2023

Travel agency recommender system based on social media sentiment analysis.

King'ori, Stephanie Wambaire
School of Computing and Engineering Sciences
Strathmore University

Travel Agency Recommender System Based on Social Media Sentiment Analysis

By

Stephanie Wambaire King'ori

124383

Master of Science in Information Technology

2023

Travel Agency Recommender System Based on Social Media Sentiment Analysis

By

Stephanie Wambaire Kingori

124383

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology at Strathmore University

School of Computing & Engineering Sciences

Strathmore University

Nairobi, Kenya

July, 2023

Declaration and Approval

Dedication

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Student's Name: Stephanie Wambaire King'ori

Sign: …… **Date: ……16/06/2023…….**

Approval

The thesis of Stephanie Wambaire Kingori was reviewed and approved for examination by the following:

Dr Henry Muchiri
School of Computing & Engineering Sciences,
Strathmore University

Dr. Julius Butime,
Dean, School of Computing & Engineering Sciences,
Strathmore University

Dr. Bernard Shibwabo,
Director of Graduate Studies,
Strathmore University

Abstract

In today's highly competitive e-tourism industry, online reviews and recommendations are crucial to customers' travel decisions. This study focuses on reviewing the level of service quality in the e-tourism sector through social media sentiment analysis, aiming to aid travelers in making informed decisions about their travel arrangements through a recommender system. While other methods exist for recommender systems, they are not sufficient for the specific context of Kenya. To address this, the researcher develops a tool capable of providing personalized recommendations based on user destinations using the transformed data received from sentiment analysis. The study adopts an exploratory research design, targeting individuals who engage in social media discussions related to the e-tourism industry, including travelers, travel companies, and other tourists which is supported by Agile Development Methodology.

The collected data is retrieved from Twitter, a prominent communication platform for travel agencies. The study utilizes SnScrape for tweet extraction and employs data preprocessing techniques to categorize and analyze the collected data using the Vader lexicon model. The collected data undergoes sentiment analysis, where each evaluated tweet is assigned a polarity tag indicating whether it is positive, negative, or neutral sentiment, with the analyzed results presented using charts and tables. Machine learning algorithms play a crucial role in providing personalized and relevant recommendations to users based on their destination preferences and historical data. K-Nearest neighbor, Support Vector Machine, and Naive Bayes are explored and evaluated to show the best-performing algorithm for the recommending system. A hybrid filtering approach is incorporated using content-based and collaborative filtering to create travel profiles based on the selected Reliability and validity measures are applied to ensure research quality, e research quality, both reliability and validity measures are applied which exhibit high levels of accuracy, precision, and F1 scores indicating their effectiveness in recommending travel agencies.

Streamlit is used to build an interface and deploy the machine learning models using a set of rules to recommend travel agencies based on the user's destination. The study concludes that recommender systems rely on feedback and online reviews shared by customers after traveling to various destinations. It recommends that travel providers acknowledge this trend and actively encourage customers to share their experiences through social media and other platforms. Additionally, the study suggests that users of sentiment analysis tools ensure diverse training data to mitigate bias and accurately reflect the sentiment of the target audience.

Keywords: Service Quality, Sentiment Analysis, Travel Agencies, Recommender System, Twitter, Machine learning

TABLE OF CONTENTS

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
API	Application of Programming Interface
DFD	Data Flow Diagram
ICT	Information Communication Technology
NLP	Natural Language Processing
RAD	Rapid Application Development
SEM	Structural Equation Model
SVM	Support Vector Machine
US	United States
WEF	World Economic Forum
WTO	World Tourism Organization
WTTC	World Travel and Tourism Council

Acknowledgements

I would like to express my deepest gratitude and appreciation to all those who have supported me throughout the journey of completing this thesis. Their guidance, encouragement, and contributions have been invaluable in shaping the success of this research endeavor.

First and foremost, I extend my heartfelt thanks to my supervisor Dr Henry Muchiri and thesis lecturer Prof Ismail Ateya. Their insightful feedback, constructive criticism, and continuous encouragement have been instrumental in shaping the direction and quality of this research. I am truly grateful for their mentorship and belief in my capabilities.

I would also like to acknowledge the support and encouragement from my friends and family. Their understanding, patience, and unwavering belief in my abilities have been a constant source of motivation throughout this journey. Their presence and support have provided me with the strength to overcome challenges and persevere in my academic pursuits.

To all those who have directly or indirectly contributed to the completion of this thesis, I am truly grateful for your support and encouragement. Your contributions have played a significant role in the successful culmination of this research endeavor. Thank you for being an integral part of this

Definition of Terms

E-Tourism	Electronic tourism is a discipline that combines three disparate disciplines, namely, business management, information and management and tourism (Kamuzora, 2016)
Machine learning	Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and models that enable computers to learn and improve their performance based on data (Kamel et al., 2018).
Sentiment analysis	Also known as opinion mining, this is a machine learning concept that involves using natural language processing (NLP) and machine learning techniques to analyze and classify the sentiment of text data (Hutto & Gilbert, 2014).
Service Quality	Service quality refers to the degree to which a service meets or exceeds customers' expectations (Gronroos, 2004)

Chapter 1: Introduction

1.1 Background

Information technology (IT) has been extensively employed by the tourism industry to increase operational effectiveness, boost consumer satisfaction, and improve service quality (Law, Leung and Buhalis, 2009). Several applications such as Airbnb, Booking, TripAdvisor have offered a platform for tourist suppliers to expand their customer base and end-consumers to satisfy their drive for traveling and discovering new places. The ideas and structures of the tourism business have steadily evolved because of internet use. (Guttentag, 2018).

Experiences are driving the travel industry and influencing the consumers' decisions which makes the travel product more and more diverse. Travel providers serve as an intermediary between travel suppliers such as transport providers, hotels and consumers (Cheyne, Downes and Legg, 2016). Travel is revolutionized as the travel providers have a huge benefit of connecting travelers by analyzing their reviews and the services offered. Strong relationships between the travel providers and consumers are directly motivated and enable increase in customer satisfaction, service quality and loyalty as the overall service is more effectively coordinated. (Yagil, 2008).

The travel and tourism industry has experienced significant growth in recent years, fueled by globalization, technological advancements, and the widespread availability of online information (Mamaghani, F., 2009).As a result, travelers are faced with an overwhelming number of options when selecting travel agencies to meet their specific needs and preferences. To address this challenge, recommender systems have emerged as valuable tools to assist travelers in making informed decisions by providing personalized recommendations. In the context of the travel industry, incorporating social media sentiment analysis into recommender systems holds great promise for improving the accuracy and relevance of recommendations.

Recommender systems are algorithms that leverage user preferences and historical data to generate tailored suggestions (Konstan, Joseph & Riedl, John, 2012). By integrating social media sentiment analysis, recommender systems can tap into the vast amount of user-generated content on social media platforms, such as Twitter, Facebook, and Instagram. Social media platforms have become key channels for travelers to share their experiences, opinions, and sentiments regarding travel agencies (Leung, D., Law, R., Van Hoof, H., & Buhalis, D.,2013). By analyzing the sentiment

expressed in this content, recommender systems can gain valuable insights into customer preferences, satisfaction levels, and overall sentiment towards specific travel agencies. The rapid development of information and communication technologies has changed how businesses operate and how enterprises compete (Porter, 2011).

In recent years, the literature on tourism and leisure has given more and more emphasis to the problem of evaluating service quality. To assess the quality of digital platforms and service perceptions, several conceptual frameworks have been established. The SERVQUAL is one of the instruments that has been proposed for the measurement of perceived service quality within a wide range of service categories. Another feature is the E-travel Service Quality Scale that has been developed to measure and evaluate service quality, customer satisfaction and behavioral loyalty. The E-Satisfaction Model was developed back in 2006 by Masoomeh to measure the electronic satisfaction in the tourism industry. The model has five aspects: Convenience, Product Offering, Product Information, Site design and financial security (Tasci, William and Gartner, 2007).

A thorough review of the existing literature on recommender systems, sentiment analysis, and the travel industry will be conducted. This review will identify gaps and opportunities in the field and inform the development of the proposed recommender system. The research will adopt a mixed-methods approach, combining quantitative analysis of social media data using sentiment analysis algorithms to gather deeper insights into traveler preferences and experiences.

Various Artificial Intelligence technologies are analyzed in the tourism industry and the potential uses are thoroughly explained in customer profiling, internet marketing and customer profile management. AI is highly effective as it draws conclusions from vast information using techniques such as machine learning, neural networks, robotics, expert systems, and natural language processing. In this study, we will analyze the use of machine learning in facilitating service quality of travel providers. Machine learning will aim at showing how globalization has changed tourist consumer behavior as it has the capacity to create impacts on cultural criteria (culture), social criteria (reference groups), personal criteria (occupation, economic circumstances), psychological criteria (beliefs and attitudes). The e-tourism industry succeeds and grows if it identifies the consumer demands and satisfies their needs.

In this study, we will design and develop a travel agency recommender system that incorporates social media sentiment analysis. The recommender system will utilize sentiment analysis techniques to extract sentiments from social media data and generate personalized recommendations for travelers. By considering the sentiment expressed by users, the system aims to provide more accurate and relevant recommendations that align with individual preferences.

1.2 Problem Statement

The travel and tourism industry are characterized by a vast number of travel agencies offering a wide range of services to cater to the diverse needs and preferences of travelers. However, the abundance of options poses a significant challenge for travelers in selecting the most suitable travel agency that aligns with their specific requirements. While recommender systems have emerged as valuable tools for assisting travelers in decision-making, existing approaches often overlook the wealth of sentiment and opinions expressed on social media platforms, which can provide valuable insights into customer preferences and overall satisfaction.

Although social media platforms, such as Twitter, Facebook, and Instagram, have become prominent channels for travelers to share their experiences and opinions, current travel agency recommender systems do not fully exploit the potential of social media sentiment analysis. The lack of integration of sentiment analysis techniques into recommender systems limits their ability to generate accurate and personalized recommendations that account for customer sentiments and preferences.

The problem at hand is the absence of an effective travel agency recommender system that harnesses the power of social media sentiment analysis to provide travelers with personalized recommendations. By leveraging sentiment analysis techniques, it is possible to analyze the sentiments expressed in user-generated content on social media platforms and gain insights into customer satisfaction levels, preferences, and overall sentiment towards specific travel agencies. However, there is a gap in the existing research on the development of a robust travel agency recommender system that incorporates social media sentiment analysis.

Additionally, the integration of sentiment analysis into recommender systems poses challenges related to data collection, preprocessing, and algorithm design. The vast amount of social media data and the complexity of sentiment analysis algorithms require careful consideration and

customization to ensure accurate sentiment classification and relevant recommendations. Furthermore, addressing potential biases and ensuring the generalizability and scalability of the system present additional challenges in the development process.

There are various existing and currently used techniques for identifying appropriate travel agencies. These include Online Travel Agencies, Travel Review Websites, Local Tourism Boards and Associations, Travel Blogs and Forums and the traditional Local Guides and Tour Operators (Kim et al., 2021). However, these techniques suffer from various weaknesses that limit their effectiveness in providing accurate and personalized recommendations. Yagil (2018) noted that current techniques often lack personalized recommendations, providing generic suggestions that do not consider individual user preferences, travel styles, and specific requirements. Xie et al (2021) also determined that many existing approaches overlook the valuable insights available from social media sentiment analysis failing to leverage user-generated content to capture the positive or negative experiences shared by travelers and their sentiments towards different travel agencies.

Additionally, Paraskevas & Leonidou (2018) found that current techniques may rely on a limited set of data sources, such as official websites or travel review platforms, without considering the extensive pool of opinions and experiences expressed on social media platforms. Hussain (2022) also revealed that existing techniques may suffer from inaccurate or outdated information, as they often rely on static databases or manual reviews that cannot keep up with the real-time changes in user sentiments and preferences. These limitations hinder the ability of users to make informed decisions when selecting travel agencies thereby creating an information gap. The study aims to bridge this gap by developing a recommender system that leverages sentiment analysis of social media data to provide more informed and personalized travel agency recommendations. By investigating and addressing this information gap, the study seeks to enhance the travel agency selection process, provide users with personalized recommendations, and improve overall customer satisfaction in the e-tourism industry.

1.3 Objectives

1.3.1 Main Objective

To design and develop an effective and personalized travel agency recommender system that leverages social media sentiment analysis to provide travelers with accurate and relevant recommendations.

1.3.2 Specific Objectives

1. Establish the influence of service quality on travel providers in the E-tourism industry.
2. To identify the key challenges and limitations in existing recommender systems.
3. To design and develop a travel agency recommender system that incorporates social media sentiment analysis.
4. To evaluate the performance and effectiveness of the developed recommender system.

1.4 Research Questions

1. What effect does service quality have on travel providers in the E-tourism industry?
2. Which are the key challenges facing the existing recommender systems?
3. How to design and develop a travel agency recommender system that incorporates social media sentiment analysis?
4. How to test the performance and effectiveness of the developer recommender system?

1.5 Justification

Recommender systems have emerged as valuable tools in various domains to address information overload problems and provide personalized recommendations. In the context of the travel industry, existing recommender systems primarily rely on historical user behavior and preferences, neglecting the vast amount of sentiment and opinions expressed on social media platforms. Incorporating social media sentiment analysis into travel agency recommender systems holds great potential for enhancing the accuracy and relevance of recommendations.

The proposed study fills a significant research gap in the field. While sentiment analysis has been widely applied in various domains, its integration into travel agency recommender systems remains limited. There is a lack of comprehensive research that explores the potential benefits,

challenges, and implications of integrating social media sentiment analysis into recommender systems in the travel industry.

The outcomes of this study have several implications. Firstly, the proposed travel agency recommender system can enhance the overall customer experience by providing more accurate and relevant recommendations, leading to higher satisfaction levels. Secondly, travel agencies can benefit from the insights gained through sentiment analysis, allowing them to improve their services and tailor their offerings to meet customer preferences. Thirdly, the research contributes to the broader field of recommender systems by exploring the integration of sentiment analysis in the travel context, offering insights into the challenges, opportunities, and best practices for incorporating sentiment analysis techniques into recommender systems.

The result of this project provides insights to the government as they come to the realization that the tourism industry contributes to the local economy via purchases, travel retail and expenditures thus they should focus on offering competitively priced offerings and up to standard amenities that appeal to the needs of the customer. Practitioners such as data analysts, the study will help them create an all-around representation of measures of service quality and analyze statistics based on customer feedback. Sustained interactions between academic research and the e-tourism industry creates more artificial intelligence opportunities and provides a platform for fusion of knowledge and reference to analyze how different elements influence service quality of travel providers in the e-tourism industry. More fundamental is the contribution of service quality to the socio-economic status in that it improves the local and global economy and determines the societal production of facilities that dominates the E-tourism industry.

1.6 Risk Benefit Analysis

Risk/benefit analysis is an important tool for evaluating the potential risks and benefits associated with a study(Schindler, 2009).

The benefits include:

- The development of a machine learning tool that can help travel agencies in the e-tourism industry to maintain high service quality, resulting in increased customer satisfaction, loyalty, and positive reviews.

- The use of sentiment analysis to identify and address common customer complaints and issues, allowing travel agencies to improve their services and better meet customer needs.
- The potential to gain a competitive advantage over other travel agencies in the market by offering a more personalized and high-quality service to customers.
- The ability to collect and analyze large amounts of customer feedback and data in a more efficient and systematic manner, leading to more informed decision-making and improved business practices.

The risks include:

- Privacy concerns related to the collection and use of personal data and online reviews of customers. These concerns could result in negative publicity or legal action against the travel agencies.
- The potential for inaccuracies or biases in the sentiment analysis algorithm, leading to incorrect or misleading recommendations to travel agencies.
- The cost and resources required to develop and implement the machine learning tool, which could be substantial.
- The risk that the machine learning tool may not be widely adopted or accepted by travel agencies or customers, resulting in limited impact or uptake.

The benefits of the study outweigh the risks, as the development of a machine learning tool can bring significant benefits to the e-tourism industry in terms of improved service quality and offer personalized recommendations.

1.7 Scope

The focus of the study will be on developing a travel agency recommender system that generates personalized recommendations based on the sentiment analysis of social media data. The system will consider individual traveler preferences and sentiments to provide accurate and relevant recommendations.

Travel agencies support various geographical areas and key destinations across the world, this study will concentrate on travelers within the Kenyan region as it considers domestic and luxury travelers. The study will primarily focus on social media platforms where users share their travel experiences, opinions, and sentiments regarding travel agencies. Common platforms such as Twitter, Facebook and Instagram are popularly used to gain feedback, the proposed research will be limited to Twitter, a popular social media platform, which will be considered for data collection and sentiment analysis.

The research will employ sentiment analysis techniques to extract sentiments from social media data. It will utilize the Vader Lexicon model and machine learning algorithms. Data collection process will involve techniques such as web scraping from Twitter.

The research results can also be used as a point of reference for various consumers and academic practitioners whose intention is to increase customer experience and loyalty.

Chapter 2: Literature Review

2.1 Introduction

The second chapter of the study comprised the empirical review sections, models sections, algorithm and application sections and finally the conceptual framework. The empirical section presented findings from a literature review on the study variables. These were arranged according to the study objectives.

2.2 Empirical Review

This section presents a critical review of studies relating to the study variables presented in line with the study objectives.

2.2.1 Influence of Service Quality on Travel Providers in the E-Tourism Industry

One of the most important metrics for assessing consumer satisfaction is service quality. The expectations of the consumer affect their happiness, choice of service provider, and assessment of the quality of the service (O'Connor, Trinh, and Shewchuk, 2010). Instead of what the provider puts in, service quality is defined as what the consumer receives and is prepared to pay for. (Ducker, 2021). The customer's first experience with the travel provider determines whether the customer will remain a visitor or not. Naz & Khan, (2015) noted that there is a need to place customers within the context of partnership rather than regarding them as the marketing target. This contributes to the customer becoming your visitor and increases the value that the customer has towards your brand as the travel provider. An organization that is customer-centric and caters for their needs tends to increase product/service and improve the relationship with the customer.

