

HOTEL RECOMMENDATION SYSTEM WITH MACHINE LEARNING

BY

PANG CHI CHONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS

ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2023

REPORT STATUS DECLARATION FORM

Title: Hotel Recommendation System with Machine Learning

Academic Session: Y3S3

I PANG CHI CHONG

(CAPITAL LETTER)

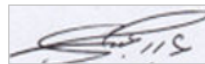
declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)



(Supervisor's signature)

Address:

No.2, Taman Pengkalan Aor

Lorong Utama 5,34000, ___

Taiping,Perak

Abdulkarim Kanaan Jebna

Supervisor's name

Date: 15/09/2023

Date: 15/09/2023

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY OF INFORMATION SYSTEM AND COMMUNICATION

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 15/09/2023

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Pang Chi Chong (ID No: 19ACB03734) has completed this final year project/ dissertation/ thesis* entitled “Hotel Recommendation System” under the supervision of Dr. Abdulkarim Kanaan Jebna (Supervisor) from the Department of Information Systems, Faculty of Information and Communication Technology, and Syed Muhammad Bin Syed Omar (Co-Supervisor)* from the Department of Information System, Faculty of Information Systems

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,




(Pang Chi Chong)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**METHODOLOGY, CONCEPT AND DESIGN OF HOTEL RECOMMENDATION SYSEM WITH MACHINE LEARNING**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____


Name : _____Pang Chi Chong_____

Date : _____15/9/2023_____

ACKNOWLEDGEMENTS

I would like to thank my FYP supervisor, Dr. Abdulkarim Kanaan Jebna, for his crucial advice and support during the course of this research. Whenever I had an issue, he was always willing to provide his experience and patience to help me determine the core cause and find a solution. Dr. Abdulkarim Kanaan Jebna is not only my FYP supervisor, but also my Data Mining instructor. I gained a lot from his classes, and it was via his introduction to machine learning that I discovered my interest in this topic. He is the most responsible instructor I've ever met, and his expectations for me have inspired me to work more and achieve more. I appreciate his patience, wisdom, and support, and I will do everything in my power to meet his expectations.

ABSTRACT

The hospitality industry has witnessed a notable increase in the utilisation of online booking platforms and the reliance on online reviews in recent years. This phenomenon presents a challenge for customers in their search for a suitable hotel that aligns with their specific requirements. Machine learning techniques were employed to develop a hotel recommendation system as a means of addressing this issue. The objective of this report is to elucidate the development process and evaluate the efficacy of the aforementioned system. The CRISP-DM approach was employed in conducting the present investigation. The system underwent training using a dataset scraped from Web using beautifulsoup webscraping , consisting of hotel data that had been preprocessed and transformed into a format suitable for utilisation by machine learning models. The system employs a database including hotel attributes and customer evaluations in order to compute the cosine similarity between the characteristics of each hotel and the user's preferences. The TF-IDF technique is employed to assign weight to each word in a review, taking into account its frequency across the entire database. By integrating these two methodologies, the system is capable of delivering tailored recommendations to users, taking into account their individual tastes. The study's findings indicate that the implementation of a machine learning-based hotel recommendation system has the potential to provide customers with valuable suggestions, hence enhancing their hotel booking experience. This study holds significance as it contributes to the field of hospitality by providing a practical resolution to the issue of hotel suggestion and addressing the existing knowledge gap about the utilisation of machine learning for enhancing hotel recommendations.

TABLE OF CONTENTS

TITLE PAGE	i
REPORT STATUS DECLARATION FORM	ii
FYP THESIS SUBMISSION FORM	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xi
LIST OF SYMBOLS	xii
LIST OF ABBREVIATIONS	xiii
CHAPTER 1 INTRODUCTION	14
1.1 Problem Statement and Motivation	15
1.2 Objectives	16
1.3 Project Scope and Direction	17
1.3.1 Research Emphasis	17
1.3.2 Target Audience of the project	17
1.3.3 Advantages for the audience for individuals engaged in travel	17
1.3.4 Components of the project	18
1.4 Contributions	19
1.5 Report Organization	20

CHAPTER 2 LITERATURE REVIEW	21
2.1 Studies of Algorithms on Machine Learning	21
2.1.1 Cosine Similiarity	21
2.1.2 TF - IDF	23
2.1.3 Simple Imputation	24
2.1.4 Flask	25
2.1.5 R Studio	27
2.1.6 Beautiful Soup	28
2.1.7 Jaccard Similiarity	30
2.2 Previous Work on Hotel Recommendations	32
2.2.1 Using Online Hotel Customer Reviews to Improve the Booking Process	32
2.2.2 Recommendation System based on Data Mining	34
2.2.3 Trivago	36
CHAPTER 3 SYSTEM METHODOLOGY	39
	39
3.1 CRISP – DM	39
3.1.1 Business Understading	40
3.1.2 Data Understanding	40
3.1.3 Data Preprocessing	41
3.1.4 Modeling	41
3.1.5 Evaluation	42
3.1.6 Implementation and Deployment	42
3.2 Hardware	43
CHAPTER 4 SYSTEM DESIGN	44
4.1 System Block Diagram	44
4.1.1 Overall System Architecture	44
4.2 Performance and Scalability	45

CHAPTER 5 SYSTEM IMPLEMENTATION	46
5.1 Setting Up	46
5.1.1 Software	46
5.2 CRISP-DM	47
5.2.1 Business Understanding	47
5.2.2 Data Understanding	50
5.2.3 Data Preparation	54
5.2.4 Modeling	59
5.2.5 Evaluation	61
5.2.6 Deployment	62
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION	67
6.1 Survey Question Evaluation	67
CHAPTER 7 CONCLUSION AND RECOMMENDATION	71
7.0 Conclusion & Recommendation	71
REFERENCES	74
APPENDIX	75
WEEKLY LOG	78
POSTER	83
PLAGIARISM CHECK RESULT	84
FYP2 CHECKLIST	87

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	Cosine Similarity example	21
Figure 2.0	TF – IDF Calculations	23
Figure 2.1	Web Scrabing Snapshots	29
Figure 3.0	Trivago	36
Figure 4.0	Crisp – DM	39
Figure 5.0	System Block Diagram	44
Figure 6.0	Recommendation Functions	48
Figure 6.1	Dataset	51
Figure 6.2	Data Visualizing Bar Chart	52
Figure 6.3	Data Visualizing Bar Chart	52
Figure 6.4	Data Cleaning Python Scripts	54
Figure 6.5	Data type Checking Python Scripts	56
Figure 6.6	Data type Checking Python Scripts	56
Figure 6.7	Simple Imputation Python Scripts	57
Figure 6.8	Cosine/Jaccard Similiarity Recommendation Functions	60
Figure 6.8	Evaluation with Precision / Recall	61
Figure 6.9	Deployment Functions	61
Figure 6.10	System Interface	63
Figure 6.11	System Output	64
Figure 7.0	Survery Question 1	67
Figure 7.1	Survery Question 2	68
Figure 7.2	Survery Question 3	68
Figure 7.3	Survery Question 4	69

LIST OF TABLES

Table Number	Title	Page
Table 1.0	Data Quality Report	50
Table 2.0	Data Quality Issues	50

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

<i>CRISP-DM</i>	Cross-Industry Standard Process for Data Mining
<i>TF-IDF</i>	Term Frequency Inverse Document Frequency

Chapter 1

Introduction

The hospitality sector is characterised by intense competition, prompting hotels to continuously explore innovative approaches in order to augment customer satisfaction and foster loyalty. Using hotel recommendation systems that provide personalised suggestions. These systems utilise machine learning algorithms that undergo training on extensive datasets of user behaviour, enabling the system to offer personalised recommendations for individual customers.

Finding the ideal hotel that meets one's specific preferences and needs has become increasingly difficult in the digital age of travel. The plethora of options available on online booking platforms frequently causes travellers to experience decision fatigue. In this context, the deployment of machine learning techniques is no longer merely pertinent, but essential. This project introduces a Hotel Recommendation System that employs advanced machine learning algorithms, such as cosine similarity, content-based filtering, and TF-IDF (Term Frequency-Inverse Document Frequency), to alleviate the difficulties associated with hotel selection, thereby empowering travellers with personalised recommendations that reflect their unique preferences.

REFERENCES

1.1 Problem Statement and Rationale

This section will provide a discussion on the problem statement and motivation of the study.

The motivation for this undertaking stems from the need to enhance the travel experience for individuals seeking lodging options. The dominant method for choosing hotels in conventional practises predominantly relies on general star ratings or user evaluations, which often prove insufficient in catering to individual preferences. The aim of this project is to develop a hotel recommendation system that exceeds conventional methodologies by harnessing the potential of machine learning techniques.

In order to provide a more comprehensive analysis of the issue at hand, a meticulous evaluation of the prevailing systems within the sector has been undertaken. The findings of this investigation indicate that existing methods for hotel selection exhibit a deficiency in their capacity to thoroughly account for the specific preferences of travelers. Contemporary travelers exhibit a wide array of requirements, encompassing distinct facilities and geographical preferences, as well as individualized trip objectives. Hence, the uniform methodology employed by current systems may lead to inefficient decision-making and less than ideal travel experiences.

The primary impetus behind our efforts is to overcome these constraints and offer travelers a more personalized and gratifying procedure for choosing hotels. Through the utilization of machine learning techniques, our objective is to develop a system that possesses a comprehensive understanding of consumers at an individual level, encompassing their unique preferences, specific needs, and previous choices. The use of this method aims to mitigate the challenges associated with the abundance of hotel options and provide consumers with tailored recommendations that cater to their specific requirements.

REFERENCES

1.2 Objectives

The project's objectives have been formulated with a focus on quantifiable results, so assuring precision and congruence with the expectations of stakeholders. Our primary focus is to attain the following objectives:

The objective of this project is to develop a system that utilizes user input and historical data to provide tailored hotel suggestions. The main indicator of achievement for this objective will be the customer satisfaction score, which will be obtained by post-travel questionnaires. Success will be defined as achieving a score of 90% or above.

The integration of machine learning approaches, such as cosine similarity, content-based filtering, and TF-IDF, is proposed in order to improve the precision of suggestions. A quantifiable objective has been established with the aim of attaining a minimum enhancement of 10% in the accuracy of recommendations as compared to conventional methods.

Preference Refinement for Users. Enable users to input their preferences and constraints, including factors such as geographical location, financial limitations, and desired amenities, in order to further enhance the accuracy and relevance of the recommendations provided. The criterion for success will be determined by a quantifiable metric, specifically a 20% augmentation in the quantity of consumers who effectively locate a hotel that aligns with their individualized requirements.

Evaluation of System Performance. The performance of the recommendation system will be assessed through comprehensive testing. The performance criteria to be utilized in this study encompass precision, recall, and F1-score. The primary objective is to get a minimum accuracy of 85% across all aforementioned metrics.

The objective of this study is to optimize the user interface for tourists, with the aim of simplifying the process of selecting hotels and improving efficiency and user satisfaction. The assessment of achievement in this particular goal will be conducted using user feedback surveys, aiming to attain a user satisfaction score of 80% or above.

1.3 The Scope and Direction of the Project

1.3.1 Research Emphasis

The primary objective of this project is to facilitate the advancement of a sophisticated hotel recommendation system by harnessing the capabilities of machine learning and natural language processing methodologies. The primary aim of this study is to examine user preferences and hotel attributes in order to provide tailored recommendations. In order to ensure that the recommendations closely fit with the interests and requirements of users, several key methods will be utilized, such as cosine similarity, content-based filtering, and TF-IDF.

1.3.2 Target Audience of the Project

The main target demographic for this project comprises the hotel suggestion system that have developed is designed to cater to the requirements and tastes of individual tourists in search of suitable lodging options.

The hospitality industry can derive valuable insights from the system, which can be utilized by hotel owners and operators to improve their services and effectively cater to client demands.

1.3.3 Advantages for the Audience For Individuals Engaged in Travel

The major objective of the system is to optimize and improve the travel experience for individual travelers. Our objective is to offer individualized hotel suggestions.

The process of decision-making can be simplified for travelers who are frequently confronted with a wide array of options when it comes to selecting accommodations. The technology that have developed facilitates this process by providing customized choices that closely align with the individual's preferences, thereby optimizing the decision-making process.

Enhancing happiness. The implementation of personalized recommendations has been shown to enhance the probability of passengers selecting hotels that align with their individual preferences and requirements. Consequently, this leads to a heightened level of happiness with their overall travel experiences.

REFERENCES

The system's generated insights can provide several advantages for hotel owners and operators within the Hospitality Industry.

Enhanced service customization allows hotels to optimize their offerings and services in order to align with the specific tastes and requirements of their visitors, resulting in heightened levels of guest satisfaction and loyalty. The utilization of data-driven decision-making is crucial in informing strategic decisions, including but not limited to marketing tactics, pricing adjustments, and facility enhancements, by providing access to data pertaining to user preferences and trends.

1.3.4 Components of the Project

The project's scope comprises the following components. The process of data acquisition and preprocessing involves the collection and refinement of hotel data, user profiles, and reviews from several sources. User Profiling. The process of constructing user profiles by analyzing their preferences, historical booking data, and feedback. Recommendation Generation. Leveraging machine learning techniques for the purpose of generating individualized hotel suggestions.

1.4 Significant Contributions

The acquisition of consumers in the hospitality and tourism sectors is heavily dependent on online reviews. Customers may encounter difficulties and spend a significant amount of time in their search for the optimal hotel, especially when faced with a substantial volume of reviews. Consequently, the primary aim of this project is to design and implement a hotel recommendation system that effectively utilises trustworthy review data to assist consumers in efficiently identifying their desired hotel.

