         wordcloud = WordCloud(max_words=500,width=1600,height=800).generate(text)
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis("off")
         plt.title('Most frequent words in positive reviews')
         plt.show()
```

```
In [55]: hotel['Rating'][hotel['Polarity_Scores'] > 0].value_counts()
```

```
Out[55]: 5    9006
         4    5972
         3    2016
         2    1321
         1     640
         Name: Rating, dtype: int64
```

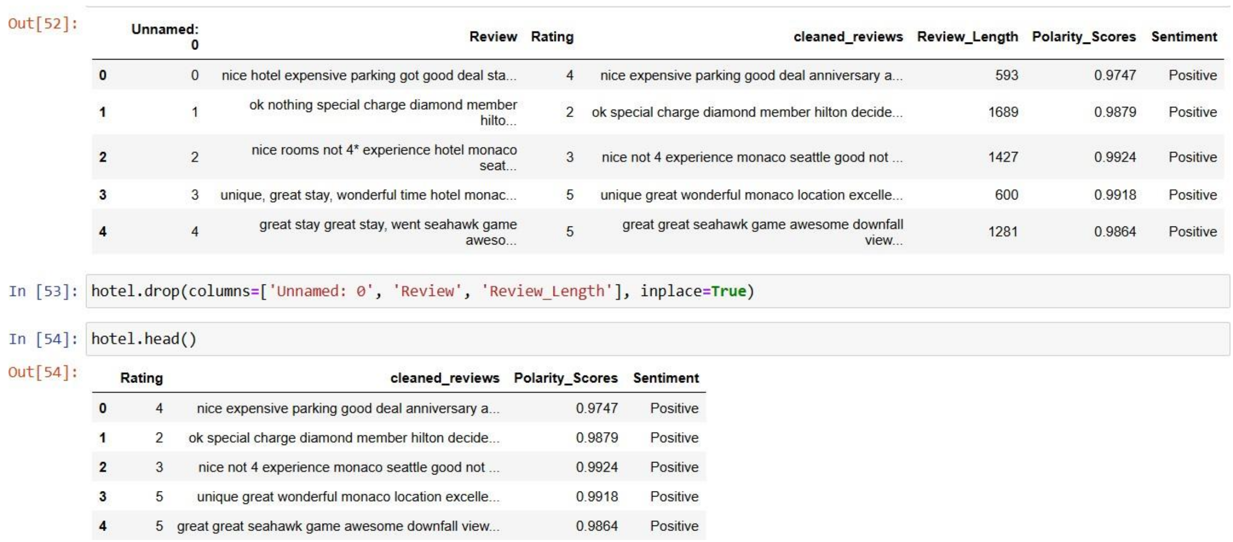
```
In [56]: hotel['Rating'][hotel['Polarity_Scores'] < 0].value_counts()
```

```
Out[56]: 1    778
         2    470
         3    165
         4     66
         5     46
         Name: Rating, dtype: int64
```

```
In [57]: pos_revs = hotel[hotel.Sentiment == 'Positive']
         pos_revs = pos_revs.sort_values(['Polarity_Scores'], ascending=False)
         pos_revs.head()
```

Out[57]:

	Rating	cleaned_reviews	Polarity_Scores	Sentiment
18917	5	ocean blue excellent boyfriend mid20 ocean blu...	0.9999	Positive
2967	4	bravo bavaro knew not fivestar incredibly reas...	0.9998	Positive
17777	5	fantastic return trip firstly apology length r...	0.9998	Positive
14116	5	not wait till year 2nd 16th 2008we lofts week ...	0.9998	Positive
14183	4	awesomebeautiful vacation little boonie gran m...	0.9998	Positive



Out[52]:

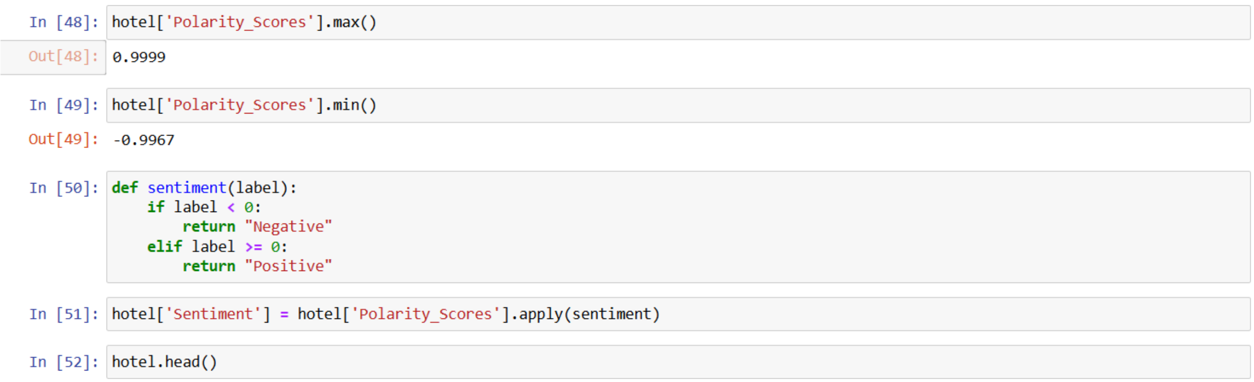
	Unnamed: 0	Review	Rating	cleaned_reviews	Review_Length	Polarity_Scores	Sentiment
0	0	nice hotel expensive parking got good deal sta...	4	nice expensive parking good deal anniversary a...	593	0.9747	Positive
1	1	ok nothing special charge diamond member hilto...	2	ok special charge diamond member hilton decide...	1689	0.9879	Positive
2	2	nice rooms not 4* experience hotel monaco seat...	3	nice not 4 experience monaco seattle good not ...	1427	0.9924	Positive
3	3	unique, great stay, wonderful time hotel monac...	5	unique great wonderful monaco location excelle...	600	0.9918	Positive
4	4	great stay great stay, went seahawk game aweso...	5	great great seahawk game awesome downfall view...	1281	0.9864	Positive

```
In [53]: hotel.drop(columns=['Unnamed: 0', 'Review', 'Review_Length'], inplace=True)
```