Confente (2015) asserts that given the rapidly changing competitive environment brought on by current consumer and technological trends, which make customer satisfaction more crucial than ever, tourism destinations and service providers must pay even closer attention to customer satisfaction in the modern environment (Confente, 2015). These trends include the shift in the tourism industry from a service economy to an experience economy, where visitors are more seasoned and have higher expectations, the explosive growth of sharing platforms like Airbnb and Uber that upend the traditional business model and increase competition for traditional service providers, and the increasing influence of electronic word-of-mouth (eWOM) on Web 2.0 platforms like social media on consumers' expectations and decisions. The internet has made a

powerful impact on the tourism industry by customers reviewing websites to look for pictures and read reviews from past visitors, utilization of online advertising, social media, blogs and online purchasing to help customers to choose their preferences.

Mohd-Any, Winklhofer and Ennew (2015) carried out a study seeking to measure multidimensional user's value experience when using travel firms. The study specifically sought after how users create value through using travel experience. The study which was based in the United Kingdom undertook an extensive literature review to identify the dimensions of customer value and randomly sampled 3000 customers who had used online travel providers. Partial least squares (PLS) analysis showed a positive influence of quality services on e-value satisfaction. The study concluded that high quality service by travel providers significantly influences customer experience which determines e-value generation.

Noting the importance of service quality, Stringham and Gerdes (2019) sought after customer satisfaction by examining the effect of service quality on hotel performance. The study focused on large international hotels with six or more affiliated hotels from six continents. A total of 259 travel providers online platforms were analyzed using the GTMetrix which is a useful tool that runs performance measurements of a firm's online presence. The study revealed there exists a significant gap between the expected service quality and the actual perceived service quality; and that lack of an e-tourism platform contributed to this gap in service delivery expectations. The researcher affirmed that lack of an e-tourism platform contributes to low hotel bookings especially among younger guests who use mobile devices rather than computers when making purchases eventually leading to brand devaluation.

2.3 Theoretical Framework

2.3.1 The Social Cognitive Theory

The Social Cognitive Theory (SCT) is a theoretical framework developed by Albert Bandura in the 1980s. The theory emphasizes the role of social interactions, personal experiences, and cognitive processes in shaping individual behavior (Hardin, J., 2010). SCT posits that people learn by observing others and then modeling their behaviors, and that this learning is influenced by cognitive factors such as motivation, self-efficacy, and outcome expectations.

At the core of SCT is the idea that behavior is a product of both personal and environmental factor. (Weaver, T. ,2013). In other words, people's behavior is shaped not only by their own characteristics and experiences, but also by the social and cultural context in which they live. SCT suggests that people are more likely to adopt new behaviors if they see others around them engaging in those behaviors and if they believe that they can successfully perform the behavior themselves.

SCT has been applied to a wide range of contexts, including health behavior, educational settings, and workplace behavior. In the context of the study on the decision-making assistant for travel agencies using sentiment analysis, SCT could be used to understand how social media interactions and experiences shape customer perceptions and preferences in the E-tourism industry. For example, the study could examine how customer reviews and feedback on social media influence the behavior and decision-making of other customers, and how this learning is influenced by cognitive factors such as motivation, self-efficacy, and outcome expectations

2.3.2 The Technology Acceptance Model (TAM)

The Technology Acceptance Model (TAM) is a theoretical framework developed by Fred Davis in the 1980s to explain the factors that influence the adoption and usage of new technology. The model posits that the adoption of new technology is influenced by two main factors: perceived usefulness and perceived ease of use (Davis, F. D. ,1989). Perceived usefulness refers to the extent to which an individual believes that using the technology will enhance their performance or productivity. Perceived ease of use refers to the extent to which an individual believes that using the technology will be effortless and easy to learn. TAM suggests that people are more likely to adopt and use new technology if they perceive it to be useful and easy to use. The model also suggests that these perceptions are influenced by external factors such as social influence, training, and technical support (Karahanna, E., & Straub, D. W., 1999). TAM has been widely applied to a range of technology adoption contexts, including e-commerce, mobile devices, and online learning.

In the context of the study on the decision-making assistant for travel agencies using sentiment analysis, TAM could be used to understand users' attitudes and behaviors towards the machine learning tool that the study aims to develop. For example, the study could examine how users

perceive the usefulness and ease of use of the machine learning tool and how these perceptions influence their adoption and usage of the tool. Additionally, the study could examine how external factors such as social influence and technical support influence users' perceptions and behaviors towards the tool.

2.4 Existing Recommender Systems

The concept of recommender systems has attracted much attention in recent years thus it has formed a base for a few systems that have been utilized to explain the relationship of recommender systems with different industries. They include TripAdvisor, Kayak, Amadeus, Booking.com, Airbnb among others (Ingaldi, 2021). This research will analyze three key systems: Trip Advisor, Kayak and Amadeus.

2.4.1 TripAdvisor

TripAdvisor is a popular travel website that allows users to read reviews, compare prices, and book travel arrangements (TripAdvisor, 2023). It also offers a travel recommender system that suggests destinations, activities, and accommodations based on user preferences. The recommender system employs collaborative filtering techniques to analyze user preferences and behaviors, as well as aggregated user reviews and ratings for activities, attractions, and tours. By considering the experiences of like-minded travelers, TripAdvisor suggests personalized recommendations to enhance the travel itinerary and ensure a memorable travel experience. TripAdvisor has a large database of user-submitted reviews, making it a great resource for finding out what other travelers think of a particular destination or accommodation. The website also offers a variety of tools to help users compare prices and book travel arrangements.

However, TripAdvisor has various limitations which highlight the complexity of developing and maintaining effective travel recommender systems. TripAdvisor has been criticized for its lack of transparency and for allowing businesses to pay to have their reviews displayed more prominently. Secondly, TripAdvisor's recommendations heavily rely on user-generated content such as reviews and ratings. However, ensuring the quality and reliability of these user-contributed reviews can be challenging and requires robust mechanisms to filter out biased or fake reviews and to maintain a high level of trustworthiness in the recommendations. Finally, TripAdvisor offers a vast amount of information and recommendations, which can sometimes be overwhelming for users. Sorting

through numerous options and reviews to find the most suitable activities or attractions may be time-consuming and may require additional filtering options or personalized recommendations.

2.4.2 Kayak

Kayak is a travel search engine that allows users to compare prices for flights, hotels, and rental cars. It also offers a travel recommender system named Kayak Explore that suggests destinations based on user preferences (Kayak, 2023). Kayak Explore is a feature of the popular travel search engine Kayak. It allows users to explore travel destinations based on their budget, travel dates, and preferences. The interactive map interface enables users to view flight and hotel prices for different locations, helping them discover new destinations and find inspiration for their trips. Users can adjust search filters and explore various regions to find travel options that meet their criteria. Kayak is a great option for travelers who are looking for the best possible price on their travel arrangements. The website allows users to compare prices from a variety of different travel providers, and it also offers a variety of tools to help users save money on their travel.

However, Kayak is limited in a number of ways. First, While Kayak Explore provides an interactive map and enables users to explore different travel destinations, its coverage might be limited compared to other travel platforms. It relies on partnerships and data availability from airlines, hotels, and other sources. This can result in certain regions or lesser-known destinations having limited or incomplete information. Secondly, The accuracy of pricing information can be a challenge in travel recommender systems. Fluctuating prices, hidden fees, and varying availability can make it difficult to provide real-time and accurate pricing data. Users may need to verify the pricing details directly with the travel providers. Finally, While Kayak Explore allows users to filter search results based on budget and travel dates, the level of personalization may be limited compared to other platforms. It primarily focuses on providing a visual representation of destinations and prices, rather than deeply analyzing individual preferences and travel behavior.

2.4.3 Amadeus

Amadeus is a leading technology provider for the travel industry (Amadeus, 2023). Their Personalized Travel offering utilizes advanced machine learning and user profiling techniques to deliver personalized recommendations to travelers. The system analyzes a variety of data sources, including user preferences, historical booking data, and contextual information, to create tailored

recommendations for flights, hotels, and activities. By understanding individual traveler preferences, Amadeus aims to enhance the travel experience by providing relevant and customized options.

One of the key challenges in personalized travel recommendations is ensuring the privacy and security of user data. Amadeus Personalized Travel relies on collecting and analyzing user preferences and historical data, which raises concerns about data protection and the ethical use of personal information. Achieving accurate and highly personalized recommendations can also be challenging. The system needs to accurately interpret user preferences and match them with relevant travel options. It requires robust machine learning algorithms and continuous fine-tuning to improve recommendation accuracy and adapt to evolving user preferences. Lastly, access to comprehensive and up-to-date travel data can be a challenge. Amadeus Personalized Travel relies on integrating and processing data from various sources, including airlines, hotels, and other travel providers. Ensuring the availability, quality, and real-time updates of this data can be a complex task.

2.4.4 Other Systems

iTravel is a personalized travel recommender system that utilizes a collaborative filtering algorithm to provide recommendations based on user preferences and past travel experiences (Yang and Hwang, 2013). The system collects user feedback and ratings on various travel destinations, accommodations, and activities then identifies similar users based on their travel preferences and generates recommendations based on the preferences of those similar users. iTravel incorporates a variety of factors such as destination popularity, travel time, and user preferences (e.g., budget, preferred activities) to generate personalized recommendations and aims to enhance user satisfaction and help travelers discover new destinations and experiences based on the experiences and feedback of others (Yang and Hwang, 2013).

Hsu, Tsai, and Wu (2012) proposed a 4-level Analytic Hierarchy Process (AHP) model for travel recommender systems in Taiwan. The model involves four levels: goal, criteria, sub-criteria, and travel options. Users are asked to provide their preferences on various criteria and sub-criteria (e.g., destination attractiveness, cost, safety, convenience) based on their importance. The AHP model assigns weights to the criteria and sub-criteria based on the user preferences, and these

weights are used to prioritize and rank the travel options. The system aims to assist travel agents in recommending suitable travel options to their clients by considering their preferences and priorities (Hsu, Tsai, and Wu, 2012).

e-Tourism is a travel recommender system proposed by Sebastia, García, Onaindia, and Alvarez, (2009) that employs collaborative filtering techniques to recommend travel destinations, accommodations, and activities. The system utilizes a novel similarity computation method that considers both user ratings and the similarity of item ratings to generate recommendations. e-Tourism incorporates user preferences and historical data, such as user ratings and reviews, to identify similar users and recommend travel options based on the preferences of those similar users. The system aims to provide personalized recommendations that match the individual preferences of users, helping them discover new and relevant travel options (Sebastia, et. al., 2009).

ATRS, a system proposed by Etaati and Sundaram, (2015), is an adaptive tourist recommendation system that was built on top of the existing systems to improve their adaptiveness and growth. The new system was based on an existing location-based travel recommender system proposed by Ravi and Vairavasundaram (2013) which utilizes the Social Pertinent Trust Walker (SPTW) algorithm. The system leverages social network data and user trust relationships to provide recommendations based on user preferences, location, and trust relationships within their social network. SPTW algorithm combines user's preferences, check-in history, and trust relationships to calculate trust scores for various travel options. The system recommends travel destinations and activities based on the calculated trust scores and considers the proximity of the recommended options to the user's current location.

2.5 Machine Learning Concept

Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and models that enable computers to learn and improve their performance based on data (Kamel et al., 2018). Machine learning algorithms are designed to learn from data by identifying patterns and relationships and using these to make predictions or decisions. These algorithms use statistical methods to find patterns and relationships in the data and create models that can be used to make predictions on new data. The key concept in machine learning is the idea of "training" a model on a set of labeled data, where the desired output is known for each input. The model uses this data to identify patterns and relationships, and to develop a set of rules or parameters that can be used to predict or classify new inputs (Kamel et al., 2018).

There are three main types of machine learning: supervised learning, unsupervised learning, and reinforcement learning (Nilashi *et al.*, 2017). In supervised learning, the algorithm is trained on a labeled dataset, meaning that the data has already been classified using a classifier. A classifier is a function *f* that maps input feature vectors $x \in X$ to output class labels $y \in \{1, \ldots, C\}$, where X is the feature space. We will typically assume $X = IRD$ or $X = \{0, 1\}^D$, i.e., that the feature vector is a vector of D real numbers or D binary bits, but in general, we may mix discrete and continuous features. We assume the class labels are unordered (categorical) and mutually exclusive. Our goal is to learn f from a labeled training set of N input-output pairs, (xn, yn), $n = 1 : N$; (Murphy, K. P., 2006) The algorithm then uses this labeled data to make predictions or classifications on new, unlabeled data, this is an example of supervised learning.

In unsupervised learning, the algorithm is trained on an unlabeled dataset, meaning that there are no predefined classifications. The algorithm must identify patterns and relationships in the data on its own. In reinforcement learning, the algorithm learns through a process of trial and error, receiving feedback from its environment and adjusting its actions accordingly.

Machine learning has numerous applications in various fields, including computer vision, natural language processing, recommendation systems, fraud detection, and predictive maintenance, among others. The goal of machine learning is to develop models and algorithms that can learn and adapt to new data, and that can make accurate predictions or decisions based on that data. (Nilashi *et al.*, 2017).

2.5.1 Machine learning Natural Language Processing (NLP)

Machine learning Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) that focuses on the intersection of machine learning and language understanding. It involves the development and application of algorithms and models to enable computers to understand, analyze, and generate human language (Bishop, 2009). NLP techniques leverage machine learning algorithms to process and extract meaning from textual data. These algorithms learn patterns and relationships from large amounts of labeled or unlabeled text data, allowing computers to perform various language-related tasks.

Machine learning NLP techniques typically involve preprocessing steps like tokenization (breaking text into individual words or tokens), stemming (reducing words to their base or root form), and removing stop words (commonly used words with little semantic value). These techniques often employ algorithms or probabilistic models to learn patterns and relationships in the text data. The objective of machine learning NLP is to enable computers to understand and process human language in a way that is similar to how humans do. It has numerous practical applications, including information retrieval, sentiment analysis, chatbots, voice assistants, and more. Sentimental analysis will be the focus in the current study.

2.5.1.1 Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a machine learning concept that involves using natural language processing (NLP) and machine learning techniques to analyze and classify the sentiment of text data (Hutto & Gilbert, 2014). The goal of sentiment analysis is to automatically determine whether a piece of text expresses a positive, negative, or neutral sentiment (Hutto & Gilbert, 2014). The process of sentiment analysis involves several steps:

1. Text pre-processing: The text data is cleaned and pre-processed to remove any irrelevant information, such as stop words, punctuations, and URLs.
2. Feature extraction: The text data is converted into a set of features that can be used to train a machine learning model. Common features include word frequencies, n-grams, and part of-speech tags.

3. Training a classifier: A machine learning algorithm, such as a support vector machine (SVM) or a neural network, is trained on a labeled dataset of text samples, where each sample is labeled as positive, negative, or neutral.
4. Evaluating the classifier: The performance of the classifier is evaluated on a separate test dataset, to measure its accuracy, precision, recall, and F1-score.
5. Predicting sentiment: Once the classifier is trained and evaluated, it can be used to predict the sentiment of new text data.

Sentiment analysis has numerous applications in various industries, such as social media monitoring, customer feedback analysis, and brand reputation management. It can help businesses and organizations to understand the opinions and attitudes of their customers, and to make informed decisions based on that information (Hutto & Gilbert, 2014).

Various studies have employed sentiment analysis in the service industry. Hu, Tsai, and Chou (2017) applied sentiment analysis on online hotel reviews to understand customer sentiments and preferences. They used machine learning algorithms to classify review sentiments and identified key factors influencing customer satisfaction. A study by Kim, Jang, and Park (2019) analyzed sentiment in online hotel reviews to identify factors affecting customer satisfaction and dissatisfaction. They used sentiment analysis and topic modeling techniques to extract sentiments and topics from reviews and the findings were useful to hotel managers in understanding customer preferences and improving service quality.

Further, Wang, Liang, and Huang (2017) utilized sentiment analysis on online travel reviews to evaluate service quality. The study employed a machine learning approach to classify review sentiments and assessed the impact of different aspects of service quality on customer satisfaction. The findings helped hotels to identify areas for improvement. A study by Xiang, Du, Ma, and Fan (2017) examined sentiment in online hotel reviews and its impact on hotel reputation and booking intentions and used sentiment analysis to classify review sentiments and found that positive sentiments significantly influenced hotel reputation and booking intentions. The findings helped hotels in managing their online reputation. These studies highlight the application of sentiment analysis in the service industry, specifically in the context of reviews. By analyzing sentiments expressed by customers, these studies provide insights into customer preferences, service quality evaluation, reputation management, and customer satisfaction.

2.5.2 Machine Learning Algorithms

A machine learning algorithm refers to a specific mathematical or computational method used to train a machine learning model on a dataset and make predictions or decisions based on that model (Hastie, Tibshirani & Friedman, 2009). There are various machine learning algorithms used in sentiment analysis including: Naive Bayes, Support Vector Machines (SVM), Random Forest, K-Nearest Neighbors, Gradient Boosting Methods, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) and Transformer-based Models. The study will adopt Naïve Bayes, Support Vector Machines (SVM) and K-Nearest Neighbors algorithms.