In order to accomplish this objective, the system will utilise machine learning methodologies, including TF-IDF and cosine similarity, to narrow down assessments that satisfy predetermined criteria. These criteria encompass a minimum word count requirement and adherence to a designated time period for submission. This feature allows consumers to optimise their time and concentrate on reviews that are most pertinent to their specific requirements.

The scope of the project encompasses the creation of a web application that enables users to search for hotels according to their preferences and specific criteria. The proposed study aims to utilise a comprehensive dataset of hotel evaluations to conduct training and evaluation of machine learning algorithms. The output generated by the system will consist of personalised hotel recommendations that align with the specific criteria provided by the user.

The proposed hotel recommendation system aims to leverage dependable review data in order to furnish consumers with precise and tailored hotel recommendations. Customers have the ability to locate their desired hotel more efficiently, resulting in time savings. Simultaneously, hotels experience advantages such as enhanced customer acquisition and retention rates.

1.5 Organisational Structure of the Report

The subsequent chapters of this report provide a comprehensive overview of the specific details. Chapter 2 provides a comprehensive review of relevant articles and prior research pertaining to recommendation systems. Chapter 3 presents a proposed method, along with the inclusion of the software and hardware system. Chapter 4 presents a System Block Diagram that elucidates the operational functions of the system. Chapter 5 of the aforementioned text elucidates the subsequent actions and intricacies of the Cross-Industry Standard Process for Data Mining, as expounded upon in chapter 3. Chapter 6 presents a methodology for assessing the system's performance through the development of user survey questions. Additionally, Chapter 7 serves as a conclusion and summary of the content covered in this report.

Chapter 2

Literature Review

2.1 Studies of Algorithms on Machine Learning

2.1.1 Cosine Similarity

Initially in an inner product space, two vectors that are not zero may be compared using cosine similarity as a starting point. The cosine of the angle formed by the two vectors is measured. It is often used in data science and machine learning to compare how similar two texts or sets of attributes are to one another.

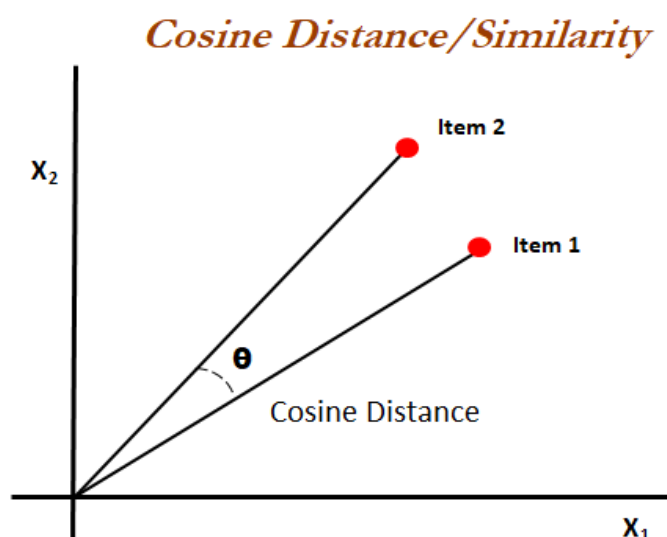


Figure 1.0 Cosine Similarity example

By measuring the cosine of the angle between the term frequency-inverse document frequency (TF-IDF) vectors of two documents, cosine similarity may be used to determine how similar two documents are in the context of text data. The cosine similarity score that results ranges from -1 to 1. A score of 1 denotes identity, a score of 0 indicates orthogonality, and a score of -1 denotes full differences between the two vectors.

In a variety of domains, including natural language processing, information retrieval, computer vision, and bioinformatics, cosine similarity is a well-liked similarity measure. In a high-dimensional space, like a document or a word embedding, it compares two vectors. To determine cosine similarity, the two vectors are represented as arrays of numerical values. The magnitude product of the two vectors is then used to

REFERENCES

divide the dot product of these arrays. This results in a value between -1 and 1, with 1 indicating that the two vectors are identical and -1 denoting that they are wholly different.

According to [3], cosine similarity is very useful in natural language processing applications like document grouping and classification. To find similar items based on their traits or attributes, recommender systems also utilise it [4]. In machine learning techniques, cosine similarity is often used, especially in natural language processing (NLP) applications. It is used in sentiment analysis to measure the similarity of word and phrase vectors as well as to judge how similar different reviews or tweets are [5]. Additionally, it is used in language modelling to evaluate the similarities between various contexts and predict the subsequent word in a sequence [6]. In computer vision, cosine similarity is used to contrast feature vectors extracted from images, such as HOG or deep neural network features. This is often used in applications involving image retrieval, object recognition, and image categorization [7].

Cosine similarity is a powerful method for comparing the similarity of high-dimensional vectors, and it has many real-world uses in a wide range of fields.

REFERENCES

2.1.2 TF - IDF

The phrase frequency-inverse document frequency (TF-IDF) weighting technique is often used in information retrieval and text mining. It uses statistics to evaluate a word's importance inside a text or corpus. The foundation of the TF-IDF algorithm is the notion that a phrase that appears often in a document or corpus is significant—but only if it does not appear frequently in all of the texts in the corpus. In other words, a phrase is considered less important if it appears often both inside a text and throughout the whole corpus.

There are two main steps to the TF-IDF approach. Calculate the term frequency (TF) first. The "frequency" of a term is how often it appears in a text. $TF(t, d) = (\text{number of words in document } d / \text{number of times the term } t \text{ appears in document } d)$ The TF of the word "cat" in that document is 0.1, for instance, if it occurs 10 times in a text with a total of 100 words.

After that, the algorithm determines a word's inverse document frequency (IDF), which is a gauge of how much information the word provides throughout the corpus. It may be calculated by taking the logarithm of the number obtained by dividing the total number of documents in the corpus by the number of documents that include the phrase. $IDF(t)$ is equal to $\log_e (\text{Total documents} / \text{documents containing term } t)$. For instance, the IDF of "cat" is $\log(1000/100) = 2$ if a corpus has 1,000 documents and the word "cat" appears in 100 of them.

TF-IDF

TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a doc with the number of docs the word appears in.

$$TF-IDF = TF(t, d) \times IDF(t)$$

Annotations for the equation:

- $TF(t, d)$: Term frequency (Number of times term t appears in a doc, d)
- $IDF(t)$: Inverse document frequency

$$\log \frac{1 + n}{1 + df(d, t)}$$

Annotations for the IDF formula:

- n : # of documents
- $df(d, t)$: Document frequency of the term t

Figure 2.0 TF – IDF Calculations

In order to get the TF-IDF value for a word in a document, lastly multiply the TF value by the IDF value. The equation reads as follows: $TF-IDF(t, d) = IDF(t) * TF(t, d)$ The

REFERENCES

resulting TF-IDF score is a gauge of a word's value in the corpus in relation to its relevance in a text. In a document, words with high TF-IDF scores are seen as crucial, while words with low TF-IDF scores are perceived as less crucial.

2.1.3 Simple Imputation

The presence of missing data is an often encountered challenge in the field of data analysis [1]. Researchers frequently employ imputation techniques when confronted with incomplete datasets in order to estimate missing values and uphold the integrity of their findings. This article explores the domain of simple imputation techniques and their significance in the context of data analysis. Simple imputation procedures refer to uncomplicated techniques utilised for the purpose of replacing missing data items. These algorithms exhibit high computing efficiency and can be readily implemented. Several often employed simple imputation strategies encompass mean imputation [2], median imputation [8], and mode imputation [9]. The aforementioned methods are utilised to substitute missing values with the mean, median, or mode of the existing data, correspondingly.

Simple imputation approaches offer numerous advantages. These algorithms have high computational efficiency, rendering them well-suited for the analysis of extensive datasets. Furthermore, these methods are characterised by their accessibility and feasibility, making them suitable for individuals lacking expertise in statistical analysis. Nevertheless, it is important to acknowledge that they also possess some constraints. For example, the inclusion of missing data in the dataset may induce bias, particularly when the missingness is not totally random (MCAR) [10]. Furthermore, the aforementioned approach fails to consider the inherent uncertainty linked to the imputed values. Simple imputation approaches are utilised in several sectors. Mean imputation is a commonly employed technique in healthcare research to estimate missing patient data in the context of clinical trials [11]. Mode imputation is a technique commonly employed in the field of finance to address the issue of missing stock price data, with the aim of facilitating analytical endeavours [12]. The aforementioned examples serve to demonstrate the adaptability of basic imputation techniques. In summary, basic imputation techniques are of paramount importance in the field of data analysis since they offer expedient and pragmatic approaches for addressing missing data. Although there are certain benefits associated with them, it is crucial to carefully

REFERENCES

evaluate their limits and the underlying assumptions they rely on when considering their usage in data analysis. It is imperative for researchers to exercise caution and deliberate thought when selecting the most appropriate imputation method that aligns with the unique characteristics of their dataset and the objectives of their research.

2.1.4 Flask

The field of web development holds significant importance in contemporary software engineering, with Python emerging as a favoured programming language for the creation of web applications. Flask, a micro web framework, is renowned for its sophisticated and minimalistic nature, making it a highly favourable choice for constructing online applications. This essay explores the realm of Flask and its possibilities for streamlining web development tasks. Flask adheres to the architectural pattern known as Model-View-Controller (MVC), while also granting developers the freedom to select their preferred architectural approach [13]. The essence of Flask is characterised by its minimalistic nature, although it provides the opportunity for expansion through the utilisation of third-party extensions [14]. The inherent simplicity of this particular attribute serves as a notable advantage, rendering it a very suitable option for projects of modest to moderate scale. According to the source, Flask provides fundamental functionalities for web development, such as routing, templating, and request processing. The routing architecture of the framework enables developers to specify URL patterns and link them with corresponding view functions, facilitating the dynamic generation of web pages. The Jinja2 template engine, which is the default engine, facilitates the process of producing dynamic material within HTML templates. In addition, Flask offers inherent functionality for managing HTTP requests and responses, hence streamlining the process of developing RESTful APIs [15]. The merits of Flask are attributed to its simplicity and minimalistic design. This feature enables developers to concentrate on the particular project requirements without enforcing an inflexible framework. In addition, the comprehensive array of extensions available in Flask empowers developers to incorporate desired functionalities, hence guaranteeing the capacity to handle increased demands and adaptability to changing requirements. The lightweight characteristic of this technology leads to accelerated application startup times and reduced resource utilisation, rendering it well-suited for microservices and containerized applications. The Flask framework has gained significant popularity and

REFERENCES

acceptance across multiple areas, encompassing e-commerce, social networking, and content management systems [16]. The exceptional suitability of this technology for quick prototyping and web application development is attributed to its versatility and user-friendly nature. The usability of the software is further enhanced by its interoperability with widely used databases such as SQLite, MySQL, and PostgreSQL [17]. Flask, being a Python web framework that is characterised by its lightweight nature, presents a pragmatic and effective approach to the building of web applications. The software's inherent simplicity, adaptability, and extensibility render it highly suitable for a diverse array of projects. Although Flask may not be the optimal selection for all situations, its merits become evident in projects that prioritise swift development, minimalism, and the ability to select components that align with specific requirements.

2.1.5 R Studio

RStudio has emerged as a robust Integrated Development Environment (IDE) designed to meet the requirements of data scientists, statisticians, and analysts that utilise the R programming language. This article offers a comprehensive analysis of RStudio, including its fundamental characteristics and its significant contribution to improving productivity in R programming. Additionally, the text explores the manner in which RStudio facilitates a smooth and uninterrupted process from the importation of data to the creation of visual representations, rendering it an essential instrument for individuals utilising the R programming language. The programming language R, which is extensively utilised for statistical computation and data analysis [18], has experienced a substantial surge in popularity in recent times. RStudio, an open-source integrated development environment (IDE) created by RStudio, Inc., provides a full platform for the development of the R programming language, enhancing its functionalities [19]. This article examines the fundamental characteristics and capabilities of RStudio and its importance inside the R programming environment. RStudio is known for its interface that is designed to enhance the efficiency of R development by providing a user-friendly experience. The software package incorporates a script editor that offers features such as syntax highlighting and autocompletion, facilitating efficient coding practises. The environment is equipped with integrated graphics and visualisation capabilities that aid in the development of engaging data visualisations [20]. Furthermore, RStudio provides a versatile environment for project management, script organisation, and access to documentation, so enhancing the overall structure of the development process. One notable aspect of RStudio is its capacity to optimise and expedite the processes involved in data analysis operations. By incorporating comprehensive capabilities for the importation, manipulation, and visualisation of data, users are able to smoothly shift from the first study of data to the subsequent stages of model development and presentation. In addition, RStudio provides support for RMarkdown, enabling the generation of dynamic reports that seamlessly integrate code, analysis, and visualisations [21]. The aforementioned capacity serves to augment cooperation and repeatability within data-driven projects. In summary, RStudio is an essential tool for individuals utilising R, providing a comprehensive range of functionalities that augment efficiency and streamline the process of analysing data. The R programming language is highly

REFERENCES

regarded among data scientists, statisticians, and analysts because to its intuitive interface, integrated development tools, and support for reproducible research. These features contribute to its significant value as a tool in these fields.