```
In [54]: hotel.head()
```

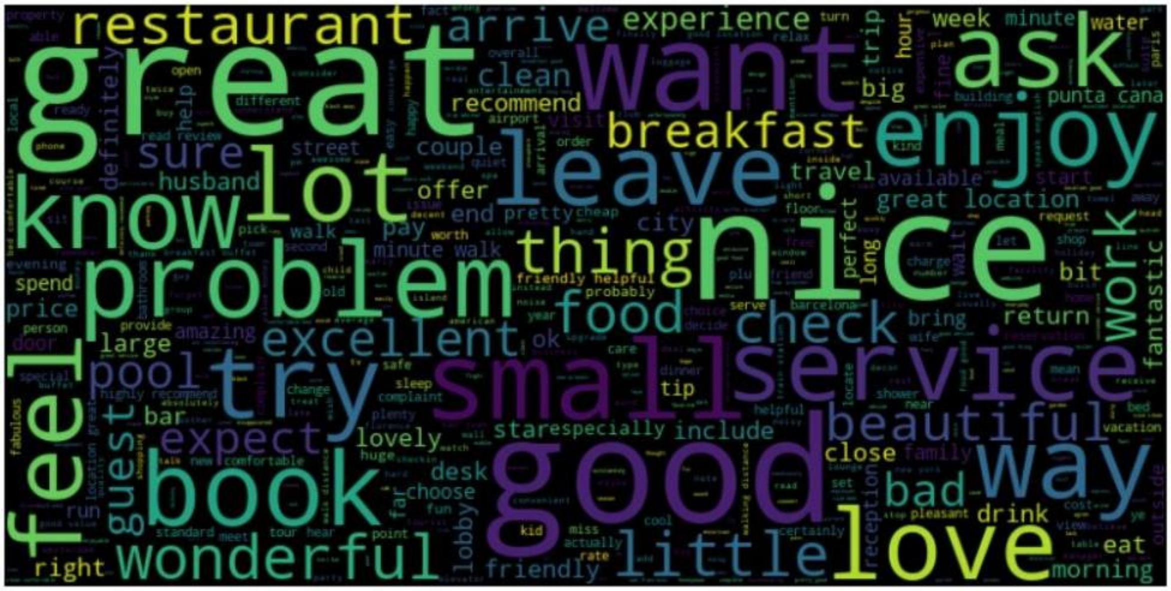
Out[54]:

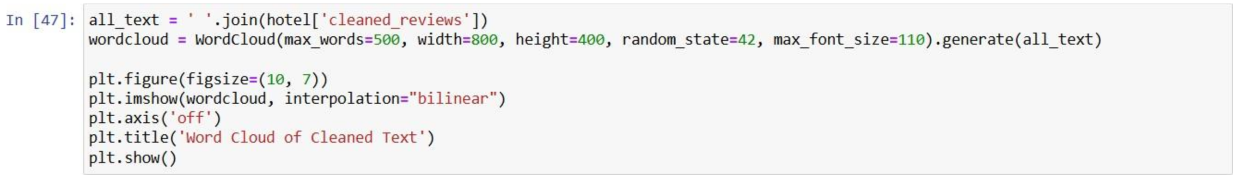
	Rating	cleaned_reviews	Polarity_Scores	Sentiment
0	4	nice expensive parking good deal anniversary a...	0.9747	Positive
1	2	ok special charge diamond member hilton decide...	0.9879	Positive
2	3	nice not 4 experience monaco seattle good not ...	0.9924	Positive
3	5	unique great wonderful monaco location excelle...	0.9918	Positive
4	5	great great seahawk game awesome downfall view...	0.9864	Positive



```
In [48]: hotel['Polarity_Scores'].max()
Out[48]: 0.9999

In [49]: hotel['Polarity_Scores'].min()
Out[49]: -0.9967

In [50]: def sentiment(label):
             if label < 0:
                 return "Negative"
             elif label >= 0:
                 return "Positive"

In [51]: hotel['Sentiment'] = hotel['Polarity_Scores'].apply(sentiment)

In [52]: hotel.head()
```

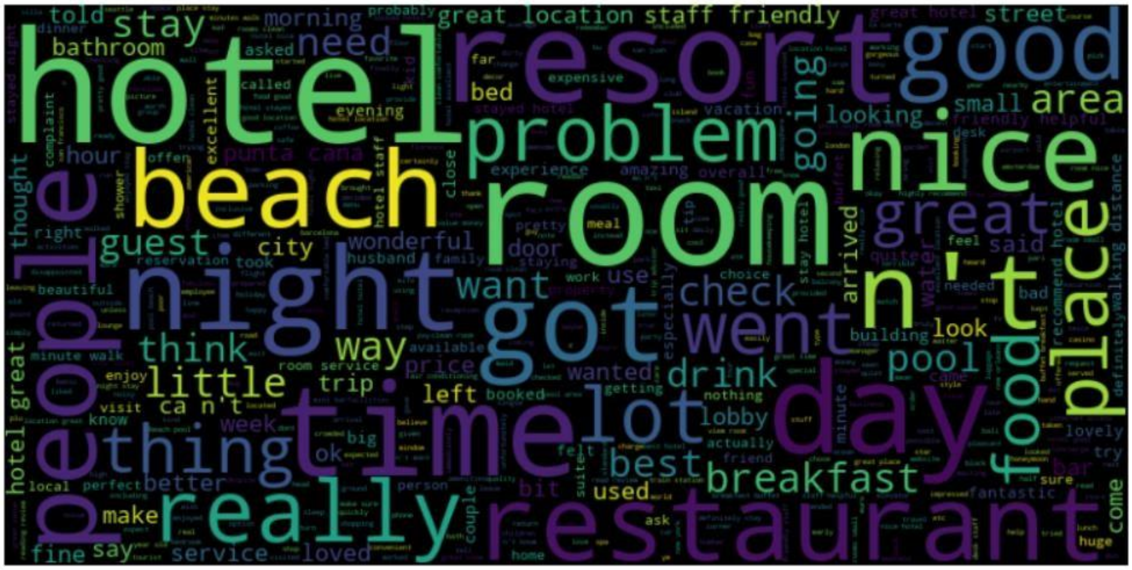
Word Cloud of Cleaned Text

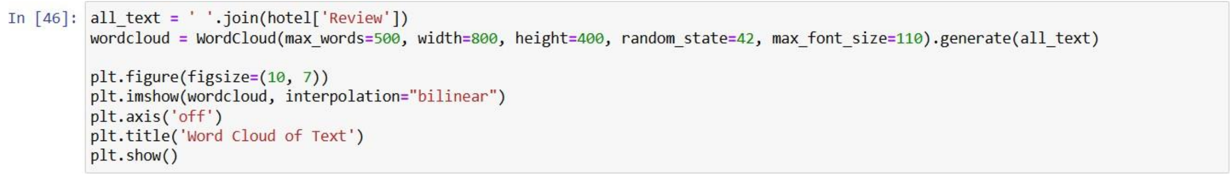




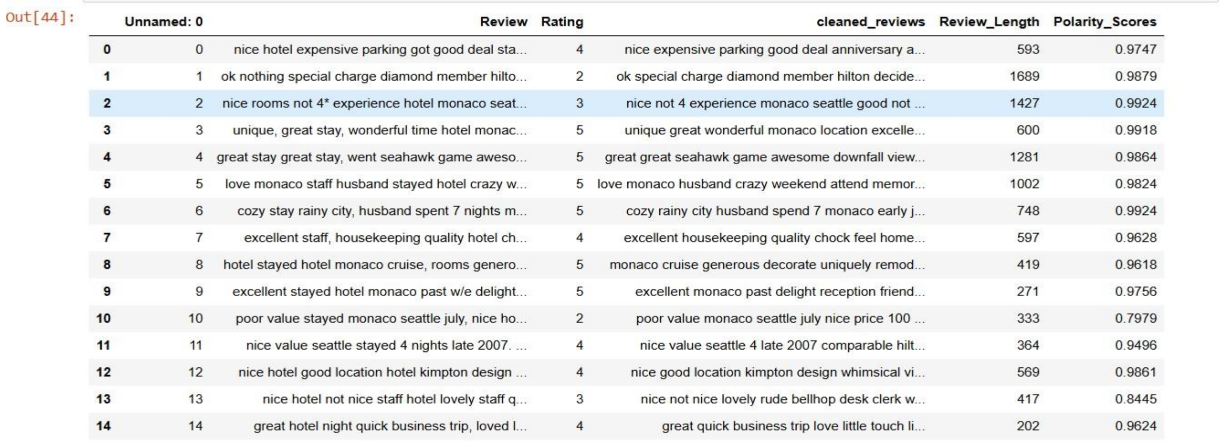
```
In [47]: all_text = ' '.join(hotel['cleaned_reviews'])
         wordcloud = WordCloud(max_words=500, width=800, height=400, random_state=42, max_font_size=110).generate(all_text)