2.5.2.1 Naïve Bayesian

The Naïve Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification (Rennie, J. D., 2001). To achieve recommendations that target the personalized recommendations, usually one must consider groups with two factors. In our case, we could proceed with the user-object pair, where we depicted user classification and objects in the K user and K object classes, respectively. K user and K object values are parameters in the algorithm like K which was a Singular Value Decomposition (SVD) parameter (Rrmoku, K., Selimi, B., & Ahmedi, L. ,2022). For a simple Bayesian network, we would create an assumption that *ck* depends only on the user class *ca* and the object class *ce*. Thus, the probability *ck*, could be written as:

$$P(ck)=\sum K \text{ user } ce =1 \sum K \text{ object} ce= P_{(}ca, ce)P(ck)P(ce)$$

The results of the Bayesian network correspond to the simplest dependency structure linking ratings to user classes and objects. In this study, we opted for a prediction model, that can give results based on the data that is observed. In this context, Naive Bayes (NB) fits our approach, mainly due to the following arguments:

It is independent—that means that we can consider all properties as independent given the target Y. It is equal—an event where all attributes are considered as being with the same importance.

Conditional probability is then calculated, which is known as the Naive Bayes algorithm since it uses the Bayesian theorem, and thus calculates the probability of an event, based on the incidence of values in historical data. The following formula, shows the definition of the Bayesian Theorem:

P(A|B) = (P(B|A) P(A))/(P(B))

A—represents the dependent event,

B—represents the preceding event, thus predictive attribute.

P(A)—represents the probability of the event before it is observed.

P(A|B)—represents the probability of the event after the evidence is observed.

While, "Naive" Assumption is defined as the evidence that is divided into pieces, that are meant and defined to be independent.

P(A) = (P(A1|B) P(A2|B) …P(An|B))/(P(B))(4) where A1 and A2 are independent occurrences.

2.5.2.2 Support Vector Machine

The support vector machine (SVM) is an extension of the soft margin classifier that allows for the creation of more general decision surfaces than just a separating hyperplane. To achieve this, a special kind of function $\boldsymbol{\phi}(\mathbf{x})$ that maps the input data $\mathbf{x}1, \ldots, \mathbf{x}n \in \mathcal{X}$ into a high-dimensional feature space $\mathscr{H}$ is used, and the linear separation is done there (B. Schölkopf, A. J. Smola, and F. Bach, 2002). This function is called kernel and the procedure of mapping **x** into a high-dimensional feature space is called the kernel trick. It takes low dimensional input space and transforms it to a higher dimensional space, i.e., it converts a not separable problem to a separable problem.

It is mostly used in non-linear data separation problems that do complex data transformations and finds out the process to separate the data based on the labels or outputs that have been defined. The hyper-plane in the original input space displays the data points in a circle.

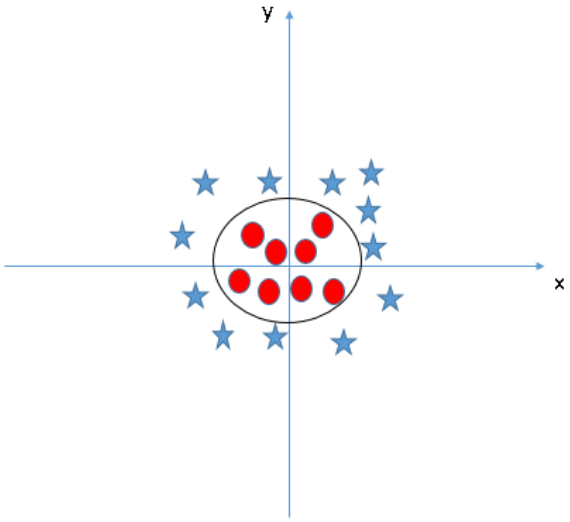


Figure 2.1: Hyper-plane with original input space

This study implements support vector machine as one of the machine learning algorithms as it is effective in handling nonlinear data patterns where user preferences and item features are complex. It uses a kernel function to map the input data into a higher-dimensional space, where it finds the optimal hyperplane that separates different classes or predicts the rating for a specific item. During preprocessing of the data there are outliers that occur due to varying user preferences and noisy data, this is handled by focusing on support vectors, which are the data points closest to the decision boundary.

2.5.2.3 K-Nearest Neighbor Algorithm

The K-nearest neighbors (k-NN) algorithm is a non-parametric supervised learning method used for both classification and regression tasks. It was first developed by Evelyn Fix and Joseph Hodges in 1951 and later expanded by Thomas Cover. The K-Nearest Neighbor (KNN) method has several advantages: ease, effectiveness, intuitiveness, and competitive classification performance in many domains (Kbaier, Masri, & Krichen, 2017). The algorithm relies on the idea that similar data points tend to be close to each other in the feature space.

In K-NN classification, the algorithm assigns a class membership to an object based on a plurality vote of its k nearest neighbors and it is the most like it in terms of common features. This is a process known as Similarity search.

For classification problems, the object is assigned to the class that is most common and has a majority vote among its k nearest neighbors. If k is equal to 1, then the object is simply assigned to the class of its single nearest neighbor. The algorithm predicts the property value for an object by taking the average of the values of its k nearest neighbors. Again, if k is equal to 1, then the output is simply assigned to the value of its single nearest neighbor. After completion, the model provides results in form of discrete values. The accuracy of the results is determined by how close the model's predictions and estimates match the known classes of the testing test.

To determine the nearest neighbors, the algorithm calculates the distance between the query point (the object to be classified or predicted) and the other data points. Various distance metrics can be used, including Euclidean distance, Manhattan distance, Minkowski distance, and Hamming distance. These distance metrics help define decision boundaries and determine which data points are closest to the query point.

One important consideration when using K-NN is that it is a local approximation algorithm. All computation is deferred until the function evaluation, which means that the algorithm approximates the function only locally. Normalizing the training data can improve the accuracy of K-NN, especially when the features represent different physical units or come in vastly different scales. The optimal k value will help you to achieve the maximum accuracy of the model.

This study leverages on a hybrid approach using collaborative and content-based filtering, KNN collaborative filtering nature is applied as it commonly used to leverage the concept of similarity between feedback and reviews to make personalized recommendations. It adapts well to changing user patters this is an added advantage as it provides recommendations based on the historical data and current data availability.

2.6 Summary of the Literature Review and Research gaps

The literature provides adequate evidence that service quality greatly impacts customer – travel provider relationships within the e-tourism industry. More importantly, the literature emphasizes the need for e-tourism businesses to direct efforts to the development and usage of tools that analyze consumer behavior and preferences as a means of learning their consumers (Ku and Chen, 2015). This knowledge allows businesses to create experience driven services and hence achieve greater deals of delivery quality services to their customers in the long run.

Based on the research done and recommender systems that have been developed over the years, service quality has not been optimized as the systems focus on the customer expectations and fail to analyze the outcomes and consumer feedback. Most of the models reviewed have not integrated their processes using machine learning and there is a need of using machine learning algorithms to analyze the consumers reviews and feedback. This will be important since it can help businesses make data-driven decisions. By analyzing large amounts of data, machine learning algorithms can provide insights that businesses can use to make informed decisions about their products, services, and customer experience. This can lead to better customer satisfaction, increased sales, and improved business performance overall.

The research further revealed that most travel recommender systems are not personalized enough. They typically recommend popular destinations, regardless of the user's individual preferences. The system developed from this research aims to fill this gap by using sentiment analysis to recommend destinations that are likely to appeal to the user's interests and preferences. Additionally, most travel recommender systems use data that is weeks or even months old which can lead to inaccurate recommendations, as travel destinations can change quickly. The system developed sought to fill this gap by using real-time data from social media to recommend destinations that are currently popular and trending.

Comparison of various recommender systems has been tabulated on table 2.2 and the gap in each model has been identified.

System	**Gaps**
TripAdvisor	TripAdvisor's recommendations can be biased towards businesses that pay for advertising. TripAdvisor offers a vast amount of information and recommendations, which can sometimes be overwhelming for users
Kayak	Its coverage is limited compared

	It focuses on providing a visual representation of destinations and prices, rather than deeply analyzing individual preferences and travel behavior.
Amadeus	It relies on collecting and analyzing user preferences and historical data, which raises concerns about data protection and the ethical use of personal information.

Table 2.1: Research Gaps

2.7 Conceptual Framework

Kothari (2004), highlights that a conceptual framework is a way to organize and present the relationships among variables, concepts, and theories that are relevant to the study. Saunders et al., (2012) further posits that it serves as a guide for the researcher, helping to define the problem, identify the research questions, and outline the methods that were used to collect and analyze data. The conceptual model in the study provides an overview of a travel agency recommender system that is based on customer feedback and reviews from social media platforms.

The tool will address the gaps in the available recommender system by analyzing the dependence of the travel providers on the customer feedback and outcomes of their travel instead of the desires, tastes, and preferences. This ensures that the travel providers remain relevant in the market by consistently offering good service which ensures that the rising margins of the travel consumers are well catered for.

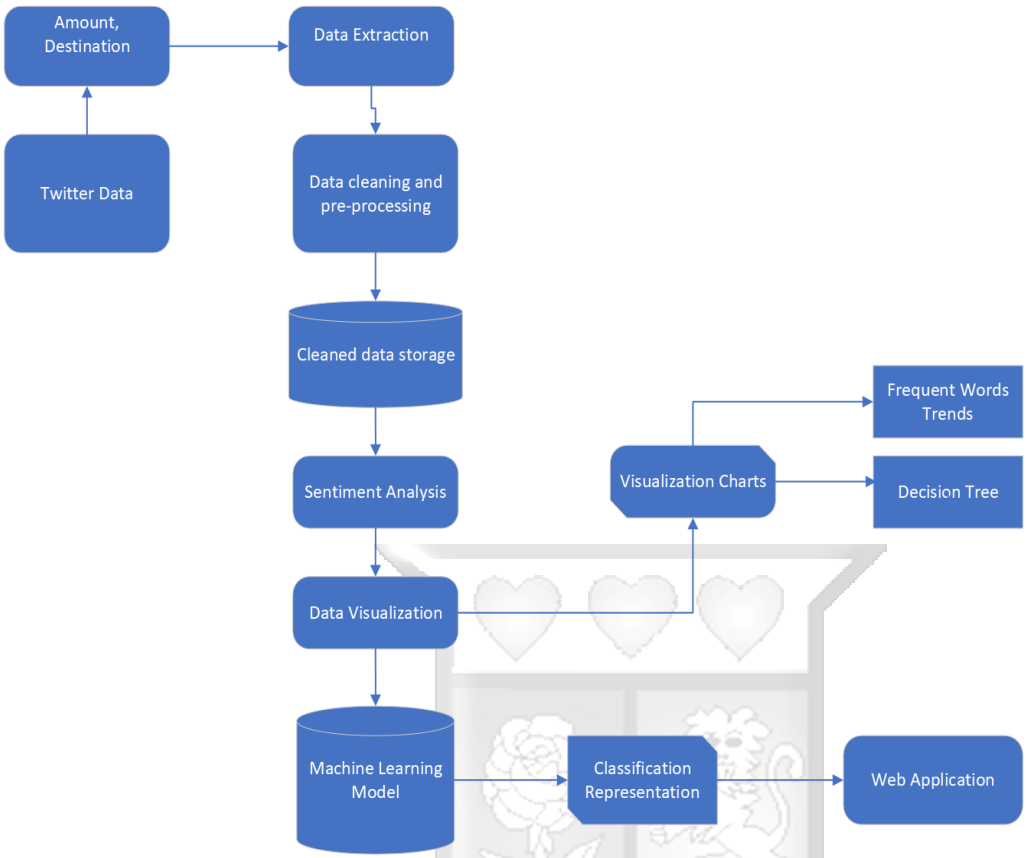


Figure 2.2: Conceptual Framework

The variables were operationalized as shown in Table 2.2 below.

Variable	Indicators	Data analysis	Supporting Literature
Service Quality	Customer Satisfaction Customer loyalty Competitive advantage	Sentiment Analysis	O'Connor, Trinh & Shewchuk, (2010) Mohd-Any, Winklhofer & Ennew (2015)
Tool Development	Customer Feedback Customer Review Customer Experience	Sentiment Analysis	(Kamel et al., 2018). (Nilashi *et al.*, 2017).
Tool Reliability	Accuracy F1 Score Confusion Matrix Precision	Sentiment Analysis	Xie et al. (2021) Kim et al. (2018) Hsu et al. (2019)
Tool Classification	Destination Travel Patterns User Similarity	Naïve Bayes K-Nearest Neighbor Support Vector Machine	(Kbaier, Masri, & Krichen, 2017). (B. Schölkopf, A. J. Smola, and F. Bach, 2002). (Rennie, J. D., 2001).

Table 2.2 Operationalization of Study Variables

Chapter 3: Research Methodology

3.1 Introduction

As indicated in the Introduction, the main aim of this research was to use machine learning to develop a travel agency recommender system. The methodology adopted was characterized by collection of data, exploring, and preparing the data, training the tool on the data and methods to be implemented to determine reliability of the tool. The machine learning algorithms that were investigated include K Nearest Neighbors, Naive Bayes and Support Vector Machine.

3.2 Research Design

The plan or strategy for doing research, including the techniques and steps that will be taken to gather and evaluate data, is known as the research design (Kombo & Tromp, 2006). The research design provides a structure for the study and guides all aspects of the research process. Tobi and Kampen (2018) posit that a study design entails the procedures taken to address the research problem including aspects related to the collection, measurement, and analysis of data. The sort of research design to be employed is determined by the study problem. The research design aids the researcher in gathering data pertinent to the study's topic. This can be done to test a theory, evaluate a program, or to describe the process involved in a phenomenon (Van Wyk, 2012). Exploratory research design was adopted for this study. The research used the exploratory research design to gain a better understanding from both travel providers and consumers, resulting in better insights and understanding of how to facilitate service quality in the E-tourism industry.

3.3 Target Population

According to Creswell & Creswell (2017), a population is the entire collection of entities from whom one seeks to learn more about them or, more precisely, from which one wishes to draw inferences. The study was focused on collecting data from Twitter. Currently Twitter has 450 million monthly active users (Twitter, 2022). The study population was focused on people who post on social media in the e-tourism industry. These include travelers, tour guides, tour operators, travel companies and other tourists. This study population was considered because social media platforms have become an integral part of the travel industry, with millions of users sharing their travel experiences, feedback, and reviews online and thus provides a rich source of data that can be used to gain valuable insights into customer behavior and preferences. The study analyzed their

social media presence and how they manage to capture feedback and reviews from their consumers. The range of their processes and procedures were analyzed to determine how to gain knowledge and insights that focuses on improving their customer experience and creating a brand image.

3.4 Sampling Technique

A sample technique is the framework, or road map, that guides the selection of a survey sample (Van Wyk, 2012). In this study purposive sampling was used. Purposive sampling involves selecting participants based on specific criteria relevant to the research question. In this case, the criteria would be people who post on social media in the E-tourism industry. This sampling technique is useful in situations where the population of interest is not easily accessible, and it is important to select participants who can provide the most relevant and useful information. Sampling is a methodical approach for selecting a subset from a sampling frame or the complete population (Tobi & Kampen, 2018). For this study the sample frame was selected from the 450 million active twitter users with focus on travel agencies related tweets.

3.4 Data Collection

According to Burns & Groove (2014), data collection is the methodical process of gathering and measuring information on variables of interest to analyze results, test hypotheses, and respond to research questions. The research focuses on twitter for the openness and platform availability to developers. To build a corpus, travel related tweets were collected from Twitter. To build robust comprehensive corpora, Snscrape will be utilized to scrap the required data from twitter. Snscrape is a Python-based scraping tool used for extracting data from social media platforms such as Twitter, Instagram, and Reddit. It can be used to gather data such as tweets, comments, user profiles, and hashtags. Snscrape uses web scraping techniques to extract data from social media sites and returns the data in a JSON or CSV format. It can be used to gather many tweets, follows, followers, likes, and other information from Twitter profiles, hashtags, and keywords, and it also has built-in support for geolocation, searching for specific tweet types, and more.

3.5 System Development Methodology

The tool will be developed by adopting Agile Development Methodology which helps in delivering solutions on time and provides improved quality of travel providers and customer

satisfaction. Agile development is preferably used as it focuses on managing time to market constraints and accommodates changes during the software development life cycle (Cao *et al.*, 2009). Agile Planning and Delivery will be dependent on the data collected and data manipulation which will provide a well-structured plan that enables better business insights. As customers are demanding better services to meet their tastes and preferences, the machine-learning based tool must have advanced capabilities that provide clear visibility of the anticipated results, create a strategy that guarantees value proposition and deliver services that meet the consumers precision from accessing the right data.

The machine learning tool will identify patterns, consumer behavior insights, analyze delivery plans, and embrace algorithms such as K-Nearest Neighbor and Support Vector Machines to detect issues that predict the frequency and impact of the reviews and feedback from the consumers. This sets the stage for the travelers to gain information based on positive feedback from the recommender system.