2.1.6 Beautiful Soup

Web scraping is a key methodology employed to gather data from webpages, and BeautifulSoup emerges as a highly valuable Python module specifically designed for this purpose. This article presents a comprehensive examination of BeautifulSoup, emphasising its proficiency in the analysis and traversal of HTML and XML documents, as well as its practical utility in the realm of data acquisition. In addition, engaging in an examination of the ethical implications associated with web scraping and offer perspectives on the integration of BeautifulSoup with other libraries to facilitate comprehensive online scraping endeavours. The practise of online scraping, which involves the extraction of data from websites, serves as a crucial component in the realms of data collection, research, and automation endeavours [22]. The Python package known as BeautifulSoup has garnered acclaim due to its user-friendly nature and its efficacy in parsing and traversing HTML and XML texts. The present paper examines the fundamental characteristics of BeautifulSoup and its significance in the practise of web scraping. The process of extracting data from web sites is made more efficient by using BeautifulSoup. This library parses the HTML or XML structure and offers a Pythonic approach to navigating and manipulating the content [23]. The primary functionalities of this software encompass element retrieval through techniques like as `.find()` and `.find_all()`, as well as attribute retrieval through `.text` and `.get()` methods. These functionalities empower users to efficiently identify and retrieve specific items present on a webpage, facilitating the extraction of their respective material with ease. BeautifulSoup is a widely utilised tool that has found applications in several sectors, encompassing data gathering, sentiment analysis, price tracking, and content scraping. Researchers and data analysts implement web scraping techniques to gather data from online sources, whereas corporations utilise it for the purpose of doing competitive analysis and market research [24]. The tool's versatility and compatibility with several Python modules, including Requests for web page retrieval, render it a great asset for automating tasks related to data collecting. The ethical and responsible approach to web scraping is of utmost importance. In order to adhere to ethical

REFERENCES

standards, online scraping activities should align with the terms of service of the targeted website, while also ensuring that the server is not burdened by an excessive number of requests. The utilisation of web scraping for personal or non-commercial objectives is often deemed more socially acceptable, but engaging in extensive scraping activities with the intention of generating profit or monetizing data may give rise to legal and ethical considerations [25].

```
links = soup.find_all("a", class_="l1ovpqvx bn2bl2p dir dir-ltr") # Find all elements with the tag <a>
for link in links:
    print("Link:", link.get("href"))

...

with open("scrapingdata.csv", "w", newline="", encoding="utf-8") as f:
    writer = csv.writer(f)
    # Writing the headers of the CSV file
    writer.writerow(["URL"])
with open("LINKS.csv", "r", newline="", encoding="utf-8") as f:
    reader = csv.reader(f)
    next(reader) # skip the header row

    for row in reader:
        url = row[0]

        response = requests.get(url)
        soup = BeautifulSoup(response.text, "html.parser")

        links = soup.find_all("a", class_="l1ovpqvx bn2bl2p dir dir-ltr") # Find all elements with the tag <a>
        for link in links:
            url=link.get("href")
            with open("scrapingdata.csv", "a", newline="", encoding="utf-8") as f:
                writer = csv.writer(f)
                writer.writerow([url])
```

Figure 2.1 Web scrabing snapshots

REFERENCES

2.1.7 Jaccard Similarity

Recommendation systems have emerged as indispensable tools for aiding users in the exploration of products, services, and content that align with their individual interests. In the domain of recommendation systems, Jaccard similarity has emerged as a powerful statistic for assessing the links between users and items. This literature study explores the importance of Jaccard similarity within recommendation systems, providing an in-depth analysis of its functionality and its utilization in the delivery of individualized recommendations.

The Jaccard similarity, which is a metric for measuring the similarity between sets, is of great significance in recommendation systems as it allows for the quantification of the degree of overlap between two sets: user preferences and item attributes. In the domain of recommendation systems, the Jaccard similarity metric is computed by evaluating the ratio of the intersecting items in the preferences of two users to the overall count of unique things that they have indicated a preference for.

The Jaccard similarity measure is widely employed in recommendation systems that utilize collaborative filtering, which is a prevalent methodology in the domain. Collaborative filtering is predicated on the notion that users who have demonstrated similar preferences in the past are likely to manifest similar preferences in the future. The utilization of Jaccard similarity is fundamental to this particular procedure.

In their influential study, Sarwar et al. (year) provide item-based collaborative filtering as a prominent approach [31]. This strategy, which largely relies on Jaccard similarity, is designed to create recommendations by analyzing user-item interactions. Furthermore, Desrosiers and Karypis provide an extensive analysis in their comprehensive review [32] that encompasses a range of neighborhood-based recommendation techniques, including those that utilize Jaccard similarity. This survey offers a detailed insight into the applications and intricacies of these methods.

REFERENCES

The Jaccard similarity metric is widely recognized for its utility in several domains. However, it encounters some obstacles pertaining to the scarcity of data and the ability to handle vast recommendation datasets, hence affecting its scalability. Significant progress has been achieved by researchers in tackling these difficulties. Algorithms designed to compute approximate Jaccard similarity have been identified as a viable approach [31] to improve the efficiency of recommendation systems, especially when dealing with large and sparse datasets.

The Jaccard similarity continues to be a fundamental concept in recommendation systems, facilitating the evaluation of user-item associations in a straightforward and efficient manner. The utilization of collaborative filtering-based recommendation systems remains effective in providing accurate and tailored recommendations. The continuous research endeavors aimed at addressing scalability issues highlight the lasting importance of Jaccard similarity in the constantly evolving field of recommendation systems.

2.2 Previous Work on Hotel Recommendations

2.2.1 Using Online Hotel Customer Reviews to Improve the Booking Process

This article presents a thorough examination of the several machine learning techniques employed in hotel recommendation systems. The authors effectively analyse the advantages and disadvantages of each algorithm and their application inside hotel recommendation systems. The study employed established and widely recognised evaluation metrics, hence enhancing the credibility of the findings. The essay also emphasises the importance of personalised recommendation systems in the hotel business. As the number of travellers and available accommodations continues to increase, there is an increasing expectation for hotels to offer personalised recommendations to their guests. The authors claim that the implementation of machine learning techniques, as exemplified in the proposed hotel recommendation system discussed in the article, has the potential to significantly enhance the precision and pertinence of such recommendations.

Additionally, the article by [26] sheds light on the challenges associated with the implementation of these systems. One of the foremost obstacles is in the acquisition of data of superior quality, encompassing customer preferences and comments. The authors suggest utilising natural language processing techniques, specifically TF-IDF and WordNetLemmatize, to extract valuable insights from user evaluations and enhance the accuracy of the recommendation system. One limitation of the research pertains to the very limited sample size employed in the investigations. The researchers utilised a dataset with a limited sample size of 80 hotels, potentially deviating from the representative nature of the whole population of hotels. The utilisation of a larger sample size would have yielded outcomes that are more credible and enhanced the potential to apply the findings to a broader population. Additionally, the essay emphasises the significance of evaluating the effectiveness of recommendation systems through the utilisation of metrics like as precision and recall. The authors introduce a novel assessment framework that integrates both objective and subjective factors, considering the relevance of suggested hotels and user satisfaction [27].

REFERENCES

In general, the source [26] provides valuable insights into the design and assessment of machine learning-driven hotel recommendation systems. The writers have made significant contributions to the progress of this field by tackling challenges and presenting novel methodologies for the processing and interpretation of data. The significance of personalised recommendation systems is expected to grow in tandem with the expansion of the hotel industry. The study presented in this article offers valuable insights that can inform and shape future developments in this domain.

2.2.2 Recommendation System based on Data Mining

The researchers utilized the Expedia Hotel Recommendation dataset obtained from Kaggle, which consists of a total of 37,670,293 items. Additionally, the test set used in the study contained 2,528,243 entries. Furthermore, the dataset has specific underlying attributes for each of the locations documented in both the train and test sets. Initially, the system acquires user search data from multiple web platforms dedicated to hotel booking. Subsequently, the gathered data is utilized to create a matrix that associates users with items, wherein each row corresponds to an individual and each column corresponds to a hotel. The matrix is employed in the context of collaborative filtering, wherein it recommends hotels by leveraging the preferences of other users who have conducted comparable searches. The researchers utilized a dataset acquired from a hotel booking service in order to train and evaluate the system. The information encompasses both the search history of users and many criteria related to hotels, including price, location, and facilities. The authors conducted a comparative analysis between the proposed strategy and several baseline strategies, such as basic collaborative filtering and content-based filtering [2].

The data has been anonymized, and the majority of the variables are represented in numerical form. The objective is to estimate five hotel clusters out of a sample of 100, which are more probable choices for a user's accommodation. The problem was framed as a task of ranked multi-class classification. The dataset presents several significant concerns, namely the presence of missing data, the necessity for ranking criteria, and the challenge posed by the curse of dimensionality.

In general, the essay provides a comprehensive analysis of a personalized hotel recommendation system that utilizes collaborative filtering and deep learning methodologies. The enhancement of customers' hotel booking experiences could potentially be achieved by the provision of personalized and precise hotel recommendations, which are derived from an analysis of their individual search history.

One advantage that the system can offer is the ability to personalize. The system provides personalized recommendations by analyzing the user's search history and preferences, potentially enhancing user experience and satisfaction.

REFERENCES

The strategy employed in this study involves a combination of methodologies, including conducting research on existing systems and drawing insights from them. In order to provide dependable recommendations, the system incorporates a combination of collaborative filtering and deep learning methodologies, hence potentially improving the accuracy of recommendations compared to the utilization of a single approach.

The system's predictive capabilities were evaluated by employing a real-world dataset, as opposed to a fabricated one, and comparing its performance against several baseline approaches. The objective was to assess the system's efficacy in predicting user preferences.

One primary problem pertains to data privacy, as the system relies on the collection of user search history data, hence raising apprehensions over privacy and security.

As the user base expands, the system's capacity to handle the increasing demands may become limited due to the resource-intensive nature of processing and storing user data, which necessitates significant computational power and storage capacity. Ultimately, the efficacy of the system in offering hotel recommendations to inexperienced users without a search history may be compromised due to its dependence on user data for generating personalized suggestions.

Ultimately, the deep learning model employed by the system may exhibit a susceptibility to overfitting, leading to suboptimal generalization and diminished accuracy in predicting outcomes on novel data.

REFERENCES

2.2.3 Trivago

The recommendation engine employed by Trivago incorporates various factors, including the user's historical search data, their present geographical location, the intended departure date, and any other pertinent search criteria. By utilizing the collected data, it is possible to formulate accommodation recommendations that are tailored to the specific requirements and interests of the user. When conducting a search, Trivago considers factors such as the hotel's availability, the number of rooms offered, and the price of each individual room. This ensures that the hotels suggested to the user are immediately operational and capable of accommodating their specific requirements. In the assessment of a hotel's reputation and status, Trivago considers the evaluations submitted by prior guests alongside the perspectives of unbiased third-party experts. Users can have confidence that the hotels they suggest have undergone a comprehensive assessment and will offer a satisfactory lodging experience should they opt for one of Trivago's recommendations.

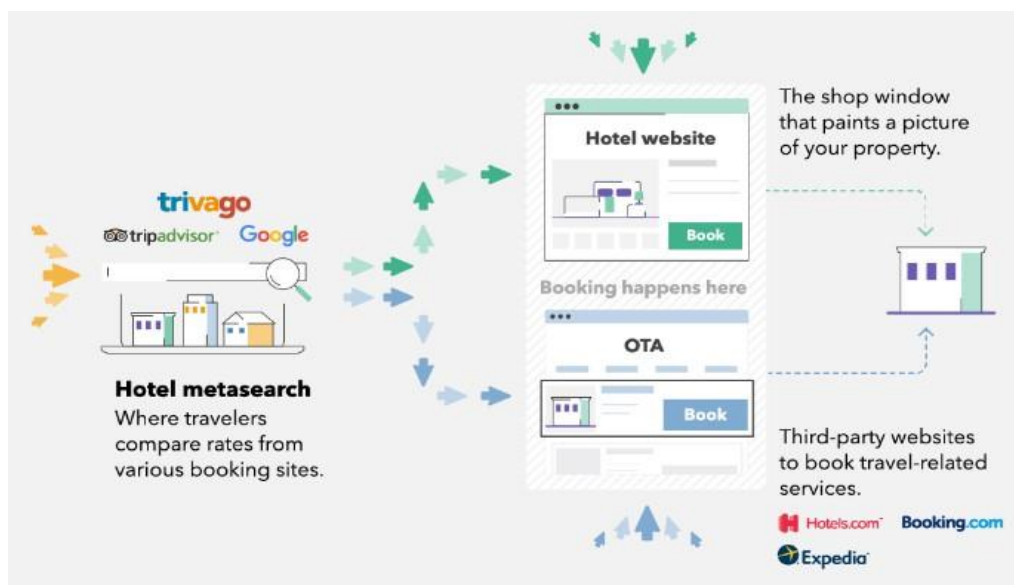


Figure 3.0 Trivago

Trivago employs a comparative approach to ensure the provision of hotels with reasonable prices by juxtaposing their expenses with those of other establishments in the same region and possessing similar services. Visitors now have the convenience of readily identifying motels that best align with their budgetary constraints. In addition to factors like as price, location, and star rating, Trivago takes into account the presence of complementary breakfast, Wi-Fi, or parking facilities offered by the hotel. This

REFERENCES

feature enhances the process of locating hotels that offer specific services and amenities that cater to the preferences of individual consumers [1].

Trivago's recommendation engine is designed to offer personalized suggestions to consumers, taking into account several factors such as their search history, preferences, location, as well as the hotel's quality, popularity, price, and facilities. The temporal and financial resources conserved via the utilization of this methodology for discerning an appropriate lodging establishment have numerous individuals encounter difficulties in selecting an appropriate hotel for their travel endeavors due to the overwhelming abundance of available choices.