         plt.figure(figsize=(10, 7))
         plt.imshow(wordcloud, interpolation="bilinear")
         plt.axis('off')
         plt.title('Word Cloud of Cleaned Text')
         plt.show()
```

Word Cloud of Text



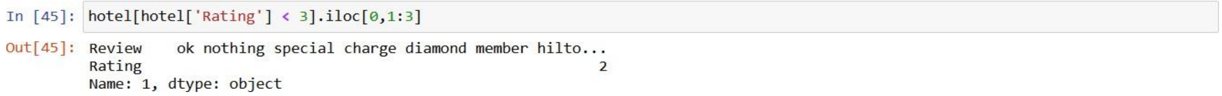


```
In [46]: all_text = ' '.join(hotel['Review'])
         wordcloud = WordCloud(max_words=500, width=800, height=400, random_state=42, max_font_size=110).generate(all_text)

         plt.figure(figsize=(10, 7))
         plt.imshow(wordcloud, interpolation="bilinear")
         plt.axis('off')
         plt.title('Word Cloud of Text')
         plt.show()
```



Out[44]:

	Unnamed: 0	Review	Rating	cleaned_reviews	Review_Length	Polarity_Scores
0	0	nice hotel expensive parking got good deal sta...	4	nice expensive parking good deal anniversary a...	593	0.9747
1	1	ok nothing special charge diamond member hilto...	2	ok special charge diamond member hilton decide...	1689	0.9879
2	2	nice rooms not 4* experience hotel monaco seat...	3	nice not 4 experience monaco seattle good not ...	1427	0.9924
3	3	unique, great stay, wonderful time hotel monac...	5	unique great wonderful monaco location excelle...	600	0.9918
4	4	great stay great stay, went seahawk game aweso...	5	great great seahawk game awesome downfall view...	1281	0.9864
5	5	love monaco staff husband stayed hotel crazy w...	5	love monaco husband crazy weekend attend memor...	1002	0.9824
6	6	cozy stay rainy city, husband spent 7 nights m...	5	cozy rainy city husband spend 7 monaco early j...	748	0.9924
7	7	excellent staff, housekeeping quality hotel ch...	4	excellent housekeeping quality chock feel home...	597	0.9628
8	8	hotel stayed hotel monaco cruise, rooms genero...	5	monaco cruise generous decorate uniquely remod...	419	0.9618
9	9	excellent stayed hotel monaco past w/e delight...	5	excellent monaco past delight reception friend...	271	0.9756
10	10	poor value stayed monaco seattle july, nice ho...	2	poor value monaco seattle july nice price 100 ...	333	0.7979
11	11	nice value seattle stayed 4 nights late 2007. ...	4	nice value seattle 4 late 2007 comparable hilt...	364	0.9496
12	12	nice hotel good location hotel kimpton design ...	4	nice good location kimpton design whimsical vi...	569	0.9861
13	13	nice hotel not nice staff hotel lovely staff q...	3	nice not nice lovely rude bellhop desk clerk w...	417	0.8445
14	14	great hotel night quick business trip, loved l...	4	great quick business trip love little touch li...	202	0.9624



```
In [45]: hotel[hotel['Rating'] < 3].iloc[0,1:3]

Out[45]: Review    ok nothing special charge diamond member hilto...
         Rating                                                   2
         Name: 1, dtype: object
```

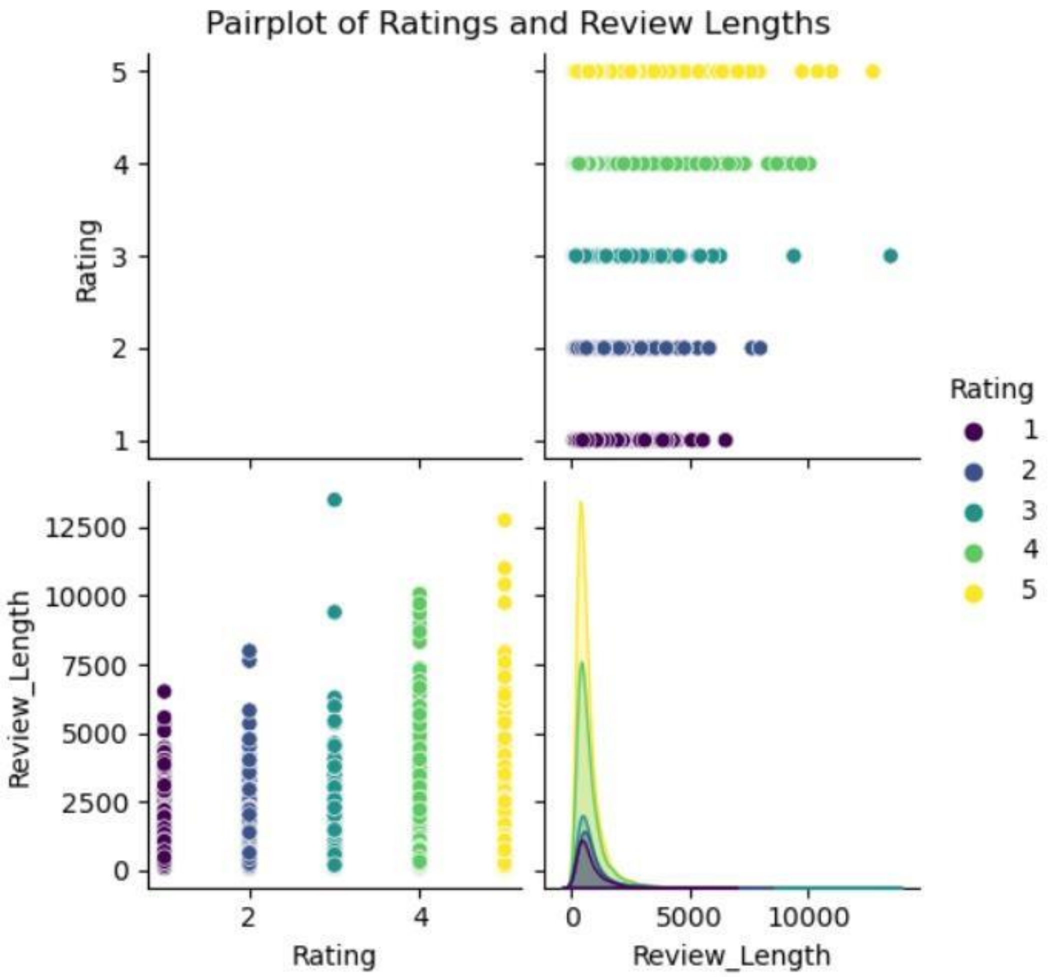
Sentiment Analysis - Polarity Scores