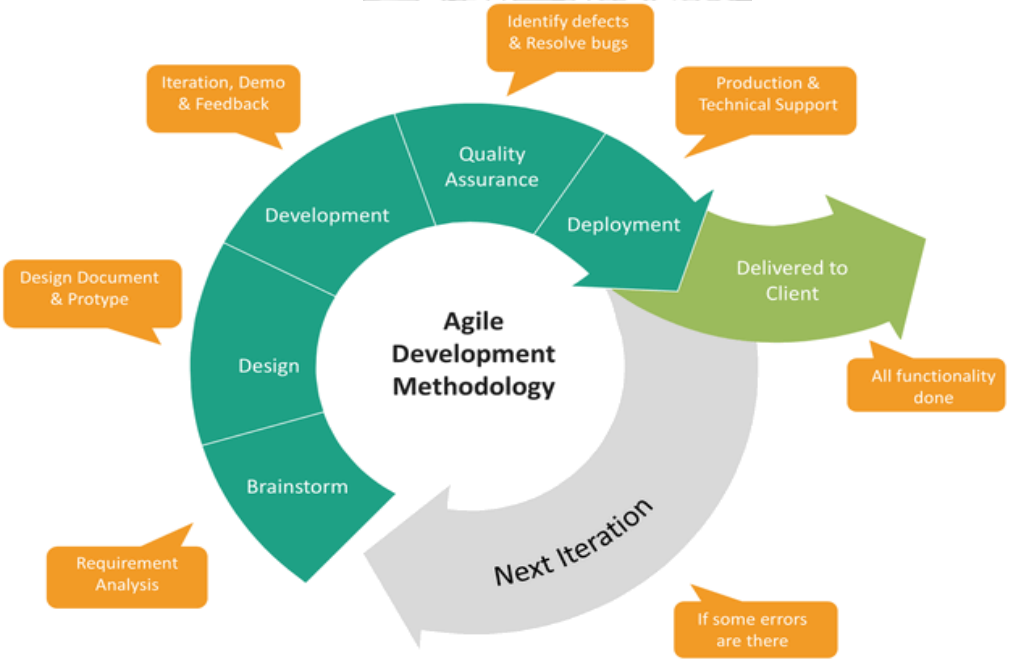


Figure 3.1: Agile Development Methodology Diagram

3.6 Data Analysis and Presentation

During the data gathering process, some irrelevant and noisy values were available on the datasets, thus it was essential to pre-process the data to remove the noisy and inconsistent data. Getting rid of inconsistent data in the systems is key to ensure a smooth process. After preprocessing the data was transformed using attribute selection as the research is focused on the values that influence service quality such as customer satisfaction, reliability, availability, and other attributes. Sentiment Analysis will then be utilized in the study. Sentiment analysis is the process of determining the emotional tone behind a piece of text, whether it be a tweet, a Facebook post, a review, or any other written document (Van Wyk, 2012). It can be used to identify the overall sentiment of a document, as well as to highlight specific examples of positive, negative, or neutral sentiment within the text (Van Wyk, 2012). Sentiment analysis was used to analyze customer reviews and feedback to gain insights on customer satisfaction and preferences, which can be useful for decision-making in travel agencies. After the data was collected and analyzed, the information was presented using tables, figures, and graphs with the aim of quantifying the elements that affect the quality of services that are offered by travel providers. The study results will be utilized for the purposes of this study only.

3.7 Research Quality

The research design standards will ensure that the research is of high quality. Four key criteria may be used to evaluate the quality of research: relevance, credibility, legitimacy, and efficacy. (Belcher, Rasmussen, Kemshaw and Zornes, 2016). These factors seek to determine whether the research is pertinent to the field on which it is based, credible, defensible in its assertions—that is, there are no exaggerations—true in the data collection, and, finally, effective—that is, that the suggested solutions outperform earlier works. By citing the research's reliability and validity, which were both used in the study, these elements may be expressed.

3.7.1 Reliability of the Research Instruments

Saunders *et al,* (2012) views reliability of data collection instruments as referring to the consistency of the results obtained using a particular instrument over time, or when used by different researchers or in different settings. Reliability is an important characteristic of any data collection instrument, as it helps to ensure that the results obtained from the instrument are accurate and

trustworthy. The study will adopt the test-retest reliability method. The Test-retest reliability according to Kombo & Tromp (2006), is a measure of the consistency of an instrument over time. This involves evaluating the model's performance on the same data set at two different times to see if the results are consistent.

3.7.2 Validity of Research Instruments

According to Borg and Gall (2013), Validity of data collection instruments refers to the extent to which the instrument measures what it is intended to measure. Validity is an important characteristic of any data collection instrument, as it helps to ensure that the results obtained from the instrument are meaningful and relevant. The study will adopt the Cross-validation and Holdout validation methods. Holdout validation involves splitting the data into training and testing sets and evaluating the model's performance on the test set, whereas cross validation involves folding the data into multiple folds and training the model on various subsets of the data to see how well it performs on unseen data. (Sekaran, 2010).

3.8 Ethical Considerations

The research will also ensure that ethical conduct is reflected in the behavior of the entire research team and process. The purpose of ethics in this study will be to ensure that no one is harmed or has negative consequences because of the research (Burns & Groove, 2014). The study will ensure that all participants are made aware of their rights to participate in the research work. Secondly, the respondents will not be required to provide any personal information which ensures their anonymity is guaranteed in the course of the survey. Thirdly, the study will ascertain the collected study data is only to be used for academic purposes and is not shared with any unauthorized people. Fourthly, the research will collect relevant approvals from the Ethics Review Committee of Strathmore University and the National Commission for Science Technology and Innovation.

Chapter 4: System Analysis and Design

4.1 Introduction

This section gives a thorough description of the created tool's logical and process architecture as well as its individual components and their respective fundamental functionalities.

4.2 System Design

This section covers the many criteria required by the suggested solution since the primary goal of this research is to create a tool that evaluates the level of service provided by travel suppliers in the e-tourism sector.

4.2.1 Functional Requirements

Functional requirements are a type of requirement that specify what a system or software application must do, in terms of its features, capabilities, and functions (Creswell & Creswell, 2017). These requirements include:

i. This program should have the ability to collect and analyze data from our main social media platform, Twitter.
ii. The system should be able to use natural language processing (NLP) techniques to analyze the sentiment of the collected data, such as positive, negative, or neutral.
iii. The system should also possess the ability to provide decision-making support to travel agents by suggesting recommendations for travel packages based on selected destinations retrieved from the algorithm.
iv. The system should have the ability to personalize its recommendations based on the preferences and needs of individual customers.

4.2.2 Non-Functional Requirements

Non-functional requirements are a type of requirement that specify characteristics of a system or software application that are not directly related to its features or capabilities (Creswell & Creswell, 2017). Non-functional requirements involved may include:

i. Performance - The system should be able to process and analyze large volumes of data quickly and efficiently, without experiencing significant delays or crashes.
ii. Reliability - The system should be reliable and consistently provide accurate and relevant recommendations to travel agents.
iii. Scalability - The system should be able to scale up or down as needed to accommodate changes in the volume of data or number of users.
iv. Maintainability - The system should be easy to maintain and update, with minimal disruption to its operations.
v. Usability - The system should be easy to use and intuitive for travel agents with varying levels of technical expertise.

4.3 System Architecture

System architecture defines the structure of modules, components, and their interactions to satisfy functional and non-functional requirements (Pang & Lee, 2008). It typically includes the identification of software components, their relationships, interfaces, and data flows. The architecture also describes how the software components are deployed and executed in the hardware infrastructure (Pang & Lee, 2008). The system architecture as illustrated in Figure 4.1 below shows the general design of the travel providers recommender system as per the objectives of the study. When a user enters the username of the travel agency as the keyword to direct the retrieval of pertinent tweets from an interested user, the categorization procedure is started. The appropriate tweets were then extracted using the given parameters using a twitter collecting tool called SnScrape. The unprocessed tweets then go through preprocessing, where the different data is categorized and saved in a csv file. The cleaned Tweets were analyzed and classified using a Vader lexicon model built using the NLTK library. Here, the classifier was trained to separate the tweets into three categories: positive, negative, and neutral. Each evaluated Tweet was then given a polarity tag. A Tweet was labeled a negative Tweet (neg) if it had more negative terms than positive words (pos), and a positive Tweet (pos) if it contained more positive words. However, the Tweet was regarded as neutral if the ratio of negative to positive terms was equal.

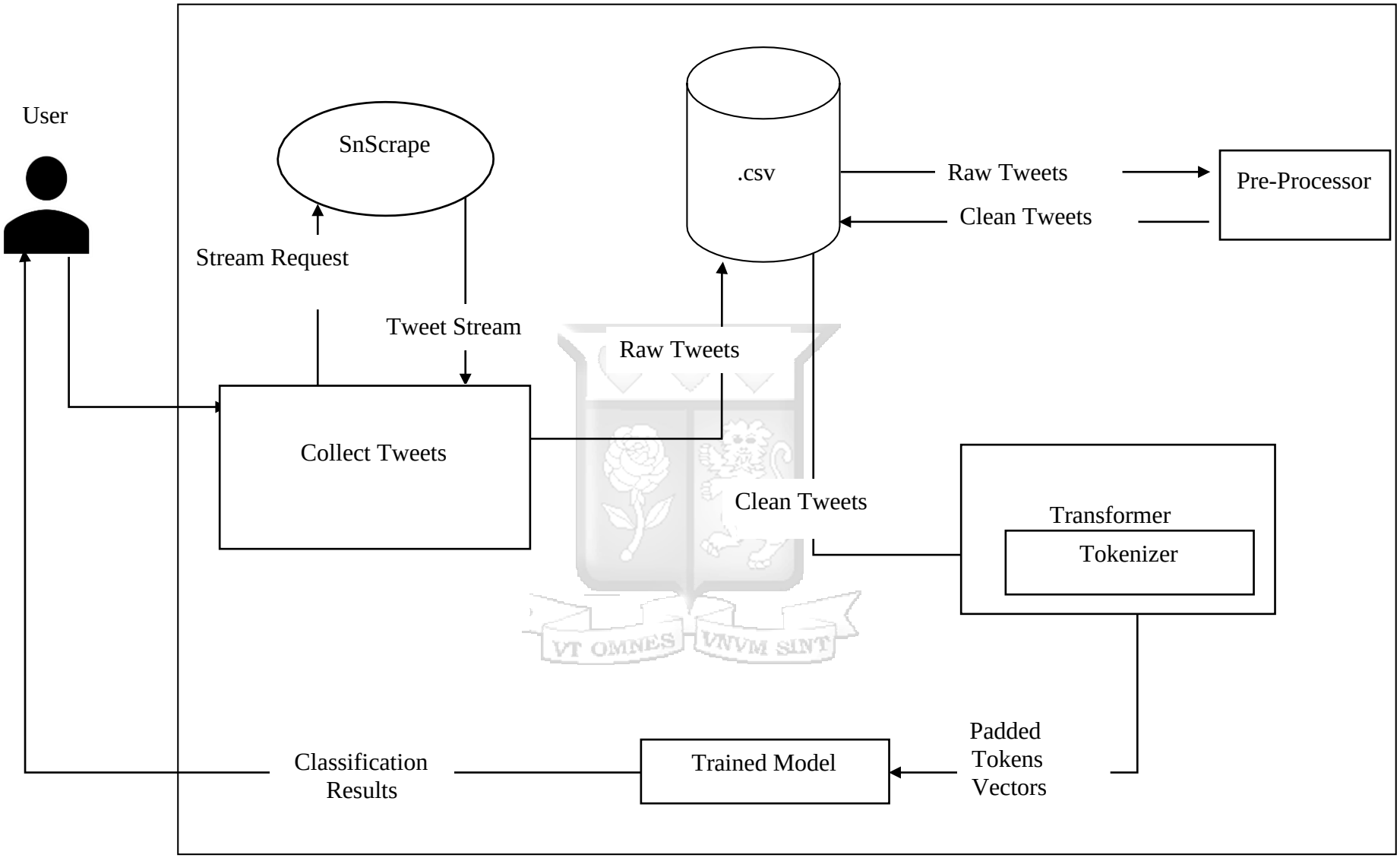


Figure 4.1 : System Architecture

4.4 Use Case Diagram

A use case diagram is a type of visual representation that describes the interactions between actors (users or systems) and a system to achieve specific goals or tasks and is used to help capture and communicate the functional requirements of a system.

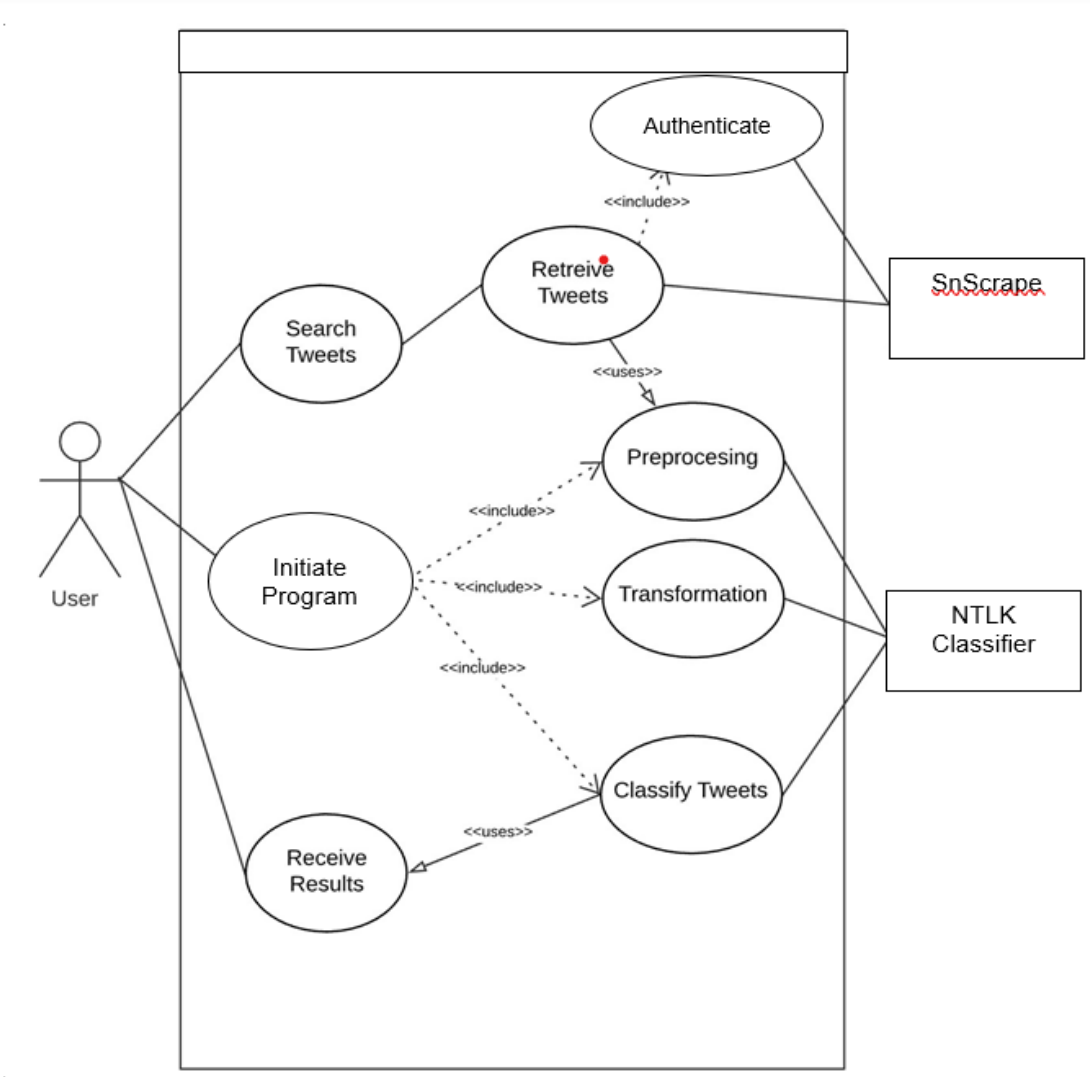


Figure 4.2: Use Case Diagram

The primary actors in the use case diagram are the user i.e. consumers and the system itself. The use cases for both the primary actors are highlighted below:

Use case: Search Tweets.

Primary Actors: User

Pre-conditions: The user has access to the internet on the platform/device being used.

Use case: Retrieve Tweets

Primary Actors: System.

Pre-conditions: Search use case completed successfully.

Use case: Initiate Program

Primary Actors: User, System

Pre-conditions: Retrieve use case completed successfully.

Use case: Pre-Process

Primary Actors: User, System.

Pre-conditions: Retrieve use case completed successfully.

Use case: Classify Tweets

Primary Actors: System.

Pre-conditions: Transform use case completed successfully.

Use case: Receive Results

Primary Actors: User.

Pre-conditions: Classify Tweets use case completed successfully.

4.5 Sequence Diagram

A sequence diagram is a graphical representation of the interactions among objects or components in a system or software application. It shows the sequence of messages exchanged between objects or components over time to accomplish a specific task or scenario (Pang & Lee, 2008). The user generates a search parameter, which is then utilized on Twitter to find related tweets containing

the specified keyword(s). Once the tweets are retrieved, they are transferred from the Twitter API to the processor for preprocessing. The processor, also known as the Analysis platform, then passes on the preprocessed tweets to the classifier for analysis and categorization as either positive, negative or neutral. The classification results are stored in a database. Eventually, the generated outcomes from the database are employed to create a report that pertains to a specific person or topic of interest.