Recommendation systems play a significant role in assisting users in making informed decisions. Trivago employs machine learning algorithms in order to provide personalized hotel recommendations to its users.

Similar to other recommendation systems, it possesses both advantages and disadvantages. The recommendation system employed by Trivago demonstrates exceptional proficiency in customisation. The system utilizes individualized recommendations derived from search patterns, geographical data, travel schedules, and additional variables. The implementation of personalization plays a crucial role in facilitating users' ability to identify suitable hotels.

One of Trivago's notable strengths is in its real-time suggestion system. The search process incorporates real-time updates to provide information on the current availability of hotels. This feature guarantees that users are presented with a comprehensive list of hotels that are currently available for their specified travel dates. This practice guarantees that the hotels recommended to users have competitive pricing options for individuals who prioritize value.

However, the recommendation mechanism of Trivago also has a number of limitations. One problem that can be identified is a lack of openness. The lack of transparency in the algorithm used for generating suggestions results in users' limited understanding of the underlying process. Certain consumers may exhibit a sense of skepticism towards the information presented on the website due to the absence of any accompanying explanations or justifications.

One further area of vulnerability pertains to the sources of data. Trivago utilizes search behavior and availability data to provide hotel recommendations. The significance of

REFERENCES

this data should be acknowledged, although it is worth noting that it may not encompass all the variables that users take into account when making a decision about a hotel, such as the availability of amenities and proximity to local attractions. When a user does a search for high-end hotels, the recommendation system may exhibit a bias towards pricier accommodations, potentially overlooking the user's specific requirements and financial constraints.

Chapter 3

Proposed Method/Approach

The Cross-Industry Standard Process for Data Mining (CRISP-DM) will be the suggested methodology for this study. It has been a tried-and-true method for directing data mining projects. The project's procedures were divided up into several development stages.

3.1 CRISP – DM

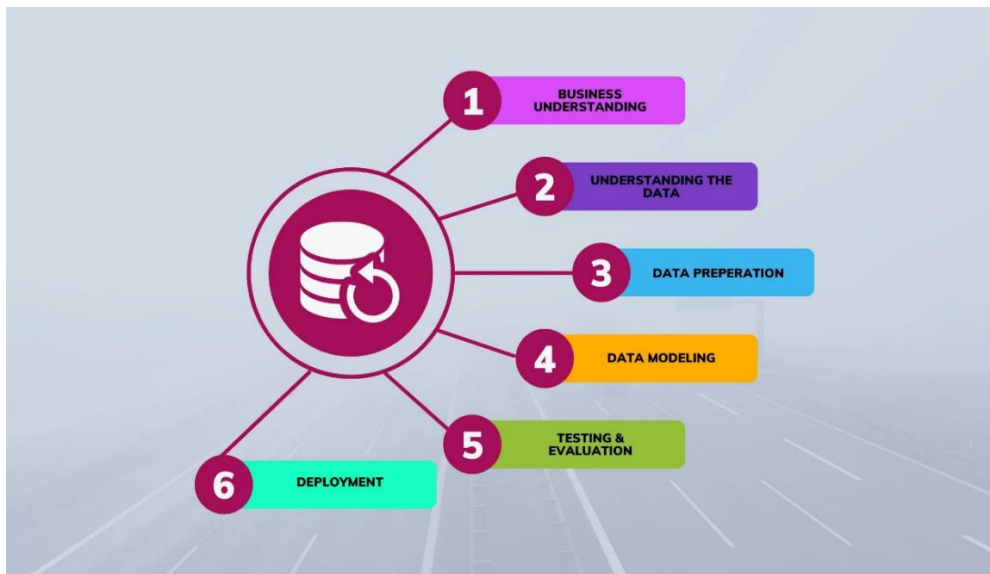


Figure 4.0 Crisp – DM

The utilization of data mining is of great importance in business and organizational domains as it provides vital insights and tackles intricate issues within the current data-driven environment. The process of data mining encompasses the utilization of statistical and machine learning methodologies to discern and analyze patterns and trends within vast collections of data. In order to get desirable results, it is important to employ a methodical strategy. The CRISP-DM model is utilized inside this particular scenario.

REFERENCES

3.1.1 Business Understanding

The first phase of the CRISP-DM system is known as Business Understanding. The task encompasses a thorough investigation of the business issue now under consideration. During this stage, the emphasis is to determine what objectives and goals it intends to accomplish. Establishing the parameters for evaluating the project's success and gaining a comprehensive comprehension of the stipulations and limitations associated with the business predicament. The initial phase of Business Understanding establishes the fundamental basis for the entirety of the data mining project, guaranteeing coherence with the objectives of the business.

3.1.2 Data Understanding

The phase of Data Understanding involves a complete exploration and analysis of the available data. The primary tasks during this period encompasses the process of gathering and evaluating the data sources that are pertinent to the project. Acquiring a comprehensive understanding of the structure, quality, and potential challenges associated with the data. The process involves the identification of data patterns, correlations, and first insights that can provide valuable information for following phases and the process of Data Understanding is essential for the project team to have a comprehensive understanding of the data prior to advancing to further stages.

3.1.3 Data Preprocessing

The process of Data Preparation encompasses the manipulation and preprocessing of data in order to render it appropriate for the purpose of modeling. The activities encompassed during this phase includes the process of cleaning and managing data that is either absent or inconsistent. The process of identifying pertinent traits and factors. The process of encoding categorical data and scaling numerical data is commonly employed in data analysis and machine learning tasks. Categorical data refers to variables that take on discrete values from a limited set of categories, while numerical data refers to variables that take on continuous values. Encoding categorical data involves transforming these variables into a numerical representation that may be used. The process of dividing the data into separate training and testing sets and the process of data preparation is essential in ensuring that the data is appropriately formatted for the building of models.

3.1.4 Modeling

Modeling refers to the application of machine learning and statistical approaches in order to construct predictive or descriptive models. This stage includes the process of choosing suitable modeling techniques. The process of training and fine-tuning the models is a crucial step in machine learning, assessing the performance of a model through the utilization of appropriate metrics and the primary objective of modeling is to develop precise models that effectively tackle the given business problem.

REFERENCES

3.1.5 Evaluation

Evaluation is the comprehensive analysis of the model's performance in order to ascertain its alignment with the objectives of the project. The primary activities includes evaluating the models based on the success criteria established at the initial step of Business Understanding. Conducting cross-validation and evaluating the models using previously unseen data, identifying areas for enhancing the model's performance. The process of evaluation is essential in order to ascertain the reliability and effectiveness of the produced models.

3.1.6 Implementation and Deployment

The deployment phase is the culmination of the development process, during which the models that have been created are implemented and utilized in a practical manner within the operational context of the business. The activities encompassed under this context are the process of incorporating the models into pre-existing systems or processes. The provision of documentation and training aimed at end-users. The task of overseeing and preserving the operational status of the models. The process of deployment is essential in order to ensure that the models effectively provide value to the enterprise.

The CRISP-DM framework provides a flexible and iterative methodology that allows for input and modifications throughout the project. This framework offers a structured approach for data scientists and stakeholders to explain the business problem, locate pertinent data, do data preparation, design and evaluate models, and effectively deploy them.

REFERENCES

3.2 Hardware

The hardware involved in this project is a laptop. The laptop is issued for the process of data mining algorithms and recommendation model, then it will be also used for the deployment of the final version of the applications.

Table 3.1 Specifications of laptop

Description	Specifications
Model	MSI GF63 Thin 10SC
Processor	Intel Core i5-10200H
Operating System	Windows 10
Graphic	NVIDIA GeForce GTX 1650
Memory	12GB DDR4 RAM
Storage	500GB SATA SSD

CHAPTER 4

System Design

4.1 System Block Diagram

An overview of a machine learning-based hotel recommendation system is shown in the following block diagram. The system is composed of the following elements:

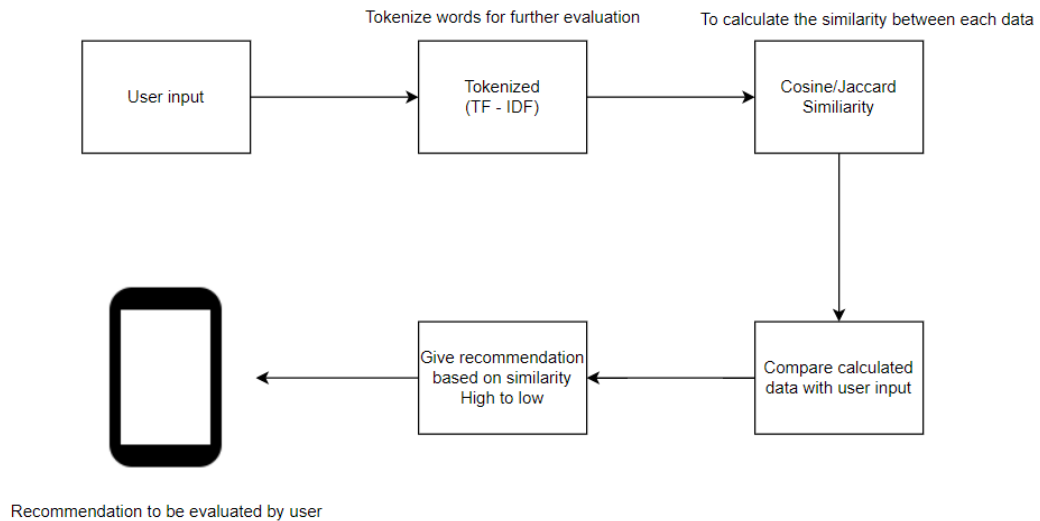


Figure 5.0 System Block Diagram

4.1.1 Overall System Architecture

Data collection: This part compiles information about lodging and user preferences. Many other sources, including hotel websites, customer reviews, and social media, may provide this information.

Preparing the data for machine learning is known as data preparation or data preprocess. This might include classifying category data, eliminating outliers, and imputing missing values.

Feature engineering: This part builds new features from the data that already exists. The machine learning model's performance can be enhanced by using these additional characteristics.

Model training/Modeling: This step teaches a machine learning model how to relate hotel amenities to guest preferences. Numerous machine learning methods, including collaborative filtering, random forests, and neural networks, may be used to do this.

Generating suggestions for users: This part makes use of the machine learning model that has been developed. The cosine/Jaccard similarity between the user's choices and the hotel attributes may be used to do this.

REFERENCES

Presentation of recommendation: This part shows the user the recommendations. Many other methods, such as a website, a mobile app, or email, are available for doing this.

4.2 Performance and Scalability

Performance Metrics and Goals

Recommendation Accuracy: This number measures how well the algorithm suggests hotels based on user choices. Set a goal of achieving at least 75% accuracy for both Jaccard and cosine similarity-based suggestions. The accuracy of suggestions may be assessed using past user feedback.

Response Time: User satisfaction is highly dependent on response time. The system should strive to react to user requests in a timely manner. For 90% of user inquiries, a goal response time aim may be to deliver suggestions in less than 2 seconds.

Scalability is vital since the system must accommodate growing user traffic and data volume. When the average system demand exceeds 70%, establishing a target to grow horizontally by adding additional servers or resources. This guarantees that response times stay constant even when traffic is high.

Usage of Resources: Efficient resource use is critical for cost-effectiveness. During typical operation, the system should attempt to keep CPU and memory use below 70%, with leeway for surges during high demand.

Recommendation Diversity: Recommendation diversity guarantees that consumers get a diverse set of recommendations. One goal may be to deliver at least five different hotel suggestions for each user inquiry, guaranteeing a balanced range of hotels.

CHAPTER 5

System Implementation

5.1 Setting up

5.1.1 Software

Before starting to develop the recommendation system model, there are four software needed to be installed and downloaded in my laptop:

1. Anaconda Navigator Jupiter Lab
2. R Studio
3. Flask

5.2 CRIPS-DM

5.2.1 Business Understanding

During the initial phase, referred to as "Business Understanding," the user will acquire knowledge regarding the specific problem faced by the company and explore the potential of data mining techniques in addressing this issue. The initial step involves the identification of the objectives to be achieved, the key stakeholders to be engaged, and the intended outcome of the endeavor. The process of gathering, delineating, and examining the data that will be utilized in the project is referred to as Data Understanding, which constitutes the second phase [12].

Individuals who have challenges in choosing a hotel that adequately meets their preferences and requirements may find value in utilizing the ongoing development of a hotel recommendation system. The primary goals of the system are to optimize consumer satisfaction, stimulate an upsurge in reservation frequency, and ultimately improve the overall consumer experience. One of the objectives of the project is to develop a system capable of generating personalized hotel suggestions by analyzing customer comments and preferences. The target demographic encompasses a diverse range of individuals, comprising customers, proprietors, and executives within the hospitality industry. The success measures will encompass the precision of the recommendations, the number of successful bookings, and the level of contentment demonstrated by the target demographic. The primary objective of the system is to optimize the whole client experience through the provision of precise and personalized hotel recommendations. This, in turn, would aid hotels and other stakeholders in the hospitality sector in attaining financial prosperity.