```
In [42]: sid = SentimentIntensityAnalyzer()
         def polarity(text):
             return TextBlob(text).sentiment.polarity
```

```
In [41]: import nltk
         nltk.download('vader_lexicon')

         [nltk_data] Downloading package vader_lexicon to
         [nltk_data]     C:\Users\selvi\AppData\Roaming\nltk_data...

Out[41]: True
```

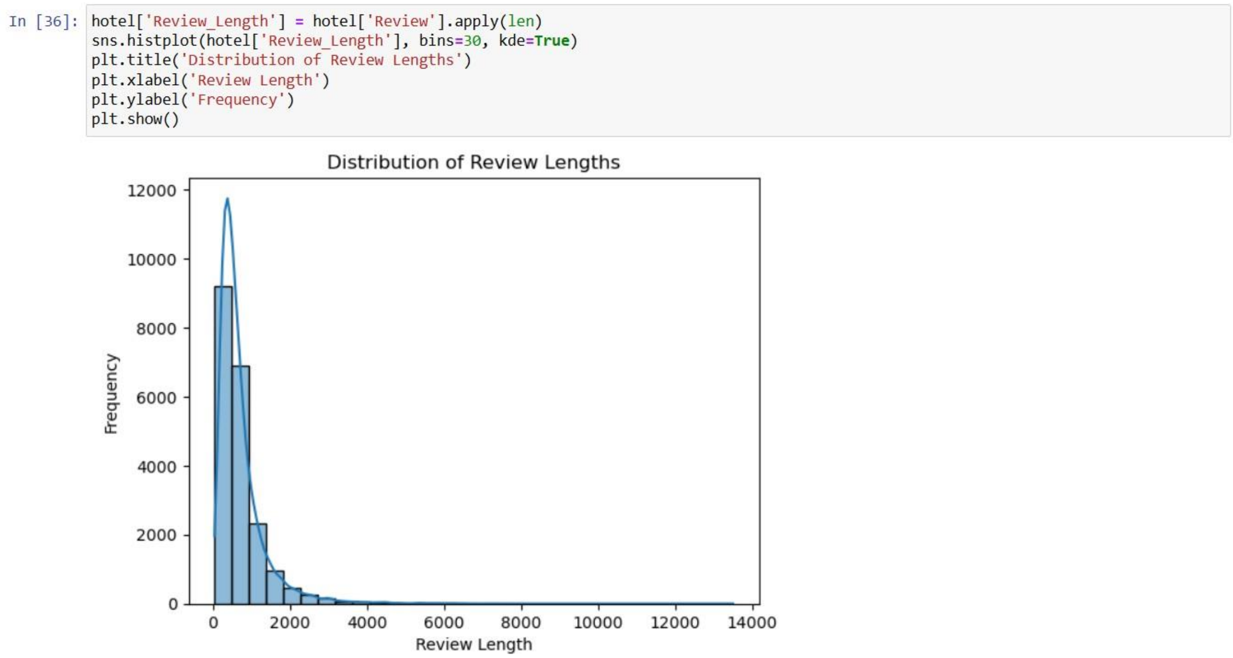
```
In [43]: hotel['Polarity_Scores'] = hotel['cleaned_reviews'].apply(lambda review: sid.polarity_scores(review)['compound'])
```