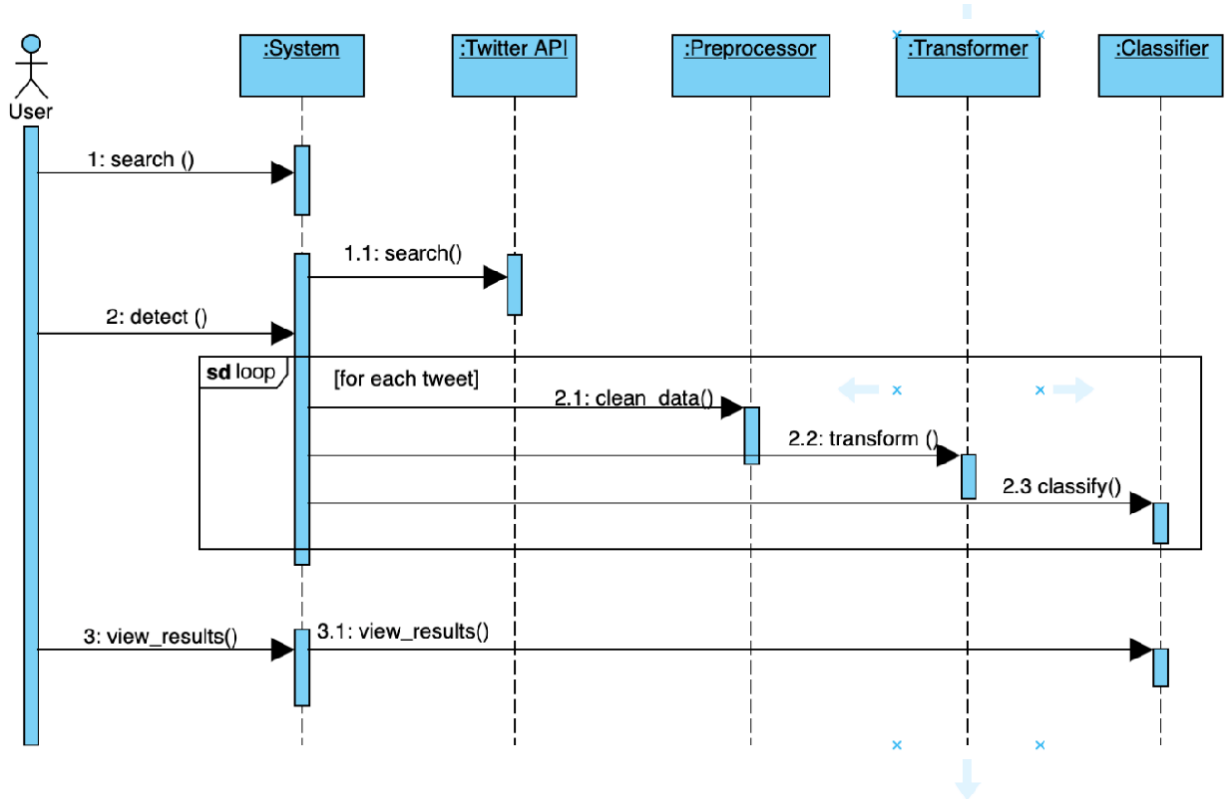


Figure 4.3: Sequence Diagram

Chapter 5 System Implementation and Testing

5.1 Introduction

In this chapter, we present the detailed implementation, testing methodologies, and the results of our recommender system for travel agencies. The system is built using tweets collected from 9 different travel agencies namely Uncharted Kenya, Bountiful Safaris, Let's Discover Travel, TrippyGo Tours, Lai Group Ke, Bonfire Safaris, Turnup.Travel, Oribi Expedition and Kilian Tours using the snscrape Python library. The collected data is preprocessed to prepare it for analysis, followed by sentiment analysis. Three machine learning models, namely KNN, Naive Bayes, and SVM, are utilized for sentiment classification. Finally, a hybrid filtering approach is implemented in the recommender system, combining content-based filtering using cosine similarity and collaborative filtering using Naive Bayes. This chapter provides a comprehensive overview of the system implementation, testing procedures, and the obtained results.

5.2 System Implementation

The main goal of this section is to classify Travel Agencies' sentiments based on the tweets; whether the customers have filed complaints, appreciations, or neutral comments. Tweets collected and analyzed were of every Kenyan who has mentioned the Travel Agencies in their timeline. The Travel Agencies used for the analysis were, Uncharted Kenya, Bountiful Safaris, Let's Discover Travel, TrippyGo Tours, Lai Group Ke, Bonfire Safaris, Turnup.Travel, Oribi Expedition and Kilian Tours among the top 9 travel agencies which was based on historical data collected for the past 5 years. The programming tools used to achieve our goal are Jupyter Notebook, python language, and several other libraries such as Matplotlib, Seaborn, Pandas, and NumPy. Nltk and Scikit-learn.

5.2.1 Collection of Tweets

In the system implementation phase, the first step involved collecting tweet data from the above named 9 travel agencies on Twitter using the snscrape library. Snscrape offers comprehensive data extraction capabilities, allowing us to retrieve tweets, user profiles, hashtags, and replies. This facilitated the collection of relevant data to analyze the online presence and customer sentiment of the travel agencies.

Advantages of snscrape in Data Collection:

- Comprehensive Data Extraction: Snscrape allows for the extraction of various types of data from Twitter, providing a more holistic view of the travel agencies' online activity.
- Flexible Querying: Snscrape enables querying based on usernames, hashtags, and keywords, allowing targeted data collection.
- Authentic Data Retrieval: Data collected through snscrape is directly retrieved from Twitter, ensuring the authenticity and integrity of the dataset.
- User-friendly Interface: Snscrape offers a user-friendly interface with clear documentation, making it accessible to users with varying levels of expertise.
- Active Community and Support: Snscrape benefits from an active community, providing ongoing support and updates for improved data collection processes.

This supported the retrieval of data from Twitter emphasizing on the tweets that are in English, posted from 1st January 2017 to the current date. The specific date range was used since the tweets available were limited, and it was necessary to expand the date range to capture more tweets for analysis.

The below figure shows the process of installing Snscrape and importing the necessary libraries for data processing and visualization. The defined parameters for data collection were username, content and date which was iterated over the search results to collect the desired data. The **snscrape** function **twitter.SearchScraper** was used to search for tweets matching a query and retrieve specific attributes of the tweets. The tweets were collected and stored in a Pandas DataFrame.

Once the data was collected in a DataFrame, it was saved in a csv file using the **"to-csv"** function provided by Pandas.

```
# Initialize empty list to store data
tweets_list = []
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('UnchartedKenya since:2017-01-01 until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }

    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('uncharted.csv', index=False)
```

Figure 5.1: Python code used to Save CSV

Once the data from all the 9 travel agencies was retrieved and saved in csv files, the data frames were merged into one dataset.

```
Tweets_df = pd.DataFrame()

# append the CSV files
for file in csv_files:
    df = pd.read_csv(file, lineterminator='\n')
    Tweets_df = Tweets_df.append(df, ignore_index=True)
```

Figure 5.2: Python code used to append CSV

5.2.2 Preprocessing of Tweets

These preprocessing steps help in reducing noise and standardizing the text data, making it suitable for sentiment analysis and modeling.

	Username	Content	Date
0	hawksboysoccer	Game change alert!!!Hawks travel to @PSHSMensS...	2023-01-30 13:30:03+00:00
1	ChrisRavenUK	@ebook_travel Thanks again Etg :)	2023-01-17 15:11:52+00:00
2	saad_kkm	@Puke2Earn @joinMovEX @hdkdocunc @ETGtravel @f...	2022-12-21 17:32:57+00:00
3	LunenburgFootb1	#15 Lunenburg (3-3) travels to #1 West Boylsto...	2022-10-21 18:43:11+00:00
4	AFTAOfficial	What an aerial display of skill at #NTIA2022! ...	2022-10-15 09:29:47+00:00
5	StockIdeas9	Making money is a good habit. What else could ...	2022-09-15 15:15:35+00:00
6	TheBountyHunt17	Watch "Yungeen Gang - "Go To War" (Official Mu...	2022-08-29 13:08:48+00:00
7	CarlMor20340998	@IAmENISA Travel around the world with ETG	2022-08-09 18:30:43+00:00
8	TTGMedia	The boss of Sri Lanka specialist Experience Tr...	2022-05-17 11:00:01+00:00
9	travelogafrica	Exit Express Travel Group, Enter Hemingways Tr...	2022-04-06 14:15:00+00:00
10	CapitalFMKenya	Sixty-Five-year-old Travel Management Company,...	2022-04-01 05:05:03+00:00
11	BiznaKeOfficial	Express Travel Group (ETG) Rebrands to Hemingw...	2022-04-01 04:56:11+00:00
12	News_Kenya	[BUSINESS] Express Travel Group rebrands to He...	2022-03-31 07:42:41+00:00
13	Blogs_Kenya	Express Travel Group (ETG) Rebrands To Hemingw...	2022-03-30 18:10:38+00:00
14	CapitalFMKenya	Travel Management Company, Express Travel Grou...	2022-03-30 14:27:07+00:00
15	worldofleedham	ETG members to benefit from new NDC connect pl...	2021-12-14 08:43:50+00:00

Figure 5.3: Raw Tweets Data

After collecting the tweet data, the next step was to preprocess the data to prepare it for sentiment analysis and modeling. The preprocessing steps involved:

Cleaning the text: Removing URLs, special characters, and punctuation marks from the tweets

```
# # Removing links (http | https)

cleaned_tweets = []

for index, row in df_clean.iterrows():
    # Here we are filtering out all the words that contains link
    words_without_links = [word for word in row.tidy_tweets.split() if 'http' not in word]
    cleaned_tweets.append(' '.join(words_without_links))

df_clean['tidy_tweets'] = cleaned_tweets
df_clean.head(10)
```

Figure 5.4: Python code used to drop links

Tokenization: Splitting the text into individual words or tokens.

```
tokenized_tweet = df_clean['absolute_tidy_tweets'].apply(lambda x: x.split())
tokenized_tweet.head()
```

Figure 5.5: Python code used to tokenize the tweets

Stop word Removal: Removing common words (e.g., "the", "is", "and") that do not carry significant meaning.

```
#removing stopwords
from nltk.corpus import stopwords
Tweets_df["combined_features"] = Tweets_df["combined_features"].str.lower().str.split()
stop = stopwords.words('english')
Tweets_df['combined_features']=Tweets_df['combined_features'].apply(lambda x: [item for item in x if item not in stop])
Tweets_df["combined_features"]= Tweets_df["combined_features"].str.join(" ") #rejoining the words to text
```

Figure 5.6: Python code used to remove stop words

Lemmatization or Stemming: Reducing words to their base form to eliminate variations (e.g., "running" to "run").

```
word_lemmatizer = WordNetLemmatizer()

tokenized_tweet = tokenized_tweet.apply(lambda x: [word_lemmatizer.lemmatize(i) for i in x])
tokenized_tweet.head()
```

Figure 5.7: Python code used for Lemmatization or Stemming

Missing Values Removal: Null values are dropped to ensure data completeness.

```
#Remove null values
clean_tweets = Tweets_df.dropna()
clean_tweets.head()
```

Figure 5.8: Python code used to drop null values

Duplicate Values Removal: Duplicates are removed to ensure data completeness.

```
#dropping  any duplicates

df_clean =clean_tweets.drop_duplicates()
df_clean.shape
```

Figure 5.9: Python code used to drop duplicates

Two columns were created from extraction information from the tweets which are amount and destination. All rows that have null values in either of the columns are dropped to ensure the dataset is complete and can provide insights correctly.

5.2.2.1 Sentiment Analysis

Sentiment analysis was done by importing the Natural Language Toolkit (NLTK) and hosting features in the Vader lexicon. VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon-based sentiment analysis tool that uses a pre-built sentiment lexicon to analyze the sentiment of text. The VADER lexicon contains a list of words and phrases that have been assigned a polarity score based on their sentiment. The polarity score is a continuous value ranging from -1 to 1, where -1 indicates very negative sentiment, 0 indicates neutral sentiment, and 1 indicates very positive sentiment. The lexicon also includes rules to handle negation and intensifiers, which modify the polarity score of words that precede.

To perform sentiment analysis using VADER, the text is first preprocessed to remove noise and tokenize it into individual words. Each word is then looked up in the VADER lexicon, and its polarity score is retrieved. The polarity scores of all the words in the text are then combined to give an overall sentiment score for the text. VADER also considers the presence of emoticons, exclamation points, and capitalization, as these can provide additional clues about the sentiment of the text. The output of VADER is a sentiment score ranging from -1 to 1, along with scores for positive, negative, and neutral sentiment.

The figure below shows the results of the sentiment analysis according to the three categories. Positive tweets lead followed by neutral tweets and lastly negative ones.

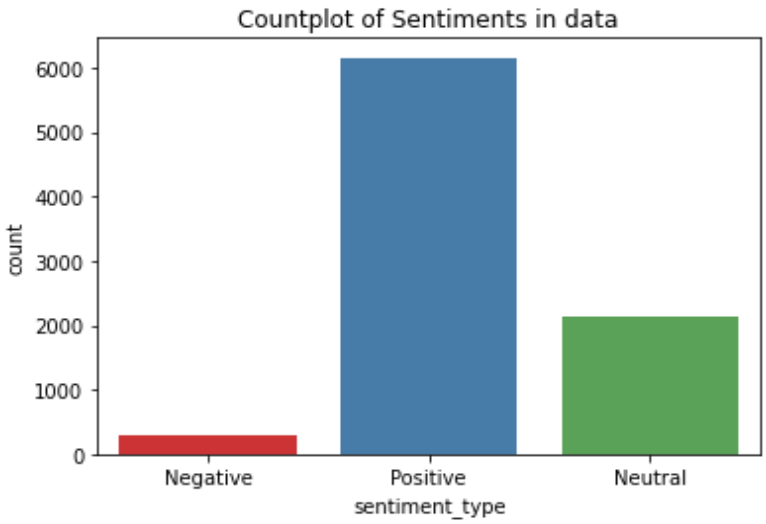


Figure 5.10: Number of tweets per sentiment type

To ensure the accuracy and generalizability of the machine learning model, the dataset was sampled to ensure that it was balanced. This was done to reduce the impact of any bias or skewedness in the data that could result in a model that only performs well on a specific subset of the data. To achieve this, we used a random sampling technique to select equal numbers of data points from each class in the dataset. This helped to ensure that the machine learning model was trained on a dataset that was representative of the entire population, and that the model could accurately classify instances from each class. By balancing the dataset, we were able to reduce the risk of overfitting, and the model was able to generalize better to unseen data.

```python
# Group the dataset by sentiment type
grouped_df = df_clean.groupby('sentiment_type')

# Initialize an empty dataframe to store the randomly selected rows
random_df = pd.DataFrame()

# Iterate through each sentiment type
for name, group in grouped_df:
    # Select 13 rows randomly
    sample = group.sample(n=13, random_state=42)
    # Append the selected rows to the random_df dataframe
    random_df = pd.concat([random_df, sample])

# Save the randomly selected rows as a separate dataset
random_df.to_csv('random_dataset.csv', index=False)
```

Figure 5.11: Generating equal sample for each sentiment type

5.2.2.2 Word Clouds

Word clouds are popularly used to visualize qualitative data. They are popular mostly because they are simple to use, and they give very meaningful insights just from a glance. One can identify different topics or subject matter by just glancing at different words displayed on the word cloud.

The positive word cloud shows the most critical information that depicts a positive tweet. As you can see words like, “Enjoy”, “Free”, “Soft”, “Exclusive”, “Luxury” and “Coast” stand out since they were used more frequently in the positive tweets.

Figure 5.12: Most frequent positive tweets

For the negative word cloud, it is evident that “Hate”, “Hot”, “Few” and “Meal” stand out as key words that depict customer dissatisfaction of services offered by travel agencies.

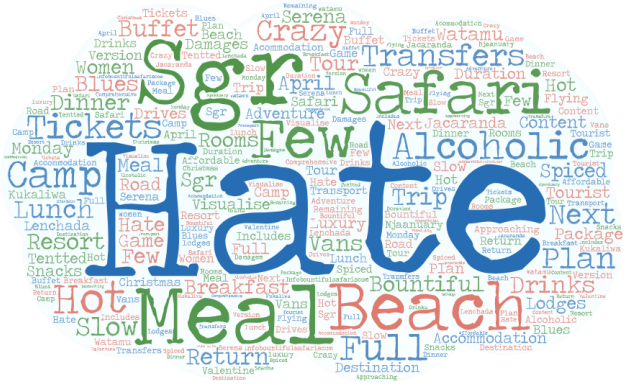


Figure 5.13: Most frequent negative tweets

You could easily spot the similarities and differences between the positive and neutral word cloud. “SGR”, “Coast”,” Beach”,” Meal”,” Ticket”, ” Board” are major words prominent in the neutral word cloud but “Coast” are common words in both categories.

Figure 5.14: Most frequent neutral tweets

Learning what customers like and don’t like about your brand assists in getting insights beyond word frequency, and can lead to informed decision making when combined with other text analysist tools such as Machine Learning algorithms.

5.3 Data Modelling

After sentiment analysis, the preprocessed data was used to build models for classification. Three different models were implemented: K-Nearest Neighbors (KNN), Naive Bayes, and Support Vector Machines (SVM).

These models were trained on the preprocessed data with known sentiment labels to learn the patterns and relationships between the text features and sentiment. The performance of each model was evaluated using appropriate metrics such as accuracy, precision, and F1-score.

5.3.1 Vectorization

In this study, we are recommending the travel agency based upon “tags”. It identifies similarities between the tags of two travel agencies which is supported by vectorization of each of the tags.

Vectorization is an approach for converting input data from its raw format into vectors of real numbers which is a format that is supported by machine learning. Feature extraction is a step in vectorization that assists to get some distinct features out of the text for the model to train on, by converting text to numerical vectors. There are various vectorization techniques which include Bag of Words, TF-IDF and GloVe. For this study, we opted to use TF-IDF technique.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score


# Initialize the TF-IDF vectorizer
vectorizer = TfidfVectorizer()
```

Figure 5.15: Vectorization using TF-IDF

5.3.2 Training Data

A train-test split procedure is done to estimate the performance of the machine learning algorithms for it to make predictions on data that was not used to train the model. The procedure involves taking a dataset and dividing into two subsets, X-train and y-train for training which includes the actual and expected values then X-test and y-test are used as test data. The training set is used to

fit the model while the test data is used as the test data where the input element of the dataset is provided to the model. The predictions are made and compared to the expected values.

The procedure contains a configuration parameter of 100 percent where the training set has a size of 80 percent and the remainder percentage of 20 percent is used as a test set.

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(Tweets_df['Content'], Tweets_df['sentiment_type'], test_size=0.2, random_state=42)
```

Figure 5.16: Splitting of the dataset to train and test data.