REFERENCES

```
header = ["name", "hoteltype", "address", "rating", "review"]

# Initialize the output CSV file with the header row
with open("modern.csv", "w", newline="", encoding="utf-8") as f:
    writer = csv.writer(f)
    writer.writerow(header)
with open("scrapingdata.csv", "r", newline="", encoding="utf-8") as f:
    reader = csv.reader(f)
    next(reader) # skip the header row

for row in reader:
    url = row[0]
    url = "https://www.airbnb.com/" + url
    try:
        response = requests.get(url, headers=headers)
        response.raise_for_status()
    except requests.exceptions.RequestException as e:
        print(f"Error requesting {url}: {e}")
        continue
    soup = BeautifulSoup(response.content, "html.parser")
    script = soup.find("title").text
    name = script

# Define the regular expression pattern
pattern = r"content="(.*?)" property="og:title"
script = soup.find_all("meta")

# Iterate over the content list
for item in script:
    # Convert the item to a string
    item_str = str(item)

    # Use the re.search() function to find the matching pattern in each item
    match = re.search(pattern, item_str)

    # Extract the desired data from the match
    if match:
        data = match.group(1)
        hoteltype = data.split(" in ")[0]
        address = None
        address_data = data.split(" in ")
        if len(address_data) > 1:
            address = address_data[1].split(".")[0].strip()
        rating = None
        rating_data = data.split("★")
        if len(rating_data) > 1:
            rating = rating_data[1].split()[0]
        review = None
        review_data = data.split("★")
        if len(review_data) > 1:
            review = " ".join(review_data[1].split()[2:])
        with open("modern.csv", "a", newline="", encoding="utf-8") as f:
            writer = csv.writer(f)
            writer.writerow([name, hoteltype, address, rating, review])
```

Figure 6.0 Recommendation Functions

The Python script shown functions as a means of gathering and preparing data within the confines of the CRISP-DM architecture. Commencing with the initial phase of "Business Understanding," the script aptly opts for Airbnb as the chosen data source, thereby aligning with the project's defined objectives. The process of extraction involves retrieving essential data such as the names of hotels, their respective types, addresses, ratings, and reviews, all of which are relevant to the objectives of the project.

In the phase of "Data Understanding," the methodology utilises web scraping methods to gather data from Airbnb listings, which is deemed as a viable data source in

REFERENCES

alignment with the project's objectives. Data preparation is conducted, employing regular expressions to parse and extract pertinent information from the HTML text.

During the "Data Preparation" step, the script addresses missing values by assigning the value of "None" in instances where data is not present in the web page content. Nevertheless, it is crucial to take into account the potential consequences of these missing variables on subsequent studies. The data that has been extracted is converted into a structured format and afterwards saved in a CSV file, so facilitating its suitability for subsequent study.

The script does not incorporate a modelling step, as its main emphasis is on the extraction and preparation of data. In order to fully adhere to the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, it is important to include the sequential stages of modelling, assessment, and implementation. The process of evaluation involves the comprehensive assessment of several aspects such as the quality, correctness, and completeness of the data. This evaluation aims to ensure that the scraped data is in alignment with the objectives of the project.

During the "Deployment" step, the output data generated by the script can be utilised for the purposes of modelling or conducting additional analysis. It is imperative to acknowledge that the CRISP-DM methodology is characterised by its iterative nature. The present script exemplifies a solitary iteration that centres on the tasks of data gathering and preparation. Potential future iterations of the project may encompass the enhancement of the scraping procedure, the resolution of exceptional scenarios, or the adjustment to modifications occurring on the Airbnb website.

In order to optimise the efficacy of the script, it is advisable to address the issue of missing data in a more specific manner, as well as conduct a thorough and rigorous assessment of the scraped data's quality and trustworthiness. These methods will guarantee that the data gathered is in perfect accordance with the project's needs and objectives within the framework of a full CRISP-DM workflow.

REFERENCES

5.2.2 Data Understanding

Data Quality Report

Combined_Hotel_Details.csv	
Number of Rows	1544
Number of Columns	6
Missing Value	543
Duplicate Rows	0
Column Name	Missing Value
hoteldetails	40
hoteltype	63
address	20
rating	112
specialities	107
hotelname	201

Table 1.0 Data quality

Column Name	Quality Issues
hoteldetails	none
hoteltype	The text has a significant number of stop words and punctuation marks.
address	The dataset exhibits the presence of unclean or inaccurate data, and lacks the ability to effectively assign distinct values to individual data points.
rating	The user did not provide any text to rewrite.
specialities	The dataset exhibits the presence of unclean or inaccurate data, and lacks the ability to effectively assign distinct values.
hotelname	The dataset exhibits the presence of unclean or inaccurate data, and lacks the ability to effectively assign distinct values.

Table 2.0 Data Quality issues

REFERENCES

During the second phase, known as "Data Understanding," the project entails the collection, description, and examination of the data that will be used. This includes the identification of data sources, assessment of data quality, and performance of exploratory data analysis [30].

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	hoteldetails	hoteltype	address	rating	roomamenities	specialities	hotelname						
2	ATTIC Sky Garden Private Pool Luxury Family Suite		Melaka	4.84	2 bedrooms	3 normal	ATTIC Sky Garden Private Pool Luxury Family Suite						
3	Alamanda Hilltop Villa	Villa	Langkawi	4.77	1 bedroom	1 b normal	Alamanda Hilltop Villa - Stunning Sunset View						
4	[ATTIC] Peak of MLK S Condo		Melaka	4.84	Studio	1 bed normal	[ATTIC] Peak of MLK Seaview @Melaka Town[TVBOX]						
5	Alamanda Tropical Wooden Villa - Pool View		Langkawi	4.87	Studio	3 beds normal	Alamanda Tropical Wooden Villa - Pool View						
6	180 Sunrise Seaview Seafront Duplex		George Town	4.92	2 bedrooms	5 normal	180 Sunrise Seaview Seafront Duplex						
7	Not Just A Homestay @ Broga, Semenyih		Kuala Lumpur	4.97	1 bedroom	2 b normal	Not Just A Homestay @ bathtub netflix projector						
8	MJHolidayB2630 {PlayfulPalace}Sunrise+ FreeCarPark		Melaka	5	1 bedroom	3 b normal	MJHolidayB2630 {PlayfulPalace}Sunrise+ FreeCarPark						
9	TTS Beach Village @ Broga, Semenyih		Semenyih	4.93	5 bedrooms	2 normal	TTS Beach Village @ Broga, Semenyih						
10	IKAN Residence Hidden Gem Villa with Forest View		Bentong	4.96	6 bedrooms	21 normal	IKAN Residence Hidden Gem Villa with Forest View						
11	Mountain View Getaway House with Pool (Lauhaus)		Kuala Kubu	4.76	2 bedrooms	4 normal	Mountain View Getaway House with Pool (Lauhaus)						
12	Port Dickson Pool Villa - TwentyFour.7		Port Dickson	5	5 bedrooms	11 normal	Port Dickson Pool Villa - TwentyFour.7						
13	Port Dickson Beach Front Villa w/ Private Pool		Port Dickson	4.91	4 bedrooms	5 normal	Port Dickson Beach Front Villa w/ Private Pool						
14	TTS Lake Villa @ Broga, Semenyih		Semenyih	4.75	4 bedrooms	1 normal	TTS Lake Villa @ Broga, Semenyih						
15	Arte Mont Kiara Muji Suites		Federal Territory	4.59	2 bedrooms	3 normal	Arte Mont Kiara Muji Suites						
16	Templer Park Rainforest Retreat - Villa		Rawang	4.76	4 bedrooms	1 normal	Templer Park Rainforest Retreat - Villa						
17	Rumah Hitam Puteh + Private Swimming Pool		Kajang	4.98	5 bedrooms	8 normal	Rumah Hitam Puteh + Private Swimming Pool						
18	A21-08/5 min drive to Jonker/Imperio/Studio/3 pax		Melaka	4.37	Studio	2 beds normal	A21-08/5 min drive to Jonker/Imperio/Studio/3 pax						
19	Cozy Imperio Suites/ Bathtub Free Bathbomb/ Ramen		Melaka	4.79	1 bedroom	1 b normal	Cozy Imperio Suites/ Bathtub Free Bathbomb/ Ramen						
20	Imperium Residence Kuantan, City Light View+NETFLIX		Kuantan	4.96	1 bedroom	1 b normal	Imperium Residence Kuantan, City Light View+NETFLIX						
21	Yuma Atlantis Melaka/3BR 8Pax WiFi Balcony SeaView		Melaka	4.54	3 bedrooms	4 normal	Yuma Atlantis Melaka/3BR 8Pax WiFi Balcony SeaView						
22	Pulaithree retreat, Kuala Kubu Bharu Heights		Kuala Kubu	4.79	2 bedrooms	6 normal	Pulaithree retreat, Kuala Kubu Bharu Heights						
23	MWHolidayB2441 Cozy Sunset Private Jacuzzi+Seaview		Melaka	4.88	1 bedroom	2 b normal	MWHolidayB2441 Cozy Sunset Private Jacuzzi+Seaview						
24	VILLA SAJURI Homestay, Event Space @Pantai Remis		Jeram	4.67	4 bedrooms	1 normal	VILLA SAJURI Homestay, Event Space @Pantai Remis						
25	Yussy Fairy Projector SeaView 2BR @ R&F Princess		Johor Bahru	4.84	2 bedrooms	2 normal	Yussy Fairy Projector SeaView 2BR @ R&F Princess						
26	Monochrome Designer Loft KLIA Cyberjaya WiFi		Cyberjaya	4.76	1 bedroom	1 b normal	Monochrome Designer Loft KLIA Cyberjaya WiFi						

Figure 6.1 Data Set

The dataset used in this project has undergone a transition from its initial source on Kaggle to being scraped directly from a website. The visualizations presented in this report are based on a combined CSV file created from a small portion of this newly acquired dataset. This modification was made to facilitate easier data manipulation and analysis. The objective of these visualizations is to enhance our understanding of the dataset by exploring potential patterns within it. Specifically, they aim to identify patterns related to room types, room amenities, and the hotel's location—key factors that users consider when using a recommendation system to select a personalized hotel. This report outlines the visualizations conducted on the dataset to uncover insights and patterns that can inform the development of a recommendation system.

REFERENCES

The original dataset acquired from Kaggle has been substituted with a novel dataset collected via web scraping. As a result, the visualisations showcased in this work have been modified to align with the qualities and features of the revised dataset. The primary objective remains unchanged: to derive important insights and identify relevant patterns within this dataset. These patterns play a crucial role in the formulation of a recommendation system that is tailored to user preferences, specifically emphasising three primary factors: room types, room amenities, and hotel locations. This part presents a selection of visualisations that have been customised for the dataset at hand. The focus is on highlighting how these visualisations enhance understanding of the data and their usefulness in generating practical insights for the design of the recommendation system.

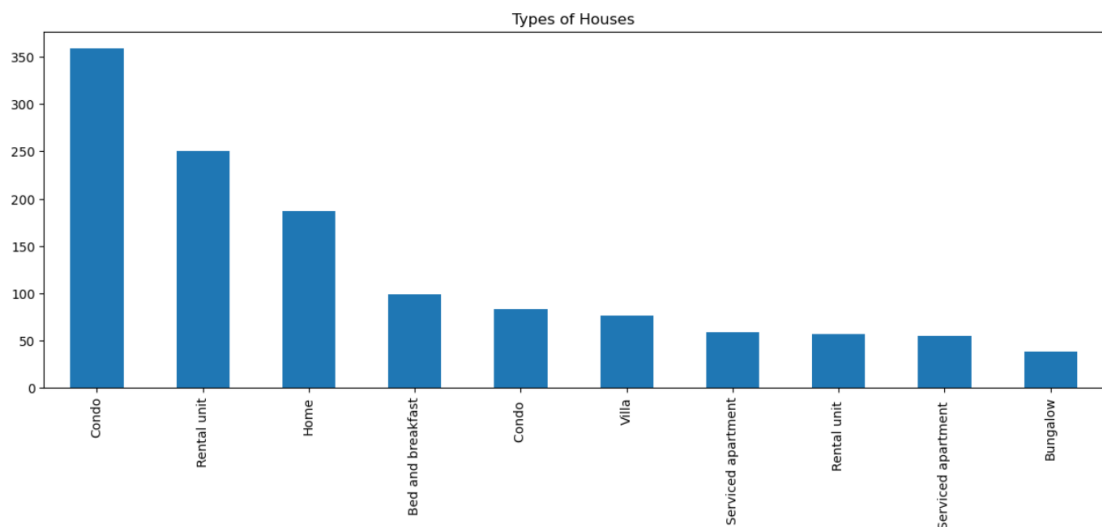


Figure 6.2 Data Visualizing Bar Chart

REFERENCES

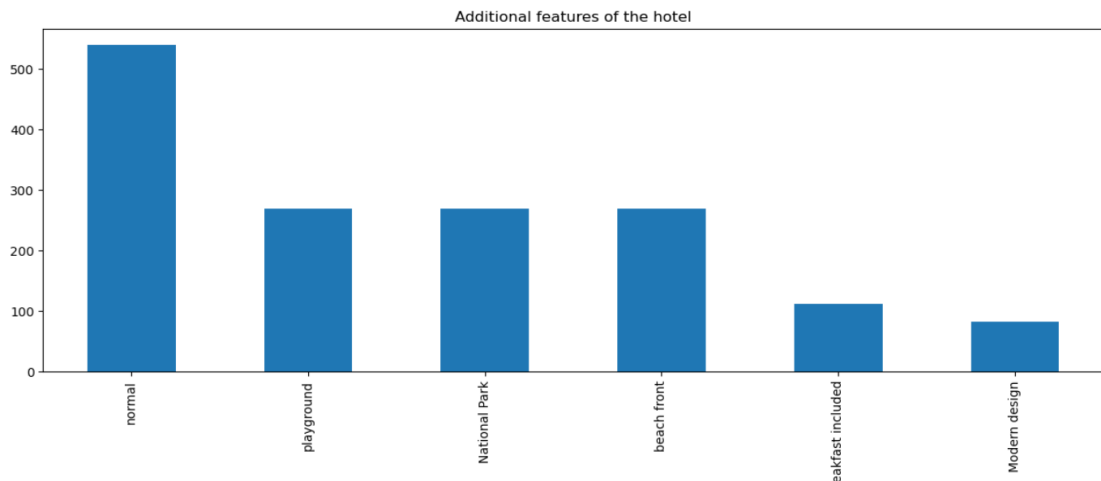


Figure 6.3 Data Visualizing Bar Chart

The provided figure shows a graph, illustrating the several categories of residential properties, including condominiums, rental units, and bungalows. Prior to advancing to the subsequent stage of the project, a thorough examination of the dataset was conducted. During each precautionary stage of the inspection process, measures were taken to ensure the absence of duplicate entries, empty fields, or null values inside the dataset. Nevertheless, it has been determined that there are specific flaws that continue to exist within the dataset, especially impacting the categorization of room types and the identification of hotel names. These concerns suggest the inclusion of data that is inaccurate, inconsistent, or incomplete. Hence, the precise identification of values for these variables was a significant challenge.