```
In [44]: hotel.head(15)
```

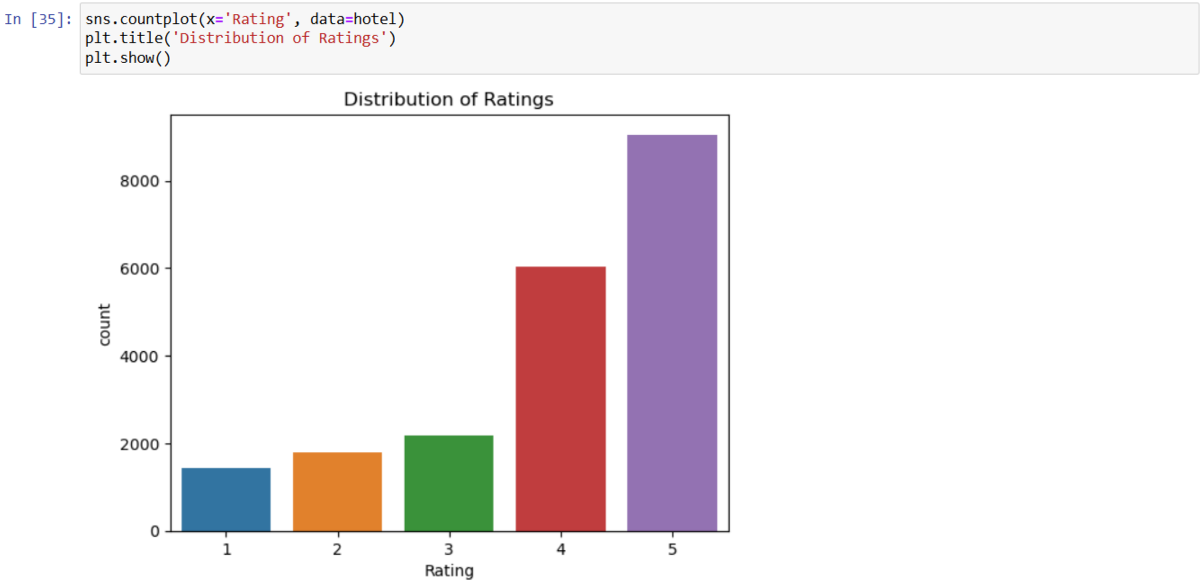


```
In [37]: max_seq_len = hotel['Review_Length'].max()
```

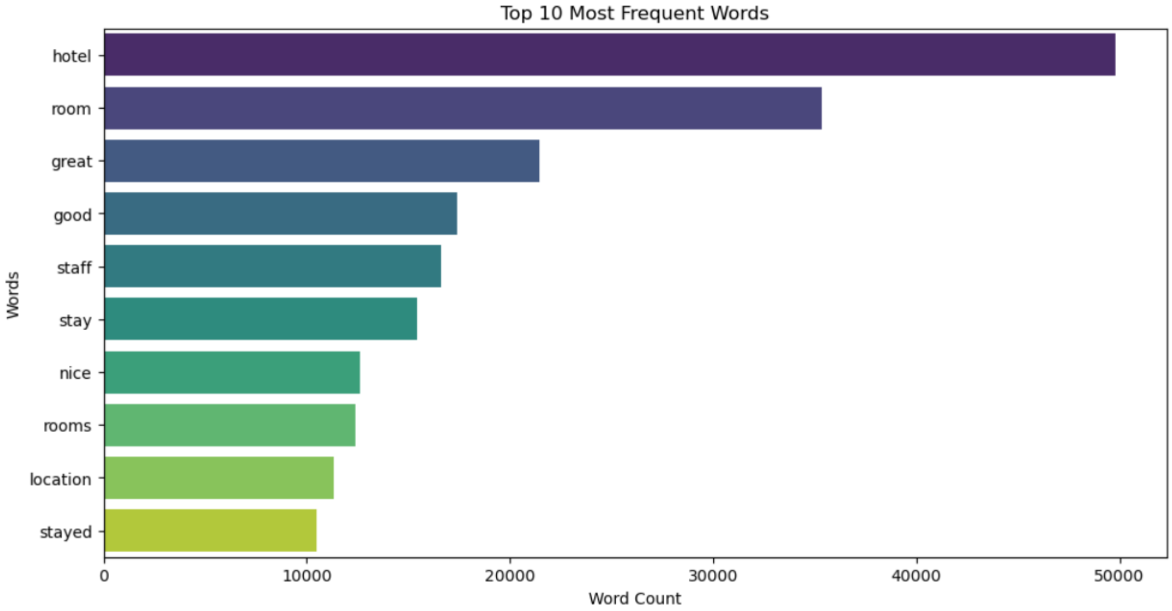
```
In [40]: sns.pairplot(hotel, vars=['Rating', 'Review_Length'], hue='Rating', palette='viridis')
         plt.suptitle('Pairplot of Ratings and Review Lengths', y=1.02)
         plt.show()
```

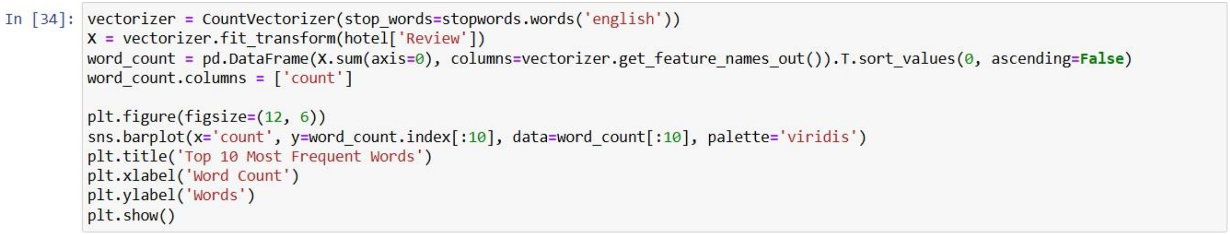


```
In [36]: hotel['Review_Length'] = hotel['Review'].apply(len)
         sns.histplot(hotel['Review_Length'], bins=30, kde=True)
         plt.title('Distribution of Review Lengths')
         plt.xlabel('Review Length')
         plt.ylabel('Frequency')
         plt.show()
```

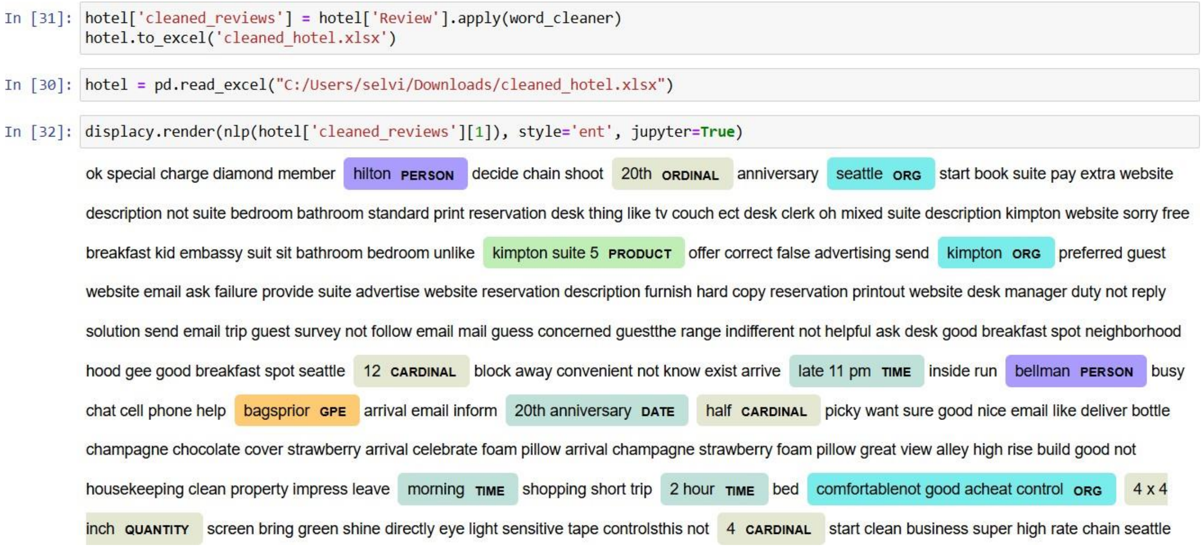


```
In [35]: sns.countplot(x='Rating', data=hotel)
         plt.title('Distribution of Ratings')
         plt.show()
```