5.3.3 Performance evaluation

In this section, we will explore using the train-test split procedure to evaluate the machine learning algorithms on classification datasets. This will be explored in the Naïve Bayes algorithm, Support Vector Machine and K-Nearest Neighbor (KNN) algorithm.

5.3.3.1 Naïve Bayes

Naive Bayes: Naive Bayes is a probabilistic algorithm that applies Bayesian theorem with the assumption of independence between features. It calculates the probability of a tweet belonging to a specific sentiment class based on the occurrence of words in the tweet.

The loaded dataset is split into input and output components. We then split the dataset so that 80 percent is used to train the model and 20 percent is used to evaluate it. The split was chosen arbitrarily.

We went ahead to define and fit the Naïve Bayes algorithm on the training dataset. Then the fit model was used to make predictions using several classification performance metrics namely accuracy, precision, F1 score and confusion matrix. Tying all this together, the evaluation procedure was as shown below:

```
# Initialize the Naive Bayes classifier
naive_bayes = MultinomialNB()

# Perform grid search for hyperparameter tuning
grid_search = GridSearchCV(naive_bayes, param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train_vectorized, y_train)

# Get the best hyperparameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

# Predict the sentiment of the testing data using the best model
y_pred = best_model.predict(X_test_vectorized)

# Evaluate the accuracy of the best model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8292682926829268
```

Figure 5.17: Initialization and prediction using the Naïve Bayes Model

Our model achieved an overall accuracy of ~0.829. This result seems to be good, as it shows, 82.9 percent of the tweets were classified correctly with the right sentiment type. If we look at the class level predictions using confusion matrix and F1 score.

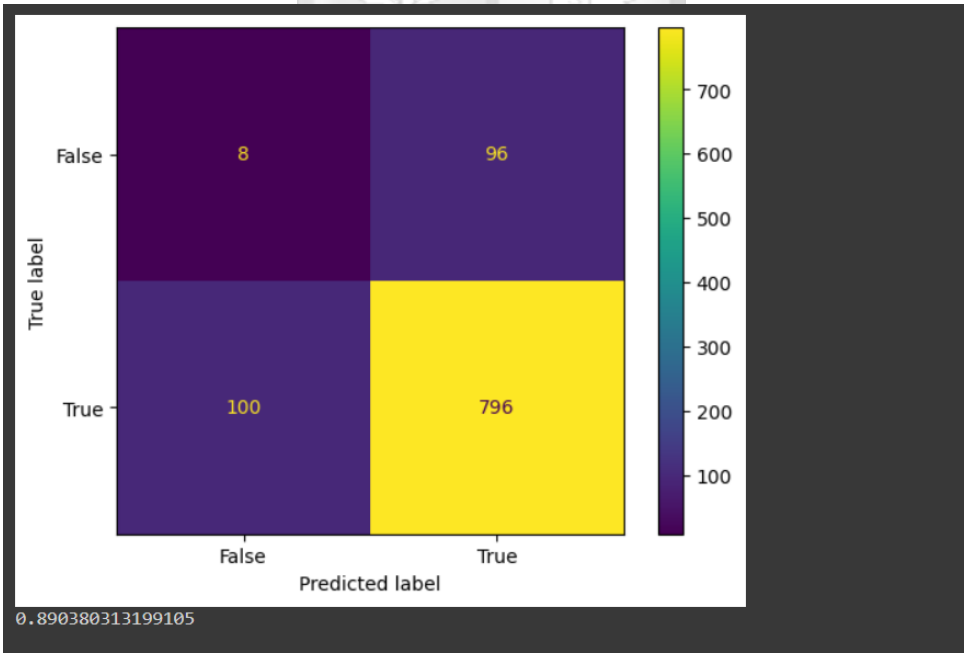


Figure 5.18: Confusion matrix and F1 score graphical representation

The results are almost the same as what we expected based on the overall accuracy metric as the algorithm achieved an F1 score of 0.89.

The table below shows the results of our Naïve Bayes classification model by quantifying the number of correct and incorrect predictions for each class.

True Positive (Non - Radical)	80%
False Positive (Non - Radical)	10%
True Negative (Radical)	0.8%
False Negative (Radical)	10%

Table 5.1: Classification of values using Naïve Bayes

The true positive, false positive, true negative and false negative values were obtained and used to calculate the precision using the formula:

Precision = True Positives / (True Positives + False Positives)

Precision = 796/(796 + 96)

Precision = 0.89

The results from the various performance metrics suggest that the Naive Bayes model performs reasonably well in distinguishing between positive and negative sentiments. This is summarized in the table below.

Accuracy	**F-Score**	**Precision**
0.829	0.89	0.89

Table 5.2: Performance of the Naive Bayes Algorithm

5.3.3.2 Support Vector Machine (SVM)

Support Vector Machines (SVM): SVM is a supervised learning algorithm that creates a hyperplane in a high-dimensional feature space to separate different classes. It aims to find the optimal hyperplane that maximizes the margin between the classes.

As narrated under Naives Bayes algorithm, the loaded dataset is split into input and output components. 80 percent is used to train the model and 20 percent is used to evaluate it.

We went ahead to define and fit the Support Vector Machine (SVM) on the training dataset. The fit model was used to make predictions using several classification performance metrics namely accuracy, precision, F1 score and confusion matrix. Tying all this together, the evaluation procedure was as shown below:

```
# Train the SVM classifier
classifier = SVC(kernel='linear')
classifier.fit(X_train_vectorized, y_train)

# Transform the testing data
X_test_vectorized = vectorizer.transform(X_test)

# Predict the sentiment of the testing data
y_pred = classifier.predict(X_test_vectorized)

# Evaluate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8048780487804879
```

Figure 5.19: Initialization and prediction using the Support Vector Machine Model

Our model achieved an overall accuracy of ~0.80. This result was closely related to Naïve Bayes, as it shows 80 percent of the tweets were classified correctly with the right sentiment type.

If we look at the class level predictions using confusion matrix and F1 score.

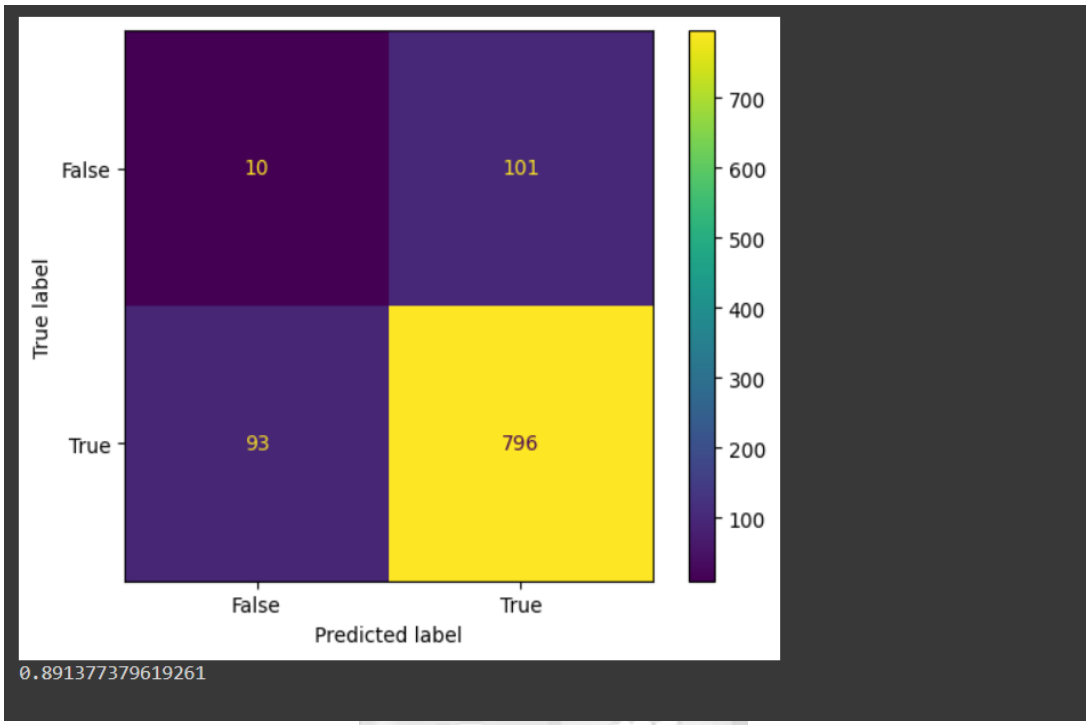


Figure 5.20: Confusion matrix and F1 score graphical representation

The results are almost the same as what we expected based on the overall accuracy metric as the algorithm achieved an F1 score of 0.89.

The table below shows the results of our Support Vector Machine classification model by quantifying the number of correct and incorrect predictions for each class.

True Positive (Non - Radical)	80%
False Positive (Non - Radical)	9%
True Negative (Radical)	1%
False Negative (Radical)	10%

Table 5.3: Classification of values under Support Vector Machine

The true positive, false positive, true negative and false negative values were obtained and used to calculate the precision using the formula:

Precision = True Positives / (True Positives + False Positives)

Precision = 796/(796 + 93)

Precision = 0.89

The results from the various performance metrics suggest that the Support Vector Machine model also performs reasonably well as seen from the results of the F-Score and Precision metrics that are similar to Naïve Bayes algorithm. This is summarized in the table below.

Accuracy	**F-Score**	**Precision**
0.80	0.89	0.89

Table 5.4: Performance of the Support Vector Machine Algorithm

5.3.3.3 K-Nearest Neighbor (KNN)

K-Nearest Neighbors (KNN): KNN is a non-parametric algorithm that classifies data based on the majority vote of its neighbors. It assigns a label to a data point based on the labels of its k nearest neighbors in the feature space.

As narrated by the two previous algorithms, the loaded dataset is split into input and output components. 80 percent is used to train the model and 20 percent is used to evaluate it.

We went ahead to define and fit the K-Nearest Neighbor on the training dataset. The fit model was used to make predictions using several classification performance metrics namely accuracy, precision, F1 score and confusion matrix. Combining all this together, the evaluation procedure was as shown below:

```
# Initialize the k-NN classifier
knn = KNeighborsClassifier()

# Perform grid search for hyperparameter tuning
grid_search = GridSearchCV(knn, param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train_vectorized, y_train)

# Get the best hyperparameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

# Predict the sentiment of the testing data using the best model
y_pred = best_model.predict(X_test_vectorized)

# Evaluate the accuracy of the best model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.7317073170731707
```

Figure 5.21: Initialization and prediction using the K-Nearest Neighbor (KNN) Model

Our model achieved an overall accuracy of ~0.73. 73 percent of the tweets were classified correctly with the right sentiment type. This result was comparatively lower than Naïve Bayes and Support Vector Machine, which had 82 and 80 percent. If we look at the class level predictions using confusion matrix and F1 score

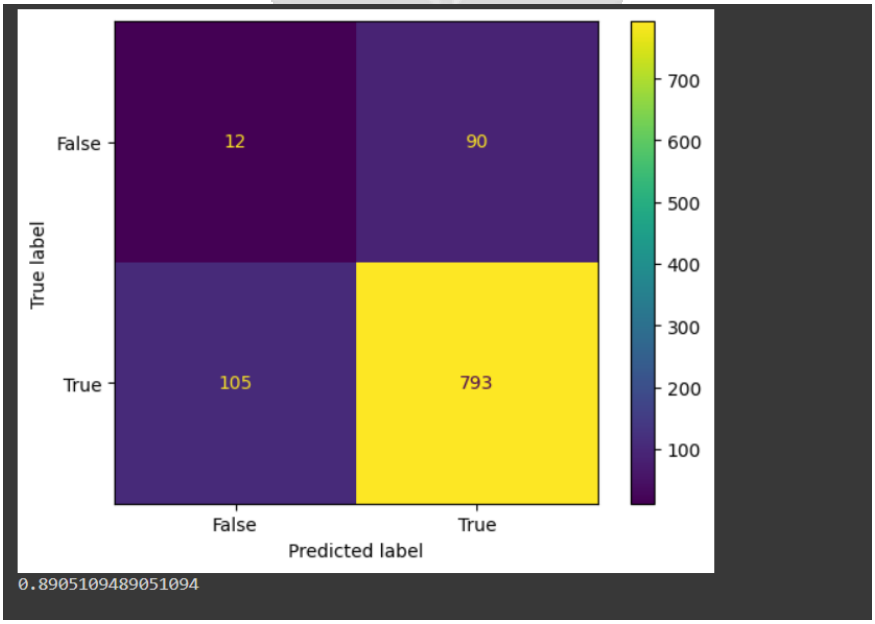


Figure 5.22: Confusion matrix and F1 score graphical representation

The results are almost the same as what we expected based on the overall accuracy metric as the algorithm achieved an F1 score of 0.89.

The table below shows the results of our K-Nearest Neighbor (KNN) classification model by quantifying the number of correct and incorrect predictions for each class.

True Positive (Non - Radical)	79%
False Positive (Non - Radical)	11%
True Negative (Radical)	1%
False Negative (Radical)	9%

Table 5.5: Classification of values under K-Nearest Neighbor (KNN)

The true positive, false positive, true negative and false negative values were obtained and used to calculate the precision using the formula:

Precision = True Positives / (True Positives + False Positives)

Precision = 793/(793 + 105)

Precision = 0.88

The results from the various performance metrics suggest that the K-Nearest Neighbor (KNN) model also performs reasonably well as seen from the results of the F-Score and Precision metrics that are similar or related to Naïve Bayes algorithm. This is summarized in the table below.

Accuracy	F-Score	Precision
0.73	0.89	0.88

Table 5.6: Performance of the K-Nearest Neighbor (KNN) Algorithm

5.4 Model Findings & Results

In the research study, the performance of three different models, namely Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM), was evaluated for a specific task. The evaluation results indicated that the Naive Bayes model outperformed the other models and was found to be the best performing among them as shown below:

Model Classification	Accuracy	F-Score	Precision
Naïve Bayes	0.829	0.89	0.89
Support Vector Machine	0.80	0.89	0.89
K-Nearest Neighbor	0.73	0.89	0.88

Table 5.7: Performance of the three models

5.5 Hybrid Filtering Approach

A hybrid filtering approach is implemented as it combines multiple recommendation techniques or filtering methods to leverage their respective strengths and overcome their limitations. It combines content-based filtering using cosine similarity and Naive Bayes and user-based collaborative filtering. By integrating the different methods, a hybrid approach can provide more accurate, diverse, and personalized recommendations.

Content-based filtering using cosine similarity is a technique employed in recommender systems to provide personalized recommendations based on the similarity between the content or attributes

of items and the user's preferences. Cosine similarity is used to calculate the similarity between travel agencies based on the text content of their tweets. This allows for the identification of agencies that have similar content and can provide similar services.

After data modelling, the distance between two travel agencies was calculated using the cosine similarity. Lesser the distance, the more the similarity between them.

```
[35] #data.replace('', np.nan, inplace=True)
     cv = CountVectorizer() #creating new CountVectorizer() object
     count_matrix = cv.fit_transform(Tweets_df['combined_features'].values.astype('U'))
     cosine_sim = cosine_similarity(count_matrix) #calculating cosine similarity
```

Figure 5.23: Cosine Similarity

From the figure below, the similarity of every destination with every destination is represented.

```
[36] cosine_sim

     array([[1.        , 0.81235996, 0.20293096, ..., 0.10848296, 0.10318787,
             0.13650473],
            [0.81235996, 1.        , 0.17106014, ..., 0.10588418, 0.10071595,
             0.17764624],
            [0.20293096, 0.17106014, 1.        , ..., 0.08055324, 0.07662142,
             0.06757374],
            ...,
            [0.10848296, 0.10588418, 0.08055324, ..., 1.        , 0.45056356,
             0.17660431],
            [0.10318787, 0.10071595, 0.07662142, ..., 0.45056356, 1.        ,
             0.16798421],
            [0.13650473, 0.17764624, 0.06757374, ..., 0.17660431, 0.16798421,
             1.        ]])
```

Figure 5.24: Cosine Similarity Results

For Collaborative Filtering, the user-based approach is employed to provide personalized recommendations based on user preferences and behaviors. The model applies additional filtering and ranking techniques to refine the recommendations based on specific criteria, such as destination. This ensures that the recommendations are tailored to the user's individual needs and constraints.

```
df = pd.DataFrame(Tweets_df)
#Collaborative Filtering
collab_matrix = df.pivot_table(index='Username', columns='Destination', values='amount', aggfunc='first').fillna(0)
cosine_sim_collab = cosine_similarity(collab_matrix)
```

Figure 5.25: Collaborative Filtering Collab Matrix

Once all the results were received, the model was converted into a web application.

5.6 Travel Agency Recommender System with Streamlit

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. It allows developers to build and deploy powerful data applications in less time.

To create the recommender system, the system was integrated with the implemented model and a list of destinations in a csv file. This process was supported by a pickle module used for serializing and de-serializing python object structures. A text input widget was added from streamlit to allow users to enter their desired Kenyan Destination. After clicking on the button, the applications recommend a suitable travel agency, the amount to be paid and number of nights and days the user could stay.

```
# Define the title and description
icon_image = Image.open("header-image.jpg")
st.image(icon_image, use_column_width=True, width=100)
st.title('Travel Agency Recommender')
st.markdown('Welcome to our travel agency recommender system! 👋 This app helps in the suggesting a preferred travel agency based on the selec
# Create an input field for the destination
destination_input = st.text_input('Destination')

# Add a button to trigger the recommendation
if st.button('Recommend'):
    # Perform recommendation based on the input destination
    best_username_output, amount_output, nights_output, days_output = get_recommendation_info(destination_input)

    if best_username_output:
        # Display the recommendation results
        st.subheader('Recommendation Results')
        st.write(f"Destination: {destination_input}")
        st.write(f"Best Username: {best_username_output}")
        st.write(f"Amount: {amount_output}")
        st.write(f"Number of Nights: {nights_output}")
        st.write(f"Number of Days: {days_output}")
    else:
        st.warning(f"No recommendations found for {destination_input}.")
```

Figure 5.26: Streamlit Python structure

The recommender system is evaluated based on its ability to recommend the best travel agency for a given destination. The recommendations provided by the system are compared against ground truth data or user feedback to measure the system's accuracy and effectiveness.