The second figure depicts the same graph as the first, but with distinct categories of data representing additional features of the hotels. These features include, but are not limited to, hotels with nearby playgrounds, national parks, and other relevant attributes. Given the aforementioned factors, the project has successfully conducted data processing, leading to the production of a refined dataset that is appropriate for further study. This accomplishment was facilitated by the utilisation of grouping methodologies and the subsequent incorporation of the processed data into the pre-defined table.

Moreover, the decreased occurrence of unclean data has significantly diminished the requirement for intensive preprocessing endeavours, hence fitting with the fundamental tenets of the CRISP-DM (Cross-Industry Standard Process for Data Mining) approach. The advantage of reducing data cleaning and preprocessing needs is that it facilitates a

REFERENCES

smoother and more efficient transition into the analysis phase of the project, resulting in time and resource savings.

5.2.3 Data Preparation

During the third stage of the modelling process, referred to as "Data Preparation," the data undergo cleaning, transformation, and preprocessing procedures to ensure their suitability for utilisation. The process entails the identification and inclusion of desirable attributes, the handling of incomplete or anomalous data points, and the generation of novel variables [30].

The data has undergone a process of cleaning since the previous phase, and via extra verification, it has been confirmed that there are no missing or duplicated data present. In order to enhance the efficiency and accuracy of modelling, it is important to preprocess the data prior to its use in the modelling process.

```
#data['hotelname'] = data['hoteldetails']
data['hotelname'] = data['hoteldetails'].apply(lambda x: x.rsplit('-', 2)[-3] if x.count('-') >= 2 else x)
data.to_csv('updateddata.csv', index=False)
print(data['hotelname'])

0      ♦ATTIC♦Sky Garden Private Pool Luxury Family S...
1      Alamanda Hilltop Villa - Stunning Sunset View
2      [ATTIC] Peak of MLK Seaview @Melaka Town[TVBOX]
3      Alamanda Tropical Wooden Villa - Pool View
4      180 Sunrise Seaview Seafront Duplex  无敌海景套房 11
...
1538   Sarcelle by ALV, 2BR Stylish Georgetown APT/HTL
1539                                     The 187-House  187小屋
1540   5* Holiday Home @ Centurixa-Tamara Putrajaya
1541   Home Art 103-Iph Garden/Tambun Lost World/6-8 ...
1542                                     (NEW) DeLux Suites / KLCC / 3BR6pax
Name: hotelname, Length: 1543, dtype: object

# Create a new column 'hotelname' to store the updated values
data['hotelname'] = data['hoteldetails']

# Iterate over the values in the 'hoteldetails' column
for i, value in enumerate(data['hoteldetails']):
    # Check if the value contains three '-'
    if value.count('-') == 3:
        # Split the value based on the second '-'
        data.loc[i, 'hotelname'] = value.split('-', 2)[-1]
    # Check if the value contains two '-'
    elif value.count('-') == 2:
        # Split the value based on the first '-'
        data.loc[i, 'hotelname'] = value.split('-', 1)[0]
print(data['hotelname'])
# Save the DataFrame to a CSV file
data.to_csv('updated_data.csv', index=False)
```

Figure 6.4 Data Cleaning Python Scripts

REFERENCES

One of the often encountered use cases of the split approach is to the task of cleansing and arranging data. The process of dividing strings into distinct segments based on designated delimiters, such as tabs, commas, spaces, or other specified patterns, facilitates the segregation of data into its corresponding columns or constituent elements. The acquisition of data frequently involves the use of sources such as CSV files, log files, or web scraping, whereby discrepancies or extraneous information may be present, hence necessitating the extraction of pertinent insights.

To begin with, the CSV file can be parsed using the pandas module in Python. After the data has been put into a data frame, it is possible to cycle through each row and utilise the split function. For instance, in the case when each row is represented as a string with comma-separated values, the split technique can be employed by setting the delimiter as a comma. This operation will partition each row into its corresponding elements, resulting in distinct columns for the name, contact information, and purchase records.

Moreover, the utilisation of the split method allows for the extraction of particular segments inside a string, while disregarding extraneous data. For example, in the case where the purchase history comprises many products, it is possible to divide the string by utilising a comma as a delimiter, so separating each individual product. This enables the examination of individual transactions, the execution of computations, and the extraction of valuable observations without the need for laborious manual intervention. Furthermore, the split method demonstrates a high level of effectiveness in handling irregular patterns within data. In instances when there are inconsistent numbers of spaces or tabs between data components, the split mechanism can be enhanced to adapt to these patterns dynamically. The utilisation of regular expressions enables the specification of various delimiters or patterns, hence enhancing the versatility of the split approach for intricate data cleansing jobs.

REFERENCES

```
print(data.isnull().sum())
hoteldetails      0
hoteltype         0
address           0
rating            155
roomamenities     155
specialities      0
dtype: int64
```

```
column_data_types = data.dtypes
print(column_data_types)
hoteldetails      object
hoteltype         object
address           object
rating            float64
roomamenities     object
specialities      object
hotelname         object
dtype: object
```

```
data['rating'] = data['rating'].astype(str)
```

Figure 6.5 & Figure 6.6 Data type Checking Python Scripts

The presence of duplicate entries within a dataset has the potential to result in erroneous analysis and an exaggerated representation of certain values. Python provides effective techniques for identifying and managing duplicate elements. One commonly employed method is utilising the Pandas library, renowned for its robust capabilities in data processing. The `duplicated()` function in the Pandas library enables the identification of duplicate rows or columns by applying specific criteria. Null values, commonly represented as NaN (Not a Number), have the potential to introduce inconsistencies in data processing if they are not appropriately managed. Python offers comprehensive capabilities for the detection and management of null values. Once again, Pandas proves to be a valuable tool with its built-in routines designed for the detection and handling of null values. In the process of inputting data into a dataset, it is possible for the allocated data types to be incorrect, resulting in inconsistencies and inaccuracies that might adversely impact subsequent analyses. Python provides a range of functions that are both straightforward and robust in converting data types and maintaining the integrity of the dataset. Data cleaning is a fundamental component of the data analysis process, and Python offers a wide range of tools and packages to streamline this effort. Through the utilisation of Python scripts, it is possible to effectively examine for duplicate entries, identify null values, and execute data type conversions. The robust capabilities offered by libraries such as Pandas facilitate the process of tidying and harmonising disorderly and incongruous data, thereby guaranteeing its precision and dependability for subsequent analysis.

REFERENCES

```
import pandas as pd
from sklearn.impute import SimpleImputer

# Identify columns with missing values
columns_with_missing = data.columns[data.isnull().any()]

# Create an instance of SimpleImputer
imputer = SimpleImputer(strategy='mean') # You can choose a different strategy like 'median' or 'most_frequent'

# Fit the imputer to the data
imputer.fit(data[columns_with_missing])

# Perform imputation on the missing values
data[columns_with_missing] = imputer.transform(data[columns_with_missing])

# Print the updated DataFrame with filled-in missing values
data.to_csv("predicted_data.csv", index=False)
```

Figure 6.7 Simple Imputation Python Scripts

Utilising simple imputation as a means to handle missing values in the ratings column of the dataset is a pragmatic and data-conserving methodology. The preservation of the dataset's overall structure is essential as it enables the uninterrupted examination of the complete dataset, hence preventing the loss of vital information. Nevertheless, it is imperative to recognise that this particular approach operates under the assumption that the absence of data is missing at random (MAR). This implies that the probability of data being absent is not tied to the unobserved value, but it may be influenced by other variables that have been observed. Hence, it is imperative to evaluate the veracity of this assumption in relation to the dataset at hand.

Furthermore, it is crucial to acknowledge that the utilisation of simple imputation techniques, such as mean or median imputation, has the potential to generate bias within the dataset. This bias arises from the replacement of missing values with measures of central tendency. The potential impact of this factor on the distribution of the ratings variable may have implications for further analysis or modelling endeavours. It is imperative to engage in a thorough examination of the potential ramifications of this prejudice within the particular scenario at hand.

Although simple imputation is frequently utilised, it is advisable to investigate more advanced imputation approaches, such as regression imputation or multiple imputation, based on the specific properties of dataset. These methodologies have the capability to consider the interdependencies among variables, which might potentially result in more

REFERENCES

precise imputed values. Furthermore, doing a sensitivity analysis to evaluate the influence of various imputation techniques on the outcomes might offer valuable insights into the reliability and stability of the results obtained from the selected imputation approach. In summary, the utilisation of simple imputation serves as a practical method to tackle the issue of missing numerical data, especially where ensuring data integrity and preserving sample size are of utmost importance. Nevertheless, it is crucial to acknowledge the limitations and potential ramifications of this approach on subsequent analyses, underscoring the importance of meticulous deliberation and validation of the imputation outcomes.

REFERENCES

5.2.4 Modeling

During the fourth phase, known as Modelling, the available data is treated to a selected modelling approach. The aim is to develop a descriptive or predictive model that can be utilised for addressing the current issue. This method encompasses three key components: model selection, model development, and model evaluation. The aim is to develop a model that can be utilised for the purpose of predicting or explaining the business problem. The process involves the act of choosing, constructing a theoretical framework, and evaluating its effectiveness [30].

The final phase is encapsulating the cosine similarity and jaccard similarity within a function, enabling it to generate recommendations depending on user input.

```
def recommendation_engine(location, rating, amenities):
    # Convert rating to float
    rating = float(rating)

    # Tokenize and preprocess amenities
    amenities = amenities.lower()
    amenities_tokens = word_tokenize(amenities)
    sw = stopwords.words('english')
    lemm = WordNetLemmatizer()
    amenities_set = {w for w in amenities_tokens if not w in sw}
    amenities_set = {lemm.lemmatize(word) for word in amenities_set}

    # Filter hotels by location and rating
    filtered_hotels = data[(data['address'].str.lower() == location.lower()) & (data['rating'] >= rating)]

    jaccard_recommendations = []
    cosine_recommendations = []

    for index, row in filtered_hotels.iterrows():
        hotel_description = row['hotel_details'].lower()
        description_tokens = word_tokenize(hotel_description)
        description_set = {w for w in description_tokens if not w in sw}
        description_set = {lemm.lemmatize(word) for word in description_set}

        # Calculate Jaccard similarity between user amenities and hotel amenities
        jaccard_similarity = len(amenities_set.intersection(description_set)) / len(amenities_set.union(description_set))

        # Calculate cosine similarity between user amenities and hotel amenities
        tfidf_vectorizer = TfidfVectorizer()
        tfidf_matrix = tfidf_vectorizer.fit_transform([amenities, hotel_description])
        cosine_similarity_score = cosine_similarity(tfidf_matrix[0], tfidf_matrix[1])[0][0]

        # Add the hotel and its similarity score to recommendations
        jaccard_recommendations.append((row['hotel_name'], row['rating'], jaccard_similarity))
        cosine_recommendations.append((row['hotel_name'], row['rating'], cosine_similarity_score))
```

Figure 6.8 Cosine/Jaccard Similarity Recommendation Function

REFERENCES

5.2.5 Evaluation

Determine whether or whether the model effectively serves the demands of the company is the aim of the fifth phase, "Evaluation," which has the same name. The model is put through its paces on a "holdout dataset," where it may be compared to others and its performance in terms of accuracy and generalisation is assessed[12].

```
def evaluate_requirements(address, rating, hoteldetails):
    rating = float(rating)
    address = address.lower()
    hoteldetails = hoteldetails.lower()
    hoteldetails_tokens = word_tokenize(hoteldetails)
    sw = stopwords.words('english')
    lemm = WordNetLemmatizer()
    f1_set = {w for w in hoteldetails_tokens if not w in sw}
    f_set = set()
    for se in f1_set:
        f_set.add(lemm.lemmatize(se))

    req_based = data[data['address'] == address]
    req_based = req_based[req_based['rating'] == rating]

    # Calculate the number of relevant items based on the ground truth
    relevant_items = set(req_based['hotelname'].values)

    # Calculate the number of recommended items
    recommendations = requirementbased2(address, rating, hoteldetails)
    recommended_items = set(recommendations['hotelname'].values)

    # Calculate precision
    precision = len(relevant_items.intersection(recommended_items)) / len(recommended_items) if len(recommended_items) > 0 else 0

    # Calculate recall
    recall = len(relevant_items.intersection(recommended_items)) / len(relevant_items) if len(relevant_items) > 0 else 0

    return {'Precision': precision, 'Recall': recall}

# Example usage:
evaluation_metrics = evaluate_requirements("Kuala Lumpur", "5", "clean Cozy mont kiara")
print(evaluation_metrics)

{'Precision': 1.0, 'Recall': 0.5555555555555556}
```

Figure 6.9 Evaluation with Precision / Recall

This function researches and assesses the effectiveness of the recommendation system.