```
In [34]: vectorizer = CountVectorizer(stop_words=stopwords.words('english'))
         X = vectorizer.fit_transform(hotel['Review'])
         word_count = pd.DataFrame(X.sum(axis=0), columns=vectorizer.get_feature_names_out()).T.sort_values(0, ascending=False)
         word_count.columns = ['count']

         plt.figure(figsize=(12, 6))
         sns.barplot(x='count', y=word_count.index[:10], data=word_count[:10], palette='viridis')
         plt.title('Top 10 Most Frequent Words')
         plt.xlabel('Word Count')
         plt.ylabel('Words')
         plt.show()
```

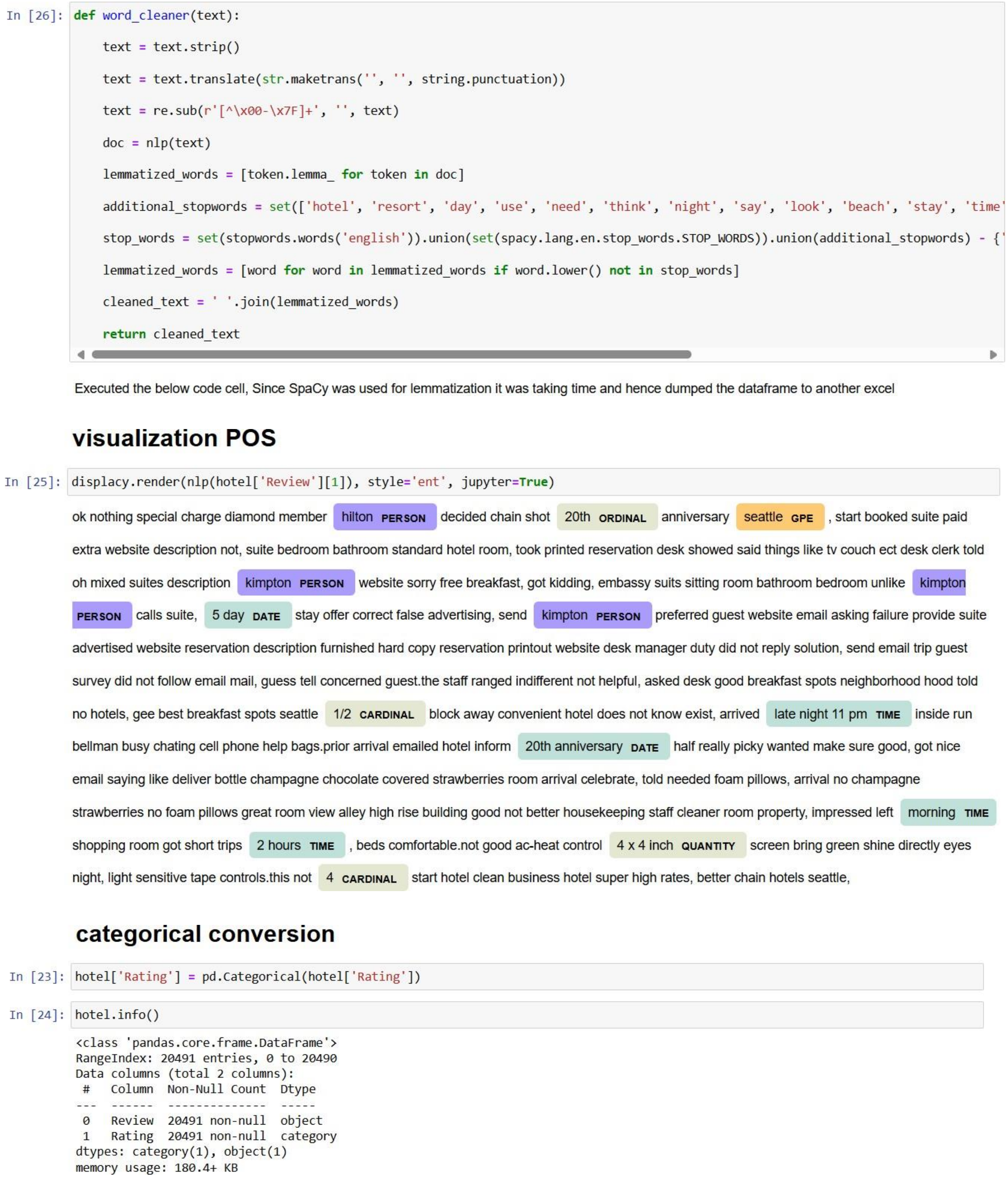


```
In [31]: hotel['cleaned_reviews'] = hotel['Review'].apply(word_cleaner)
         hotel.to_excel('cleaned_hotel.xlsx')
```

```
In [30]: hotel = pd.read_excel("C:/Users/selvi/Downloads/cleaned_hotel.xlsx")
```

```
In [32]: displacy.render(nlp(hotel['cleaned_reviews'][1]), style='ent', jupyter=True)
```

```
ok special charge diamond member hilton PERSON decide chain shoot 20th ORDINAL anniversary seattle ORG start book suite pay extra website
description not suite bedroom bathroom standard print reservation desk thing like tv couch ect desk clerk oh mixed suite description kimpton website sorry free
breakfast kid embassy suit sit bathroom bedroom unlike kimpton suite 5 PRODUCT offer correct false advertising send kimpton ORG preferred guest
website email ask failure provide suite advertise website reservation description furnish hard copy reservation printout website desk manager duty not reply
solution send email trip guest survey not follow email mail guess concerned guestthe range indifferent not helpful ask desk good breakfast spot neighborhood
hood gee good breakfast spot seattle 12 CARDINAL block away convenient not know exist arrive late 11 pm TIME inside run bellman PERSON busy
chat cell phone help bagsprior GPE arrival email inform 20th anniversary DATE half CARDINAL picky want sure good nice email like deliver bottle
champagne chocolate cover strawberry arrival celebrate foam pillow arrival champagne strawberry foam pillow great view alley high rise build good not
housekeeping clean property impress leave morning TIME shopping short trip 2 hour TIME bed comfortablenot good acheat control ORG 4 x 4
inch QUANTITY screen bring green shine directly eye light sensitive tape controlsthis not 4 CARDINAL start clean business super high rate chain seattle
```



In [26]:
```
def word_cleaner(text):

    text = text.strip()