Travel Agency Recommender

Welcome to our travel agency recommender system! This app helps in the suggesting a preferred travel agency based on the selected destination.

Destination

Watamu

Recommend

Recommendation Results

Destination: Watamu

Best Username: BonfireSafaris

Amount: 47500.0

Number of Nights:

Number of Days: 3.0

Figure 5.27: Travel Agency Recommender System

The results of the recommender system are analyzed and discussed. The accuracy of the recommendations, user feedback, and the impact of hybrid filtering using cosine similarity and Naive Bayes are assessed. The analysis provides insights into the effectiveness of the system in recommending the best travel agency.

Chapter 6: Discussion of Results

6.1 Introduction

The study's sixth chapter was dedicated to offering a discussion of the results in keeping with the study's goals.

6.2 Discussion of Findings

A discussion of the results is included in this section. The results are presented in accordance with the study's objectives. The study's main objective was to help travel providers in the e-tourism industry to facilitate and maintain high service quality for their consumers by creating a machine learning tool that accommodates consumers' feedback, reviews, and experiences and recommends the best travel agency.

6.2.1 Influence of Service Quality on Travel Providers.

The study established that service quality is an important factor to consider when choosing a travel provider in the E-tourism industry. From the word clouds, words like, "Enjoy", "Free", "Soft", "Exclusive", "Luxury" and "Coast" stand out since they were used more frequently in the positive tweets. This revealed that customers want to enjoy their experiences while enjoying luxurious and the best available services offered. The analysis also revealed that customers also prefer offers and other free and complementary services and are more likely to select a travel agency based on these services. Additionally, the coast was identified as an ideal destination from many tweets thereby increasing the chances of selecting a travel agency if they offer services to the Coast.

For the negative word cloud, it is evident that "Hate", "Hot", "Few" and "Meal" stand out as key words that depict customer dissatisfaction of services offered by travel agencies. This revealed that most travel agencies customers were dissatisfied with the food situations with meals, buffet, dinner and lunch appearing frequently among the negative tweets. This analysis further revealed dissatisfaction with accommodation facilities offered or referred to by these agencies. This shows that customers are most likely to seek the services of a travel agency which has accommodation services preplanned and quality assured to the customers.

"SGR", "Coast"," Beach"," Meal"," Ticket", " Board" are major words prominent in the neutral word cloud . Analysis of the neutral tweets showed that most customers were content with the

mode of transport used as SGR appeared majorly in neutral tweets. The analysis further revealed that most customers utilized travel agency services in December and their preferred destination is the Coast with Mombasa, Malindi and the Maasai Mara preferred destinations. This therefore shows that for a travel agency to be at par with the competition, travel services to the Coast in December, using the SGR services should be a minimum requirement.

The findings from the study were also supported by O'Connor, Trinh and Shewchuk, (2010) who also noted that Service quality is one of the key measures for analyzing customer satisfaction and that choice of service provider and service quality evaluation are influenced by the expectations of the consumer. These results were also supported by Confente (2015), who pointed out that in today's environment, tourist attractions and providers of these services must pay even closer attention to customer satisfaction due to the quickly changing competitive landscape brought about by recent consumer and technological trends, which make customer satisfaction more crucial than ever. Additionally, Mohd-Any, Winklhofer and Ennew (2015) carried out a study seeking to measure multidimensional user's value experience when using travel firms and determined a positive influence of quality services on e-value satisfaction thereby concluding that high quality service by travel providers significantly influences customer experience which determines brand valuation.

6.2.2 Existing Travel Recommender systems

The researcher also sought to find out the existing travel recommender systems that are utilized in the E-tourism industry. The study found that there are currently other systems available that are able to offer recommender services but are not sufficient to be utilized in the Kenyan context. As noted by Wang (2018), TripAdvisor relies on user-generated content such as reviews and ratings, but it may not capture the most recent experiences and sentiments of travelers. The current developed system will leverage sentiment analysis on social media to provide real-time updates and capture the latest sentiments and opinions shared by travelers. Additionally, Jong-Hyung (2020) noted that TripAdvisor does not fully harness the power of social influence in its recommendation system. The current system will utilize sentiment analysis on social media to identify influential users, experts, or trusted sources within a user's social network, providing recommendations that are influenced by trusted sources and connections.

Kayak is primarily a flight and hotel search platform and may not utilize sentiment analysis techniques extensively and Amadeus is a global distribution system that provides services to travel agencies and also does not extensively utilize sentiment analysis for recommendations. The system will bridge the gap by leveraging sentiment analysis on social media to extract sentiments and opinions related to travel destinations, accommodations, and activities. This will allow for more accurate and context-aware recommendations based on user sentiments. Additionally, Amadeus primarily focuses on providing solutions for travel agencies and may have limited personalization features for individual end-users. Your system can offer highly personalized recommendations that consider individual preferences, sentiments, and social influences, enhancing the end-user's travel planning experience. Finally, Kayak and Amadeus among other recommender systems also do not extensively incorporate user-generated content and sentiment analysis from social media platforms. The current system can leverage sentiment analysis to extract insights from user-generated content, allowing for more accurate recommendations based on the latest and authentic travel experiences shared by users.

6.2.3 Travel Agency Recommender System.

The researcher embarked on designing and developing a system that leverages social media sentiment analysis to provide travelers with accurate and relevant recommendations of travel providers. The researcher developed this system to recommend to the user the best travel agency to use based on their specifications. First, data on customers' feedback, reviews, and experiences was collected from Twitter social media posts. After collection, it was preprocessed to prepare it for analysis which involved tasks such as cleaning the data, removing irrelevant information, and standardizing the format of the data. Sentiment analysis techniques are then applied to the data to determine the overall sentiment of customers' feedback by involving machine learning algorithms to classify text data as positive, negative, or neutral based on the language used. With this sentiment analysis in hand, a decision-making assistant tool that helps travel providers make data-driven decisions based on customers' feedback and experiences is therefore developed.

This system is considered useful to both the customer and the travel providers. For users, by using a travel agency recommender system that incorporates sentiment analysis, travel agencies can improve the quality of their services and provide customers with a more personalized and tailored

travel experience. Additionally, customers can make better-informed decisions about travel options, accommodations, and activities by accessing sentiment analysis data and reviews from other customers, which can help them avoid disappointment and maximize their enjoyment of their trip.

For travel providers, by using a travel agency recommender system that incorporates sentiment analysis, they can identify other travel providers that dominate the industry thus they can improve the quality of their services and enhance customer satisfaction levels, which can lead to repeat business and positive word-of-mouth recommendations. This increases their chances of a repeated result for various destinations. Further, using machine learning and data analytics techniques to analyze customer feedback, travel agencies can identify areas for improvement and streamline their operations, resulting in cost savings and increased efficiency.

6.2.4 Performance of machine learning models

The performance of machine learning models is a critical aspect of any study that involves developing a classification model. In this study, the machine learning algorithms were used to classify and recommend travel agencies namely Naive Bayes, Support Vector Machine and K-nearest neighbor. The algorithms were evaluated using various performance metrics such as accuracy, precision, F1 score, and confusion matrix. The evaluation results indicated that the Naive Bayes model outperformed the other models and was found to be the best performing among them. This is evident from the accuracy scores where Naïve Bayes topped with 0.829 while SVM and K-Nearest Neighbor had 0.80 and 0.73. This showed that accuracy score provided the significant results for the algorithms. F1 score, Confusion Matrix and Precision were also used to evaluate the performance results of the algorithms, the results were similarly good and different from the accuracy score. The confusion matrix number of positives and negatives was evenly classified, indicating that the models had high discriminatory power and could effectively differentiate between positive and negative feedback.

These results mean that the machine learning models developed in this study have significant potential to be used in practice to recommend travel agencies to customers based on their preferences and feedback. By accurately predicting the most suitable travel agency for each customer, the models can enhance customer satisfaction levels and promote customer loyalty.

Content-based filtering and collaborative filtering were both important techniques in building effective travel recommender systems. The results revealed that Content Based filtering helps in providing personalized recommendations by analyzing the attributes provides which in our case was the preferred destination. Content-based filtering relies on the characteristics of items, making it useful for new users or users with limited historical data which is evident from the data collected from 2017 to the current date. By combining content-based filtering and collaborative filtering, hybrid approaches offer more robust and accurate recommendations. They leverage the strengths of both techniques to provide a comprehensive and personalized travel experience. The study results were in line with findings by Xie et al. (2021) who developed a machine learning-based approach for personalized travel recommendations and showed that it significantly enhanced the quality of travel services and customer experiences. Similarly, a study by Kim et al. (2018) developed a machine learning model to predict hotel ratings using online reviews and showed that it achieved high levels of accuracy and could effectively identify the most significant features affecting customer satisfaction. Furthermore, a study by Hsu et al. (2019) developed a machine learning-based approach to analyze online customer reviews and identify the most important factors affecting customer satisfaction with travel services. The study showed that machine learning techniques could effectively analyze large amounts of customer feedback and identify areas for improvement.

6.3 Limitations of the study

The study faced various limitations. There was limited training data since most tweets that were extracted did not have the amount and destination. The accuracy and reliability of sentiment analysis is limited by the quality of the data used to train the machine learning algorithms. Biases in the data or insufficient training data may result in inaccurate sentiment analysis results. This was mitigated by ensuring data is sufficiently sorted and cleaned to improve the quality of the data. The study also acknowledged that gathering and analyzing customer feedback and reviews may raise privacy concerns. This was mitigated by ensuring that any data collection and analysis methods are compliant with ethical and legal requirements. The study also noted that some of the travel agencies are not active on Twitter.

Chapter 7: Conclusions and recommendations

7.1 Introduction

The chapter offered the study's conclusions and suggestions, which were informed by the findings. Finally, ideas for additional studies were offered.

7.2 Conclusions

The study sought to determine the influence of service quality on travel providers in the E-tourism industry and established that service quality is an important factor to consider when choosing a travel provider in the E-tourism industry. The study concluded that it is essential for travel providers to recognize that customers are increasingly relying on online reviews and feedback to make informed travel decisions. Therefore, they should continuously monitor customer feedback, reviews, and experiences using sentiment analysis tools to identify areas for improvement and proactively address any concerns. The Travel Agency Recommender System harnesses the sentiment expressed by users on social media platforms to gauge their preferences and sentiments towards various travel destinations, accommodations, and activities. By incorporating social media sentiment analysis, the system empowers users to make informed decisions based on real-time and authentic user opinions. Travel providers should also focus on providing personalized services that meet the unique needs and preferences of their customers. By utilizing machine learning and data analytics techniques, travel providers can gather valuable insights into customer behavior and preferences, which can be used to create tailored travel packages and recommendations. Moreover, it is important for travel providers to ensure that their services are reliable, safe, and secure. The system effectively harnesses the vast amount of user-generated content available on social media platforms. By extracting sentiments from posts, comments, and reviews, the system taps into the collective intelligence of the online community, providing a rich source of information for travel recommendations.

The study also sought to develop a machine learning tool that accommodates consumers' feedback, reviews and experiences and recommends a travel agency to use. The study concluded that the machine learning tool developed in this study can significantly enhance the quality of travel services and customer experiences, increase customer satisfaction levels, and optimize travel agency operations. in the E-tourism industry. The study determined that by utilizing machine

learning algorithms and sentiment analysis techniques, the tool can accurately predict and recommend the most suitable travel agency for customers based on their preferences, feedback, and experiences. This personalized approach to travel agency recommendation can enhance customer satisfaction levels and promote customer loyalty. Moreover, the tool can continuously monitor and analyze customer feedback and reviews to identify areas for improvement and proactively address any concerns, leading to enhanced service quality and higher customer satisfaction levels. Additionally, the tool can optimize travel agency operations by streamlining resource allocation, reducing costs, and increasing efficiency, leading to better business outcomes and competitive advantages.

7.3 Recommendations

The study concluded that the machine learning tool developed, accommodates consumers' feedback, reviews, and experiences and is important to travel providers in the e-tourism industry to facilitate and maintain high service quality for their consumers. The study, therefore, recommends that travel agents consider using multiple data sources to collect customer feedback, such as online reviews, social media posts, and customer surveys, to ensure that the sentiment analysis is based on a comprehensive and representative sample of customer opinions. The study further recommends that the users of the tool ensure that the sentiment analysis model is trained on a diverse range of customer data to avoid bias and ensure that it accurately reflects the sentiment of the target audience. The study further recommends using additional machine learning techniques, such as natural language processing or clustering, to further analyze social media data and identify patterns and trends in customer feedback to improve the service quality The study also recommends that the ethical implications of using customer data are considered and these firms should ensure that all data collection and analysis is conducted in compliance with relevant data protection laws and regulations. Travel agency firms should also explore the potential for collaboration with travel providers and other stakeholders in the e-tourism industry to further develop and refine the decision-making assistant tool, and to promote the adoption of data analytics and machine learning techniques in the industry.

The study also recommends that users should be encouraged to provide feedback and reviews of their travel experiences through social media and other platforms, which can be used to improve service quality and enhance future travel experiences for themselves and others. The study also

recommends that Customers should adopt sentiment analysis tools to help them make better-informed decisions about travel options, accommodations, and activities, and to identify potential issues or concerns before booking. Further, there should be increased sensitization of customers to make them aware of these tools and give them an opportunity to be open to recommendations from travel agencies and decision-making assistant tools, which can help them discover new and interesting travel options and enhance their travel experiences.

7.4 Area for Further Studies

1. Explore the potential for incorporating other types of data analysis, such as network analysis or text mining, into decision-making assistant tools for travel providers.
2. Conduct comparative studies of different sentiment analysis models and machine learning algorithms to identify the most effective approaches for analyzing customer feedback in the e-tourism industry.
3. Investigate the potential for using sentiment analysis and decision-making assistant tools in other industries, such as hospitality, retail, or healthcare, to improve service quality and customer satisfaction.
4. Conduct a study on the ethical and privacy implications of using customer data for sentiment analysis and decision-making purposes in the e-tourism industry, and develop guidelines and best practices for ensuring data protection and privacy.
5. Explore the potential for incorporating other types of customer data, such as demographic or behavioral data, into decision-making assistant tools to provide more personalized and targeted recommendations for travel providers.

References

Abalo, J., Varela, J. and Manzano, V. (2007) 'Importance values for Importance–Performance Analysis: A formula for spreading out values derived from preference rankings', *Journal of Business Research*, 60(2), pp. 115–121. Doi: 10.1016/j.jbusres.2006.10.009.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Bierman, L. (2021). Random forests. Machine learning, 45(1), 5-32.

B. Schölkopf, A. J. Smola, and F. Bach, (2002). Learning with kernels: support vector machines, regularization, optimization, and beyond, First Edition ed. MIT press

Cambria, E., & Hussain, A. (2022). Applications of semantic analysis technologies in intelligence and security informatics. Information Fusion, 13(4), 241-251.

Cao, L. *et al.* (2009) 'A framework for adapting agile development methodologies', *European Journal of Information Systems*, 18(4), pp. 332–343. Doi: 10.1057/ejis.2009.26.

Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2013). SMOTE Boost: Improving prediction of the minority class in boosting. In European conference on principles of data mining and knowledge discovery (pp. 107-119). Springer.

Cheyne, J., Downes, M. and Legg, S. (2016) 'Travel agent vs internet: What influences travel consumer choices?': *Journal of Vacation Marketing*. Doi: 10.1177/1356766706059307.

Confente, I. (2015) 'Twenty-Five Years of Word-of-Mouth Studies: A Critical Review of Tourism Research', *International Journal of Tourism Research*, 17(6), pp. 613–624. Doi: 10.1002/jtr.2029.

Cronin Jr, J. J., & Taylor, S. A. (2012). Measuring service quality: a reexamination and extension. Journal of marketing, 56(3), 55-68.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2017). Random forests for classification in ecology. Ecology, 88(11), 2783-2792.

Davis, F. D. (1989). Technology acceptance model: TAM. Al-Suqri, MN, Al-Aufi, AS: Information Seeking Behavior and Technology Adoption, 205-219.

Gronroos, C. (1984). A service quality model and its marketing implications. European Journal of Marketing, 18(4), 36-44.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Guttentag, D. A. (2018) 'Why tourists choose Airbnb: A motivation-based segmentation study underpinned by innovation concepts', p. 296.

Hardin, J. (2010). A study of social cognitive theory: The relationship between professional learning communities and collective teacher efficacy in international school settings. Capella University.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.

Hsu, C. L., & Chen, M. C. (2019). Analyzing online customer reviews using sentiment analysis and machine learning: A study of hotel booking websites. Journal of Travel Research.

Hutto, C. J., & Gilbert, E. E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. Eighth International Conference on Weblogs and social media.

Ingaldi, M. (2018) 'Overview of the main methods of service quality analysis', *Production Engineering Archives*, 18(18), pp. 54–59.

Ingaldi, M. and Surkov, K. L. (2016) 'Evaluation of Service Quality in Brewery Using Importance-Performance Analysis', *Acta Technological Agriculture*, 19(1), pp. 24–28.

Jucan, C. and Jucan, M. (2013) 'Travel and Tourism as a Driver of Economic Recovery', *Procedia Economics and Finance*, 6, pp. 81–88. Doi: 10.1016/S2212-5671(13)00117-2.