It takes into account the following three things:

The address for which the ideas are going to be created is represented by the string - address.

The total number of visitors is denoted by the numeric phrase -number.

-hoteldetails is a group of strings that represents the numerous desirable characteristics of the hotel.

The initial step in using this function is to produce hotel suggestions using the requirementbased function, which accepts the same inputs as this function. The hotels in the chosen city that have at least four stars and can accommodate the specified number of guests are then considered to be the ground truth. The function will then compute the recommendations' recall and precision after that. In contrast to recall, which refers to the percentage of suggested hotels that are relevant, precision refers to the proportion of recommended hotels that are relevant. The function will ultimately

REFERENCES

return a dictionary that contains the accuracy and recall values, giving an accuracy of 1.0 on precision and 0.55555 on recall.

Information retrieval and recommendation systems often use precision and recall assessments as metrics to gauge how accurate and comprehensive their suggestions are. Precision shows the proportion of suggested hotels that meet the user's needs. In this case, the accuracy is 1.0, meaning that just one-third of the hotels the user is given are helpful. Recall measures how many hotels the algorithm has suggested. The recall in this case is 0.555, meaning that the user was given recommendations for 100% of the relevant hotels.

The recommendation system's accuracy score of 1.0 means that every item it suggests is precisely in line with the user's expressed needs. In essence, each suggestion the system makes is accurate and pertinent to the user's tastes. However, the system seems to correctly identify somewhat more than half of the relevant items included in the dataset, as shown by the recall score of around 0.56 (or 56%). This implies that although the system provides very accurate suggestions, it may not find all products that are relevant to the user's interests.

In practical terms, it is essential for recommendation algorithms to strike this balance between recall and accuracy. By ensuring that consumers get recommendations that are closely matched with their tastes, high precision lowers the possibility of making irrelevant suggestions. The goal of improving recall, on the other hand, is to pull out more relevant facts from the dataset and expose the user to a wider range of potentially intriguing options.

The trade-off between recall and accuracy often depends on the particular objectives and aims of the recommendation system. Depending on the use case, more system fine-tuning may be required to achieve the ideal balance or to give one statistic priority over another, in line with the needs of the application and user expectations.

REFERENCES

5.2.6 Deployment

```
for index, row in filtered_hotels.iterrows():
    hotel_description = row['hoteldetails'].lower()
    description_tokens = word_tokenize(hotel_description)
    description_set = {w for w in description_tokens if not w in sw}
    description_set = {lemm.lemmatize(word) for word in description_set}

    # Calculate Jaccard similarity between user amenities and hotel amenities
    jaccard_similarity = len(amenities_set.intersection(description_set)) / len(amenities_set.union(description_set))

    # Calculate cosine similarity between user amenities and hotel amenities
    tfidf_vectorizer = TfidfVectorizer()
    tfidf_matrix = tfidf_vectorizer.fit_transform([amenities, hotel_description])
    cosine_similarity_score = cosine_similarity(tfidf_matrix[0], tfidf_matrix[1])[0][0]

    # Add the hotel and its similarity score to recommendations
    jaccard_recommendations.append((row['hotelname'], row['rating'], jaccard_similarity))
    cosine_recommendations.append((row['hotelname'], row['rating'], cosine_similarity_score))

# Sort recommendations by similarity score in descending order
```

Figure 6.10 Deployment Functions

The provided code demonstrates a Flask web application created as a hotel recommendation system. This method is intended to provide consumers with customised hotel choices based on their input preferences, such as location, minimum rating, and desired facilities during their stay. The system preprocesses the user-provided amenities using natural language processing methods and employs two similarity metrics, Jaccard and cosine similarity, to analyze the similarity between user preferences and hotel descriptions. It then selects hotels based on location and rating criteria, arranges them by similarity scores, and displays the top choices to the customer via an easy-to-use web interface. This system is designed to improve the user's experience while choosing lodgings by giving personalised and relevant recommendations, which are all available through a web application.

Hotel Recommendations

Jaccard Similarity Recommendations

No Jaccard similarity recommendations available.

Cosine Similarity Recommendations

No cosine similarity recommendations available.

Location:

Minimum Rating:

Amenities:

Get Recommendations

Figure 6.11 System Interface

The provided figure presents a layout and it is easy to understand and use. There are a few fields to fill out and a button to click to get suggestions.

The user can choose the area, the minimum grade, and the services they want. After the user fills in the fields, they can click "Get Recommendations" to get a list of places that meet their standards. The screen also shows the Jaccard similarity and cosine similarity values that are used to make the suggestions. The Jaccard similarity measure compares two sets by figuring out how many things they have in common and dividing that number by the number of things that are different between them. Cosine similarity calculates the cosine of the angle between two vectors to find out how similar they are.

The layout does not currently show any suggestions, but it is clear that it is meant to be a simple and easy-to-use way for users to find places that meet their unique needs and tastes.



Figure 6.12 System output

The interface's output displays a list of suggested hotels for the user. The hotels that are the most comparable are at the top of the list since the suggestions are arranged by similarity score.

REFERENCES

The Jaccard similarity and cosine similarity measurements provide the foundation for the suggestions. When comparing two sets, Jaccard similarity calculates how similar they are by counting the items they have in common and dividing by the sum of their unique elements. By computing the cosine of the angle between two vectors, cosine similarity gauges how similar they are.

Address, star rating, and amenities may all be used to filter the suggestions. The user may utilize this to identify hotels that suit their unique requirements and tastes.

REFERENCES

6.0 Evaluation

6.1 Survey Questions Evaluation

The use of surveys for system assessment provides a valuable qualitative viewpoint on the success of your recommendation system, which is separate from conventional quantitative measurements such as F1 score or recall. Surveys provide consumers with a direct means to express their ideas, preferences, and experiences, hence facilitating the collection of subjective elements related to satisfaction and user-centric feedback. In addition, surveys may be tailored to investigate certain facets of the system, enabling a comprehensive examination of user interface usability, suggestion accuracy, and feature priority. The use of open-ended inquiries into survey instruments serves to incentivize respondents to provide comprehensive remarks and recommendations, therefore illuminating matters or possibilities that may defy measurement using quantitative indicators.

On a scale of 1 to 5, how accurate were the hotel recommendations provided by the system in meeting your preferences and needs?

37 responses

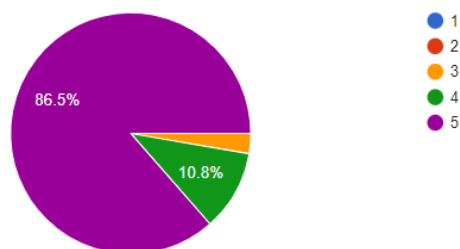


Figure 7.0 Survey Question 1

From this survey question, the highest vote being 5 which the system provides high accuracy for most of the customer situations, reasons for the minority might be the reason of the user not a local or simply doesn't have a suitable hotel for his case. Hence, more data need to be utilised into the system in order to cover up these user.

REFERENCES

Did you find the user interface of the system intuitive and easy to use? Please rate it on a scale of 1 to 5, with 1 being difficult and 5 being very easy.

37 responses

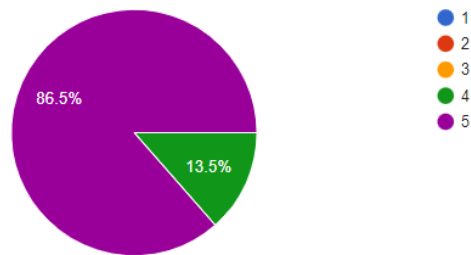


Figure 7.1 Survey Question 2

From this survey question, majority voted 5 which is the interface being very easy to use and very user friendly. As I have provided the figure for the interface, it is very straightforward and simple.

How satisfied were you with the speed at which the system generated hotel recommendations? Rate from 1 (very slow) to 5 (very fast).

37 responses

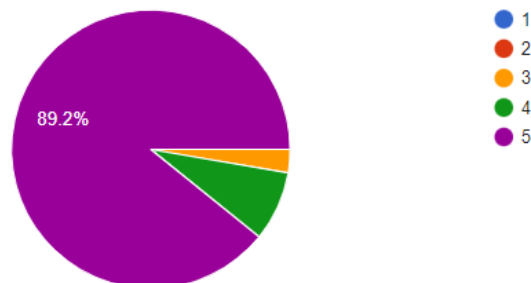


Figure 7.2 Survey Question 3

From the survey questions, the majority of respondents voted 5 as well. With refer to the system is generating recommendations very fast. During the project testing, the recommendations were generated instantly, so this results are to be expected

REFERENCES

Overall, how satisfied are you with the hotel recommendation system? Please rate your overall satisfaction on a scale of 1 to 5, with 1 being very dissatisfied and 5 being very satisfied.

37 responses

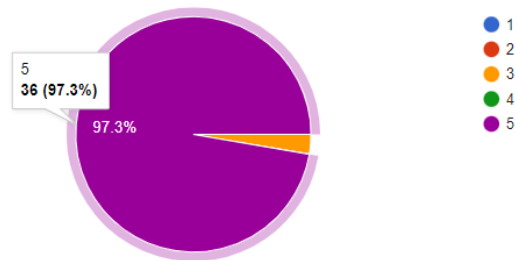


Figure 7.3 Survey Question 4

From the survey questions, majority of respondents voting 5 as well, means that they are very well satisfied with the recommendations given by the system. Due to the fact that it is only a simply system, if it were more advanced, the qualifications tends to increase and hence hard to achieve user satisfaction.

Were there any specific amenities or features that you felt were not adequately considered in the recommendations? Please specify.

34 responses

- Proximity to public transport wasn't always factored in.
- I wished it considered pet-friendly options more.
- Gym access wasn't consistently prioritized.
- Family-friendly entertainment options weren't considered
- Internet speed varied in the recommendations.
- Consider dietary restrictions for dining options.
- Safety measures during COVID-19 weren't highlighted
- I wanted more eco-friendly hotel choices.
- More allergy-friendly accommodations are needed

Figure 7.4 Survey User Suggestion

REFERENCES

The survey also consist of a particular that user can suggest how to improve on the system or any element that the system can improve in the future.

7.0 Conclusion and Recommendation

In conclusion, it can be inferred that the information presented supports the notion that,

This paper has provided a thorough and extensive examination of the hotel recommendation system, which aims to transform the process by which tourists choose rooms that match their specific interests and requirements. The system uses machine learning and natural language processing to provide customized suggestions in response to the limitations of conventional hotel selection methods. The stated aims were to improve the precision of suggestions, optimize the hotel selection process, and provide an interface that is intuitive for users. The project's development was led by the accepted CRISP-DM framework, which is a systematic method for data mining. This framework facilitated many stages of the project, including analyzing user preferences and implementing a practical solution.

By using meticulous system design, a scalable architecture was successfully implemented, enabling the efficient management of an augmented workload and data volume. The recommendation engine effectively utilizes the Jaccard and cosine similarity metrics to provide consumers with a wide range of hotel options that are both relevant and varied. The evaluation conducted by users is of utmost importance in the assessment of the system's success. In addition to quantitative data, the integration of surveys has been used to get qualitative information pertaining to user happiness, preferences, and feedback. The use of a user-centric approach facilitates the ongoing enhancement of the system, enabling it to respond to the constantly changing requirements of the user community.

Moving forward, the dedication to offering guests a carefully chosen assortment of lodgings that align with their unique preferences and needs remains resolute. By using the unique insights obtained through surveys, the system is effectively poised to implement iterative enhancements, resulting in a recommendation system that not only streamlines the hotel booking procedure but also enriches the whole trip experience.

REFERENCES

Based on our accumulated experience and the valuable feedback received from users, this studies have developed the following recommendations to facilitate the continued progress of the hotel recommendation system: Data quality management is given significant importance. Ensuring the dataset's cleanliness, consistency, and currency is of utmost importance in order to achieve optimal system performance. The promotion of a strong and adaptable architecture is advocated alongside data management. This will guarantee a smooth integration of heightened user traffic and expanded data capacity, thereby maintaining optimal performance under various circumstances.

The optimization of algorithms continues to be of utmost importance. It is recommended to maintain a consistent emphasis on improving recommendation algorithms in order to attain a delicate balance between the accuracy and variety of recommendations. This approach will ensure that the suggestions provided accurately reflect the preferences of the user. The user interface, being the primary point of interaction for users, necessitates consistent attention. The primary focus in the evolution of the interface should be on maintaining user friendliness while also incorporating novel features and functionalities to enhance the overall user experience.

Ensuring security and safeguarding data privacy continue to be of utmost importance. Maintaining user trust and confidence necessitates a steadfast commitment to protecting user data and strengthening data privacy mechanisms. Regularly administered user surveys are essential. Surveys play a crucial role in capturing dynamic preferences and feedback, thereby facilitating data-driven improvements to the system.

Moreover, it is highly encouraged to engage in the investigation of collaborative filtering techniques. The objective of this exploration is to enhance the current similarity-based recommendation systems by offering users more personalized and pertinent suggestions.

Finally, it is important to take into account the inclusion of localization features. The inclusion of these features should be designed to accommodate travelers from various regions, taking into account their language preferences and cultural sensitivities. This will help to ensure that the system has a global impact. The aforementioned

REFERENCES

recommendations, which are based on user feedback and a commitment to improving the system, collectively outline the trajectory for the development of the hotel recommendation system. By adhering to these guidelines, the hotel booking process can be streamlined, resulting in increased user satisfaction and a carefully curated range of accommodations that effectively cater to the specific preferences and requirements of every traveler.