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

    text = re.sub(r'[^\x00-\x7F]+', '', text)

    doc = nlp(text)

    lemmatized_words = [token.lemma_ for token in doc]

    additional_stopwords = set(['hotel', 'resort', 'day', 'use', 'need', 'think', 'night', 'say', 'look', 'beach', 'stay', 'time'

    stop_words = set(stopwords.words('english')).union(set(spacy.lang.en.stop_words.STOP_WORDS)).union(additional_stopwords) - {'

    lemmatized_words = [word for word in lemmatized_words if word.lower() not in stop_words]

    cleaned_text = ' '.join(lemmatized_words)

    return cleaned_text
```

Executed the below code cell, Since SpaCy was used for lemmatization it was taking time and hence dumped the dataframe to another excel

visualization POS

In [25]:
```
displacy.render(nlp(hotel['Review'][1]), style='ent', jupyter=True)
```

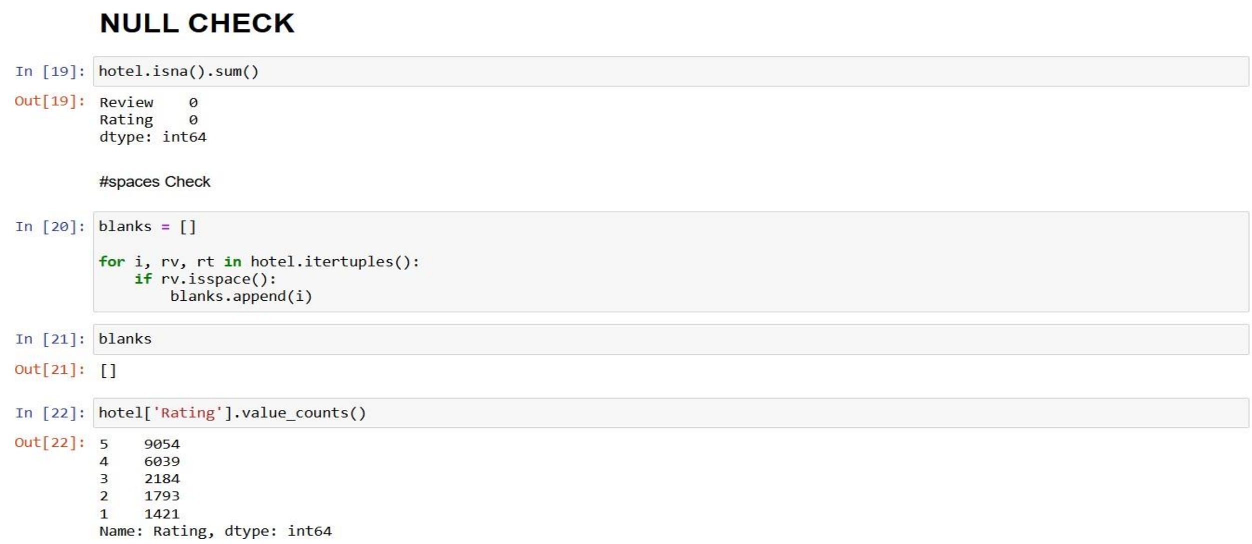
ok nothing special charge diamond member hilton PERSON decided chain shot 20th ORDINAL anniversary seattle GPE , start booked suite paid extra website description not, suite bedroom bathroom standard hotel room, took printed reservation desk showed said things like tv couch ect desk clerk told oh mixed suites description kimpton PERSON website sorry free breakfast, got kidding, embassy suits sitting room bathroom bedroom unlike kimpton PERSON calls suite, 5 day DATE stay offer correct false advertising, send kimpton PERSON preferred guest website email asking failure provide suite advertised website reservation description furnished hard copy reservation printout website desk manager duty did not reply solution, send email trip guest survey did not follow email mail, guess tell concerned guest.the staff ranged indifferent not helpful, asked desk good breakfast spots neighborhood hood told no hotels, gee best breakfast spots seattle 1/2 CARDINAL block away convenient hotel does not know exist, arrived late night 11 pm TIME inside run bellman busy chating cell phone help bags.prior arrival emailed hotel inform 20th anniversary DATE half really picky wanted make sure good, got nice email saying like deliver bottle champagne chocolate covered strawberries room arrival celebrate, told needed foam pillows, arrival no champagne strawberries no foam pillows great room view alley high rise building good not better housekeeping staff cleaner room property, impressed left morning TIME shopping room got short trips 2 hours TIME , beds comfortable.not good ac-heat control 4 x 4 inch QUANTITY screen bring green shine directly eyes night, light sensitive tape controls.this not 4 CARDINAL start hotel clean business hotel super high rates, better chain hotels seattle,

categorical conversion

In [23]:
```
hotel['Rating'] = pd.Categorical(hotel['Rating'])
```

In [24]:
```
hotel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20491 entries, 0 to 20490
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype   
---  ------  --------------  -----   
 0   Review  20491 non-null  object  
 1   Rating  20491 non-null  category
dtypes: category(1), object(1)
memory usage: 180.4+ KB
```

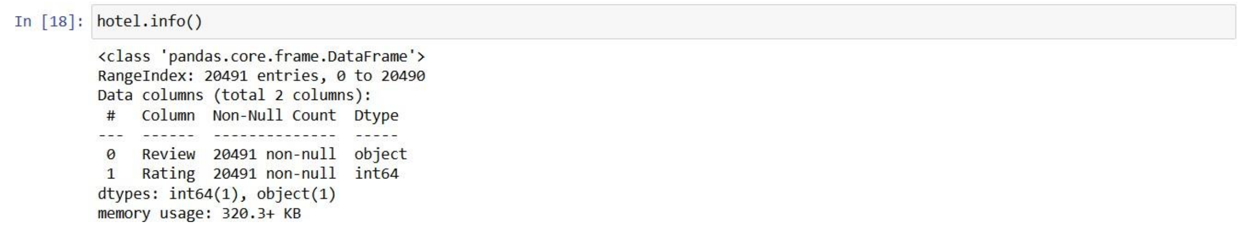


NULL CHECK

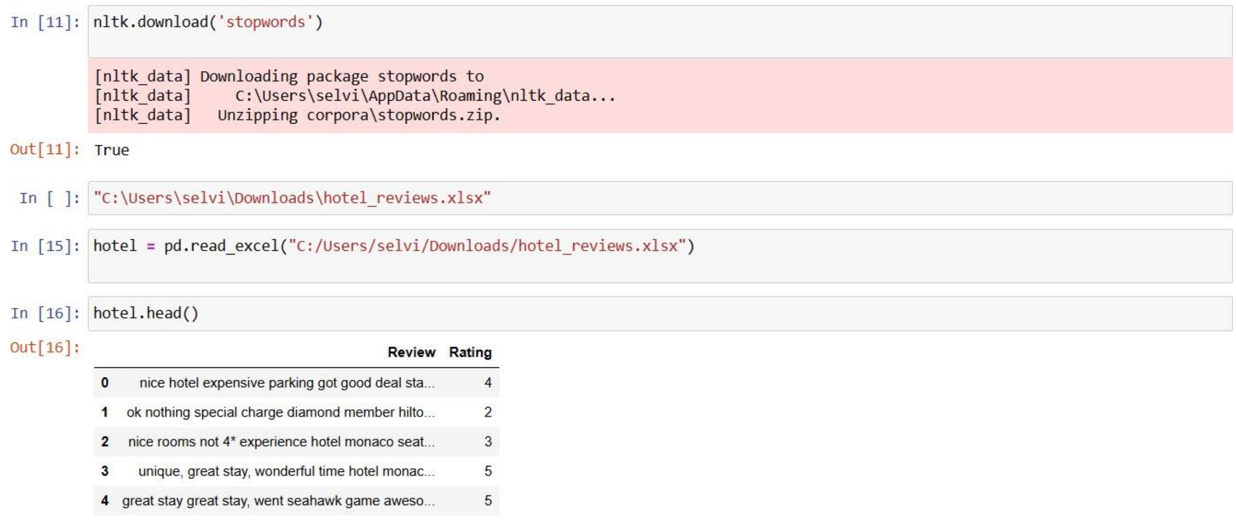
```
In [19]: hotel.isna().sum()

Out[19]: Review    0
         Rating    0
         dtype: int64
```

#spaces Check

```
In [20]: blanks = []

         for i, rv, rt in hotel.itertuples():
             if rv.isspace():
                 blanks.append(i)

In [21]: blanks

Out[21]: []

In [22]: hotel['Rating'].value_counts()

Out[22]: 5    9054
         4    6039
         3    2184
         2    1793
         1    1421
         Name: Rating, dtype: int64
```

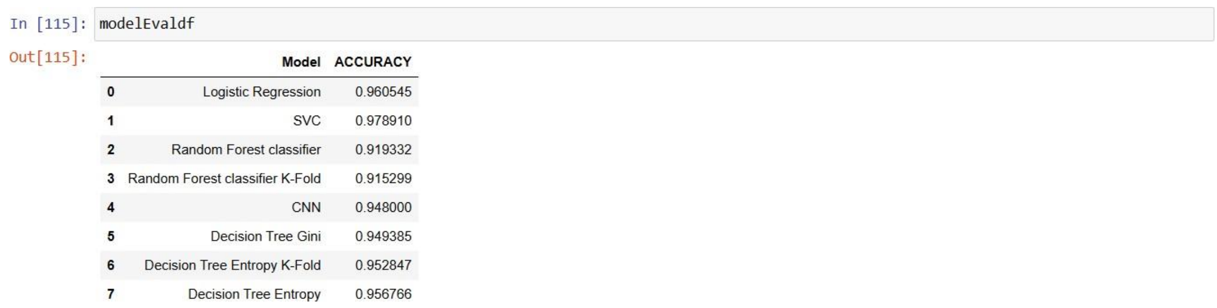


```
In [18]: hotel.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 20491 entries, 0 to 20490
         Data columns (total 2 columns):
          #   Column  Non-Null Count  Dtype
         ---  ------  --------------  -----
          0   Review  20491 non-null  object
          1   Rating  20491 non-null  int64
         dtypes: int64(1), object(1)
         memory usage: 320.3+ KB
```



```
In [11]: nltk.download('stopwords')

         [nltk_data] Downloading package stopwords to
         [nltk_data]     C:\Users\selvi\AppData\Roaming\nltk_data...
         [nltk_data]   Unzipping corpora\stopwords.zip.

Out[11]: True

In [ ]: "C:\Users\selvi\Downloads\hotel_reviews.xlsx"

In [15]: hotel = pd.read_excel("C:/Users/selvi/Downloads/hotel_reviews.xlsx")

In [16]: hotel.head()

Out[16]:
```

	Review	Rating
0	nice hotel expensive parking got good deal sta...	4
1	ok nothing special charge diamond member hilto...	2
2	nice rooms not 4* experience hotel monaco seat...	3
3	unique, great stay, wonderful time hotel monac...	5
4	great stay great stay, went seahawk game aweso...	5



```
In [115]: modelEvaldf

Out[115]:
```

	Model	ACCURACY
0	Logistic Regression	0.960545
1	SVC	0.978910
2	Random Forest classifier	0.919332
3	Random Forest classifier K-Fold	0.915299
4	CNN	0.948000
5	Decision Tree Gini	0.949385
6	Decision Tree Entropy K-Fold	0.952847
7	Decision Tree Entropy	0.956766

9. Bibliography and References

9.1 Online Reference

[1] Natural Language Processing with Python:

- [NLTK Book](http://www.nltk.org/book/) - A comprehensive guide to using the Natural Language Toolkit (NLTK) for Python.

[2] SpaCy Documentation:

- [SpaCy Documentation](https://spacy.io/usage) - Official documentation for SpaCy, an NLP library in Python.

[3] Scikit-learn Documentation:

- [Scikit-learn Documentation](https://scikit-learn.org/stable/user_guide.html) - Comprehensive guide for machine learning in Python, including sentiment analysis techniques.

[4] TensorFlow Documentation:

- [TensorFlow Documentation](https://www.tensorflow.org/tutorials) - Tutorials and guides on using TensorFlow for machine learning tasks.

[5] Keras Documentation:

- [Keras Documentation](https://keras.io/guides/) - Official documentation for Keras, a high-level neural networks API.

[6] WordCloud Documentation:

- [WordCloud GitHub](https://github.com/amueller/word_cloud) - Information on creating word clouds in Python.

[7] Sentiment Analysis with Python:

- [A Comprehensive Guide to Sentiment Analysis](https://towardsdatascience.com/sentiment-analysis-in-python-using-textblob-and-vader-15a364b55f2d) - A guide on performing sentiment analysis using TextBlob and VADER.

[8] Machine Learning for Text Data:

- [Machine Learning with Text Data](https://www.analyticsvidhya.com/blog/2021/01/introduction-to-natural-language-processing-nlp-in-machine-learning/) - An introduction to NLP in machine learning.

[9] Building a Recommendation System with Reviews:

- [Building a Recommendation System](https://www.datacamp.com/community/tutorials/recommender-systems-python) - Insights into how to use reviews to build a recommendation system.

[10] Data Visualization in Python: https://matplotlib.org/stable/index.html.

9.2 Additional Resources:

[11] Data Preprocessing in Machine Learning:.

- [Data Preprocessing Techniques in Machine Learning](https://towardsdatascience.com/data-preprocessing-techniques-in-machine-learning-with-python-30b4cdbf55b5) - Overview of various preprocessing techniques useful for NLP and ML.

[12] Handling Imbalanced Data:

- [Dealing with Imbalanced Classes in Machine Learning](https://www.kaggle.com/code/uciml/handling-imbalanced-data/notebook) - Techniques for addressing imbalanced datasets in classification tasks.

These links should provide a solid foundation for understanding the technologies and methodologies used in your hotel reviews analysis project.