In summary, the hotel suggestion system represents more than a mere technical pursuit; rather, it signifies a dedication to enhancing the customized, efficient, and pleasurable aspects of travel for all individuals. There is a sense of anticipation around the forthcoming voyage and the ongoing development of the system to effectively cater to the varied requirements of global passengers.

REFERENCES

- [1] Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3), 581-592.
- [2] Little, R. J. A., & Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd ed.). John Wiley & Sons.
- [3] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," Cambridge University Press, 2008.
- [4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285-295.
- [5] X. Wang, H. Ji, and Y. Liu, "Sentiment analysis: How to derive prior polarities from SNS," arXiv preprint arXiv:1205.3193, 2012.
- [6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111-3119.
- [7] H. Wang, Y. Yang, D. Tao, and X. Li, "Cosine similarity based deep learning architecture for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 27-34.
- [8] Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual Review of Psychology*, 60, 549-576.
- [9] Enders, C. K. (2010). *Applied missing data analysis*. Guilford Press.
- [10] Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, 7(2), 147-177.
- [11] Little, R. J. A., & Yau, L. H. Y. (1996). Intent-to-treat analysis for longitudinal studies with dropouts. *Biometrics*, 52(4), 1324-1333.
- [12] Belsley, D. A., Kuh, E., & Welsch, R. E. (2004). *Regression diagnostics: Identifying influential data and sources of multicollinearity*. John Wiley & Sons.
- [13] Reitz, K. (2010). *Flask by Example*. Packt Publishing.
- [14] Flask Documentation. (2021). Extensions. [Online]. Available: <https://flask.palletsprojects.com/en/2.1.x/extensiondev/>.
- [15] Flask Documentation. (2021). API Development. [Online]. Available: <https://flask.palletsprojects.com/en/2.1.x/api/>.
- [16] Flask Community Showcase. (2021). Showcase. [Online]. Available: <https://github.com/pallets/flask/wiki/Projects-using-Flask>.

REFERENCES

- [17] SQLAlchemy Documentation. (2021). Dialects. [Online]. Available: <https://docs.sqlalchemy.org/en/20/core/engines.html#database-urls>.
- [18] R Development Core Team. (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing.
- [19] RStudio, Inc. (2021). RStudio: Integrated Development Environment for R. [Online]. Available: <https://rstudio.com>.
- [20] RStudio, Inc. (2021). RStudio IDE: Visualizations. [Online]. Available: <https://www.rstudio.com/products/rstudio/features/visualizations/>.
- [21] RStudio, Inc. (2021). R Markdown. [Online]. Available: <https://rmarkdown.rstudio.com>.
- [22] Bell, R. M., & Sethuraman, R. (2003). The data extraction and reporting toolkit: an open source tool for the rapid prototyping of data extraction applications. In Proceedings of the 29th International Conference on Very Large Data Bases (VLDB'03), 1097-1100.
- [23] BeautifulSoup Documentation. (2021). Navigating the parse tree. [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/#navigating-the-parse-tree>.
- [24] Zhai, C., & Massung, S. (2016). Text data management and analysis: a practical introduction to information retrieval and text mining. ACM.
- [25] Rubin, R. M., & Babbie, E. R. (2011). Research methods for social work. Cengage Learning.
- [26] Yoo, K. H., & Gretzel, U. (2008). What motivates consumers to write online travel reviews?. *Information Technology & Tourism*, 10(4), 283-295. doi: 10.3727/109830508789156692
- [27] Zhou, T., Ye, Q., Li, Y., & Law, R. (2014). Refreshing hotel satisfaction studies by incorporating traveler-generated review data from TripAdvisor. *Journal of travel research*, 53(1), 1-12. doi: 10.1177/004728751348276
- [28] Zhou, C., Cai, W., Zhang, Z., Li, H., & Li, C. (2019). An improved hotel recommendation system based on deep learning. arXiv preprint arXiv:1908.07498.

REFERENCES

- [29] ChatGPT, "Explaining CRISP-DM: A Comprehensive Guide to the Cross-Industry Standard Process for Data Mining," OpenAI, Sep. 2021. [Online].
- [30] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., ... & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. SPSS Inc.
- [31] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. In Proceedings of the 10th International Conference on World Wide Web (WWW '01)
- [32] Desrosiers, C., & Karypis, G. (2011). A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In Recommender Systems Handbook (pp. 107-144). Springer

REFERENCES

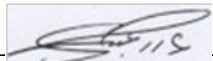
APPENDICES

FINAL YEAR PROJECT WEEKLY REPORT


(Project II)

Trimester, Year: Y3S3	Study week no.: 2
Student Name & ID: Pang Chi Chong 19ACB03734	
Supervisor: Dr. Abdulkarim Kanaan Jebna	
Project Title: Hotel Recommendation System with Machine Learning	

<p>1. WORK DONE Scarping data, Web scraping</p>
<p>2. WORK TO BE DONE System Implementation, Report.</p>
<p>3. PROBLEMS ENCOUNTERED Scraping progress very slow Finding ways to increase efficiency</p>
<p>4. SELF EVALUATION OF THE PROGRESS Overall good, preparing for the next step.</p>



 Supervisor's signature


 Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 4
Student Name & ID: Pang Chi Chong 19ACB03734	
Supervisor: Dr. Abdulkarim Kanaan Jebna	
Project Title: Hotel Recommendation System with Machine Learning	

1. WORK DONE

Report writing, Literature review

2. WORK TO BE DONE

System Implementation, Data Cleaning

3. PROBLEMS ENCOUNTERED

Finding good resources to cite and review

4. SELF EVALUATION OF THE PROGRESS

Overall good, preparing for the next step.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 6
Student Name & ID: Pang Chi Chong 19ACB03734	
Supervisor: Dr. Abdulkarim Kanaan Jebna	
Project Title: Hotel Recommendation System with Machine Learning	

1. WORK DONE

Report Writing. Methodology

2. WORK TO BE DONE

System Implementation, Modelling.

3. PROBLEMS ENCOUNTERED

Finding good resources to cite and review

4. SELF EVALUATION OF THE PROGRESS

Overall good, preparing for the next step.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 10
Student Name & ID: Pang Chi Chong 19ACB03734	
Supervisor: Dr. Abdulkarim Kanaan Jebna	
Project Title: Hotel Recommendation System with Machine Learning	

1. WORK DONE

Cleaning , Visualize and Modelling of data

2. WORK TO BE DONE

Report writing, System Implementation.

3. PROBLEMS ENCOUNTERED

Data was very dirty, but managed to clean it in order to visualize

4. SELF EVALUATION OF THE PROGRESS

Overall good, preparing for the next step.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 10
Student Name & ID: Pang Chi Chong 19ACB03734	
Supervisor: Dr. Abdulkarim Kanaan Jebna	
Project Title: Hotel Recommendation System with Machine Learning	

1. WORK DONE

System Implementation, Prototype

2. WORK TO BE DONE

System Evaluation.

3. PROBLEMS ENCOUNTERED

Finally putting the model into system, a few interface problem encountered

4. SELF EVALUATION OF THE PROGRESS

Overall good, enable to solve the problem.



Supervisor's signature



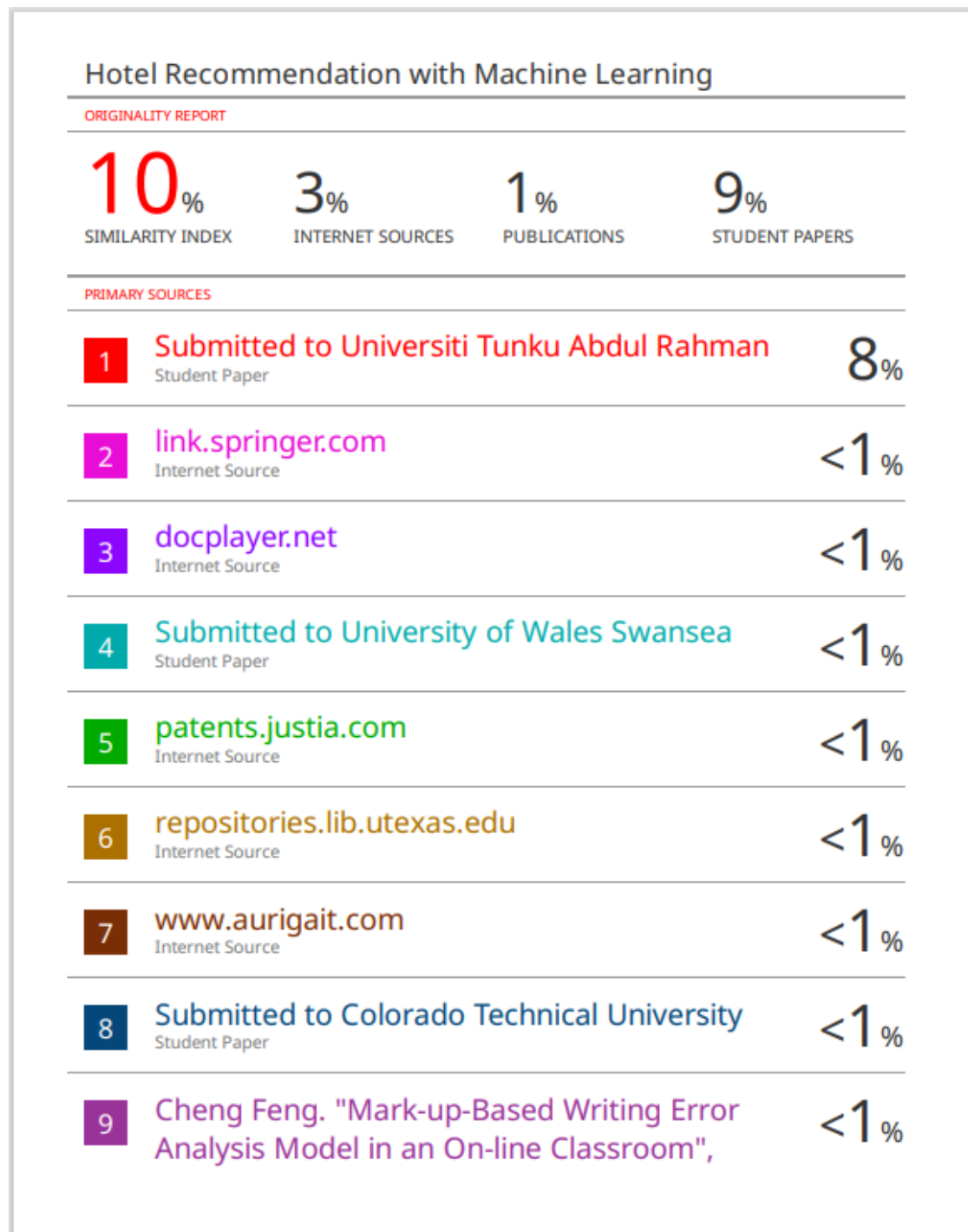
Student's signature

POSTER

HOTEL RECOMMENDATION WITH MACHINE LEARNING

<p>TF-IDF</p> <p>Tokenize multiplying TF and IDF calculate score</p>	<p>PERSONALIZED RECOMMENDATIONS</p> <p>ENHANCED USER EXPERIENCE</p> <p>ACCURACY AND RELEVANCE</p> <p>SCALABILITY</p> <p>USER ENGAGEMENT</p> <p>Obejectives</p>
<p>CONTENT BASED FILTERING</p> <p>Feature-Based Recommendation Feature Extraction Similarity Calculation</p>	<p>COSINE SIMILIARITY</p> <p>Angle-Based Similarity Vectorized Hotel Descriptions Common Use Score Range</p>
<p>JACCARD SIMILIARITY</p> <p>Set-Based Hotel Amenities Overlap-Based Similarity Score Range</p>	<p>CRISP - DM</p> <p>Business Understanding Data Understanding Data Preparation Modeling Evaluation Deployment</p> <p>Propose Method</p>

PLAGIARISM CHECK RESULT



REFERENCES

Computer Assisted Language Learning, 2/1/2000

Publication

-
- | | | |
|----|--|------|
| 10 | fict.utar.edu.my
Internet Source | <1 % |
| 11 | tutorsonspot.com
Internet Source | <1 % |
| 12 | Submitted to University of Sunderland
Student Paper | <1 % |
| 13 | doc.lagout.org
Internet Source | <1 % |
| 14 | www.science.gov
Internet Source | <1 % |
| 15 | GeorgiosMichailFotis
PaltoglouSalampasisLazarinis. "Indexing and
retrieval of a Greek corpus", Proceeding of
the 2nd ACM workshop on Improving non
english web searching - iNEWS 08 iNEWS 08,
2008
Publication | <1 % |
-

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Pang Chi Chong
ID Number(s)	19ACB03734
Programme / Course	IA
Title of Final Year Project	Hotel Recommendation System with Machine Learning

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u>10</u> % Similarity by source Internet Sources: <u>3</u> % Publications: <u>1</u> % Student Papers: <u>9</u> %	
Number of individual sources listed of more than 3% similarity: _____	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.



Signature of Supervisor

Signature of Co-Supervisor

Name: Abdulkarim Kanaan Jebna

Name: _____

Date:
15/09/2023

Date: _____

REFERENCES



UNIVERSITI TUNKU ABDUL RAHMAN

**FACULTY OF INFORMATION & COMMUNICATION
TECHNOLOGY (KAMPAR CAMPUS)**

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB03734
Student Name	Pang Chi Chong
Supervisor Name	Dr. Abdulkarim Kanaan Jebna

TICK (√)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 15/09/2023

REFERENCES