Kamel, N. *et al.* (2008) 'Tourism demand forecasting using machine learning methods', *ICGST International Journal on Artificial Intelligence and Machine Learning*, 8, pp. 1–7.

Karahanna, E., & Straub, D. W. (1999). The psychological origins of perceived usefulness and ease-of-use. Information & management, 35(4), 237-250.

Kamuzora, P. (2016) 'Non-decision making in occupational health policies in developing countries', *International Journal of Occupational and Environmental Health*, 12(1), pp. 65–71. Doi: 10.1179/oeh.2006.12.1.65.

Kbaier, M. E. B. H., Masri, H., & Krichen, S. (2017). A personalized hybrid tourism recommender system. In 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA) (pp. 244–250). IEEE

Kim, J. W., Lee, C. K., & Lee, H. B. (2018). A study on the prediction of hotel ratings using online reviews. Journal of Travel & Tourism Marketing, 35(6), 759-771.

Kim, K. H., Jang, S. S., & Park, J. Y. (2019). Identifying customer sentiments in online hotel reviews using sentiment analysis and topic modeling. Journal of Travel Research, 0047287519841300.

Konstan, Joseph & Riedl, John. (2012). Recommender systems: From algorithms to user experience. User Modeling and User-Adapted Interaction. 22. 101-123. 10.1007/s11257-011-9112-x.

Ku, E. C. and Chen, C.-D. (2015) 'Cultivating travelers' revisit intention to e-tourism service: the moderating effect of website interactivity', *Behavior & Information Technology*, 34(5), pp. 465–478.

Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167.

Law, R., Leung, R. and Buhalis, D. (2009) 'Information Technology Applications in Hospitality and Tourism: A Review of Publications from 2005 to 2007', *Journal of Travel & Tourism Marketing*, 26(5–6), pp. 599–623. Doi: 10.1080/10548400903163160.

Leung, D., Law, R., Van Hoof, H., & Buhalis, D. (2013). Social media in tourism and hospitality: A literature review. Journal of travel & tourism marketing, 30(1-2), 3-22.

Mamaghani, F. (2009). Impact of e-commerce on travel and tourism: an historical analysis. International Journal of Management, 26(3), 365.

Murphy, K. P. (2006). Naive bayes classifiers. University of British Columbia, 18(60), 1-8.

O'Connor, S., Trinh, H. and Shewchuk, R. (2000) 'Perceptual Gaps in Understanding Patient Expectations for Health Care Service Quality', *Health care management review*, 25, pp. 7–23. Doi: 10.1097/00019514-200109020-00007.

Osman, Z. and Sentosa, I. (2013) 'Mediating effect of customer satisfaction on service quality and customer loyalty relationship in Malaysian rural tourism', *International Journal of Economics Business and Management Studies*, 2(1), pp. 25–37.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1-135.

Paraskevas, A., & Leonidou, L. C. (2018). Measuring service quality in the hotel industry: A study of critical success factors and service quality dimensions. Tourism and Hospitality Research, 13(2), 109-127.

Parasuraman, A., Zeithaml, V. A., & Berry, L. L. (1985). A conceptual model of service quality and its implications for future research. The Journal of Marketing, 49(4), 41-50.

Phipps, M., Ozanne, L. K., Luchs, M. G., Subrahmanyan, S., Kapitan, S., Catlin, J. R., ... & Weaver, T. (2013). Understanding the inherent complexity of sustainable consumption: A social cognitive framework. Journal of Business Research, 66(8), 1227-1234.

Porter, M. E. (2001) 'Strategy and the Internet', *Harvard Business Review*, 1 March. Available at: https://hbr.org/2001/03/strategy-and-the-internet (Accessed: 28 September 2020).

Rennie, J. D. (2001). Improving multi-class text classification with naive Bayes.

Rrmoku, K., Selimi, B., & Ahmedi, L. (2022). Application of Trust in Recommender Systems—Utilizing Naive Bayes Classifier. Computation, 10(1), 6. https://doi.org/10.3390/computation10010006

Stockdale, R. (2007) ‘Managing customer relationships in the self-service environment of e-tourism’, *Journal of vacation marketing*, 13(3), pp. 205–219.

Sureshchandar, G. S., Rajendran, C., & Anantharaman, R. N. (2002). A conceptual model for total quality service. Total Quality Management, 13(2), 261-277.

Tasci, A. D. A., William and Gartner, C. (2007) ‘Destination image and its functional relationship’, *Journal of Travel Research*, pp. 194–223.

Urbaniak, M. (2013) ‘Zastosowanie metody SERVQUAL do oceny jakości usług rekreacyjnych’, *Zeszyty Naukowe Uczelni Vistula*, (32/2013 Ekonomia III), pp. 29–38.

Wang, D., Liang, Y., & Huang, J. (2017). Assessing hotel service quality through analyzing online reviews. Journal of Travel Research, 0047287517737629.

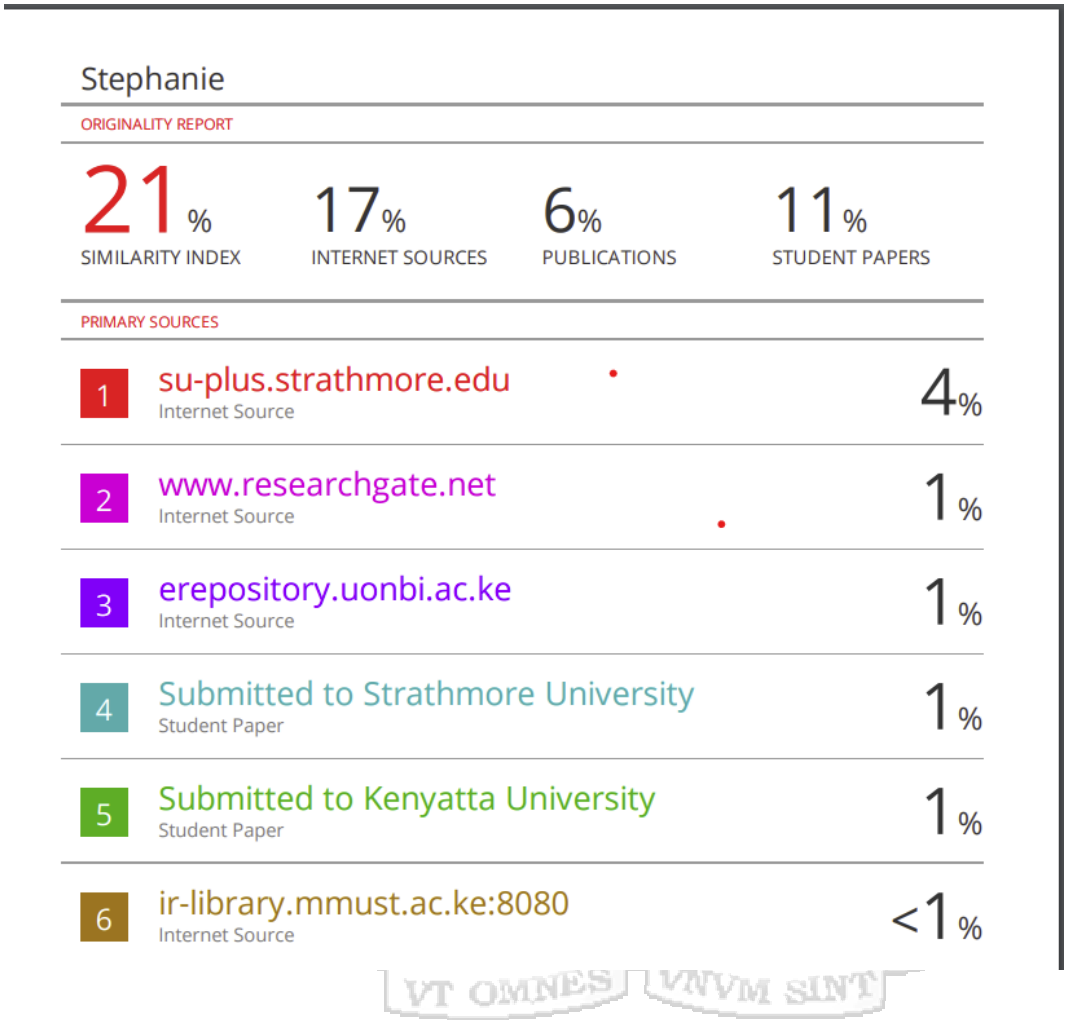
Xiang, Z., Du, Q., Ma, Y., & Fan, W. (2017). A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism. Tourism Management, 58, 51-65.

Xie, K., Lu, L., Zhang, H., & Wang, Y. (2021). Personalized travel recommendation based on machine learning. Journal of Travel Research.

Yagil, D. (2018) ‘Service Relationships: The Impact of Service Providers on Customers’, in Yagil, D. (ed.) *The Service Providers*. London: Palgrave Macmillan UK, pp. 166–185. Doi: 10.1057/9780230582675_9.

Appendices

Appendix A: Originality Report



Stephanie

ORIGINALITY REPORT

21%	17%	6%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	su-plus.strathmore.edu Internet Source	4%
2	www.researchgate.net Internet Source	1%
3	erepository.uonbi.ac.ke Internet Source	1%
4	Submitted to Strathmore University Student Paper	1%
5	Submitted to Kenyatta University Student Paper	1%
6	ir-library.mmust.ac.ke:8080 Internet Source	<1%

VT OMNES VNVM SINT

Appendix B: Ethical Clearance Confirmation

Strathmore UNIVERSITY

27th April 2023

Ms Kingori Stephanie Wambaire,
stephanie.kingori@strathmore.edu

Dear Ms Kingori,

RE: Decision-Making Assistant for Travel Agencies using Sentiment Analysis

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** research proposal. Your application reference number is **SU-ISERC1694/23.** The approval period is from **27th April 2023 to 26th April 2024.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

for: **Mr Ambrose Rachier,**
Chairperson; SU-ISERC

STRATHMORE UNIVERSITY INSTITUTIONAL SCIENTIFIC AND ETHICAL REVIEW COMMITTEE
(SU-ISERC)
27-Apr-2023
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA

Appendix C: Code

1.Code for collecting data

```
# -*- coding: utf-8 -*-
"""Travel Agencies Recommender System

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1hlSkHfjTGB6hMC7O_XF419VRglHDHoSN
"""

# Importing the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import re
import time
import string
import warnings

!pip3 install -U nltk[twitter]

# for all NLP related operations on text
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.stem import WordNetLemmatizer
from nltk.stem.porter import *
from nltk.classify import NaiveBayesClassifier
from wordcloud import WordCloud

from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score, confusion_matrix, accuracy_score
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB

# To identify the sentiment of text
from textblob import TextBlob
from textblob.sentiments import NaiveBayesAnalyzer
from textblob.np_extractors import ConllExtractor

# ignoring all the warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

# Commented out IPython magic to ensure Python compatibility.
# ignoring all the warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

```
# downloading stopwords corpus
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('vader_lexicon')
nltk.download('averaged_perceptron_tagger')
nltk.download('movie_reviews')
nltk.download('punkt')
nltk.download('conll2000')
nltk.download('brown')
stopwords = set(stopwords.words("english"))

# for showing all the plots inline
# %matplotlib inline

# Commented out IPython magic to ensure Python compatibility.
!git clone --depth=1 https://github.com/twintproject/twint.git
# %cd twint
!pip install twint
!pip install aiohttp==3.7.0
!pip3 install . -r requirements.txt

#installing vader
!pip install vaderSentiment

!pip install aiohttp==3.7.0
!pip3 install . -r requirements.txt
import twint
!snscrape twitter-user AITCofficial

!pip3 install nest_asyncio
!pip3 install git+https://github.com/JustAnotherArchivist/snscrape.git
import nest_asyncio
nest_asyncio.apply()# used once to enable concurrent actions within a Jupyter
notebook.

#Installing snscrape
!pip3 install snscrape
import snscrape.modules.twitter as sntwitter
import snscrape.modules.twitter as sntwitter
!pip install snscrape twitter-user AITCofficial

!pip3 install . -r requirements.txt
!pip3 install twint
!pip3 install --user --upgrade
git+https://github.com/twintproject/twint.git@origin/master#egg=twint

!pip3 install twint

# Commented out IPython magic to ensure Python compatibility.
!git clone --depth=1 https://github.com/twintproject/twint.git
# %cd twint
!pip3 install . -r requirements.txt
!pip3 install git+https://github.com/JustAnotherArchivist/snscrape.git

# Initialize empty list to store data
tweets_list = []
```

```
#Scraping tweets that have tagged trippygotours
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('trippygotours since:2017-01-01
until:datetime.now()', top = True).get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }


    # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date,

    }

    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('trippy.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

# Initialize empty list to store data
tweets_list = []
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('UnchartedKenya since:2017-01-01
until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }

    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
```

```
tweets_df.to_csv('uncharted.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

# Initialize empty list to store data
tweets_list = []
#Scraping tweets that have tagged trippygotours
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('Bountifulsafari since:2017-01-01
until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }


    # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }

    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('bountiful.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

# Initialize empty list to store data
tweets_list = []
#Scraping tweets that have tagged trippygotours
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('letsDiscoverke since:2017-01-01
until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }
```

```
    # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }

    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('letsDiscoverke.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

# Initialize empty list to store data
tweets_list = []
#Scraping tweets that have tagged trippygotours
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('BonfireSafaris since:2017-01-01
until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }


    # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }

    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('bonfire.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

# Initialize empty list to store data
tweets_list = []
#Scraping tweets that have tagged trippygotours
```

```
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('lai_group since:2017-01-01
until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }


    # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }

    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('lai_group.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

# Initialize empty list to store data
tweets_list = []
#Scraping tweets that have tagged trippygotours
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('Turnup_Travel since:2017-01-01
until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }


    # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }
```

```
    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('Turnup_Travel.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

# Initialize empty list to store data
tweets_list = []
#Scraping tweets that have tagged trippygotours
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('OribiExpedition since:2017-01-01
until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
     # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }


    # Store data in dictionary
    tweet_dict = {
        "Username": username,
        "Content": content,
        "Date": date
    }

    # Add dictionary to list
    tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('OribiExpedition.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

# Initialize empty list to store data
tweets_list = []
#Scraping tweets that have tagged trippygotours
# Iterate through tweets
for tweet in sntwitter.TwitterSearchScraper('KilianTours since:2017-01-01
until:datetime.now()').get_items():
    # Extract relevant data from the tweet
    username = tweet.user.username
    content = tweet.content
    date = tweet.date
```

```
        # Store data in dictionary
        tweet_dict = {
            "Username": username,
            "Content": content,
            "Date": date
        }


        # Store data in dictionary
        tweet_dict = {
            "Username": username,
            "Content": content,
            "Date": date
        }

        # Add dictionary to list
        tweets_list.append(tweet_dict)

# Convert list of dictionaries to DataFrame
tweets_df = pd.DataFrame(tweets_list)
#Saving as a csv file
tweets_df.to_csv('KilianTours.csv', index=False)
#Previewing the first 20 rows
tweets_df.head(20)

"""#Merging all the csv files """

import glob

# list all csv files only
csv_files = glob.glob('*.{}'.format('csv'))
csv_files

import pandas as pd

Tweets_df = pd.DataFrame()

# append the CSV files
for file in csv_files:
    df = pd.read_csv(file, lineterminator='\n')
    Tweets_df = Tweets_df.append(df, ignore_index=True)
#Previewing the first five rows of the combined dataset
Tweets_df.head()

#Previewing the last five rows of the dataset
Tweets_df.tail()

#Saving the merged csv file
Tweets_df.to_csv('comprehensive_df.csv')

"""##Reading the csv file"""

import pandas as pd
Tweets_df = pd.read_csv("comprehensive_df.csv")

Tweets_df.shape
```

```
Tweets_df.columns

Tweets_df.head()
```

2. Data Cleaning

```
import pandas as pd
import numpy as np
import re
import nltk
import warnings
warnings.filterwarnings("ignore")
df = pd.read_csv('/content/comprehensive_df.csv')
df.head(10)

"""##Data Pre-Processing"""

# create new column with extracted values
df['Nights'] = df['Content'].str.extract(r'(\d+)\s+night').astype(float)
df.head()

# create new column with extracted values
df['Days'] = df['Content'].str.extract(r'(\d+)\s+day').astype(float)
df.head()

# Remove columns
Tweets_df = df.drop(['Year', 'Month', 'Day', 'Date', 'Unnamed: 0'], axis=1)
Tweets_df.head()

#Joining the columns to make a unique text column which defines a package ---
The cosine similarity of this column is used to predict recommendations
features = ['Destination','Content','amount', 'Username', 'Days', 'Nights']
def combine_features(row):
 return row['Destination']+" "+row['Username']+" "+row['Content']
for feature in features:
    Tweets_df[feature] = Tweets_df[feature].fillna('') #filling all NaNs with
blank string
Tweets_df["combined_features"] = Tweets_df.apply(combine_features,axis=1)
#applying combined_features() method over each rows of dataframe and storing
the combined string in "combined_features" column
#data.head(50) ---seeing the data

#downloading stopwords to remove the common words
nltk.download('stopwords')
from nltk.corpus import stopwords

#Function to do basic text cleaning
def clean(text):

    # Urls
    text = re.sub(r"https?:\/\/t.co\/[A-Za-z0-9]+", "", text)

    # Words with punctuations and special characters
```

```
    punctuations =['@','#','!','?','+','&','*','[',']','-
','%','.',':','/','(',')',';','$','=','>','<','|','{','}','^']
    for p in punctuations:
        text = text.replace(p, f' {p} ')

    return text

Tweets_df.head()

Tweets_df.columns

#removing stopwords
from nltk.corpus import stopwords
Tweets_df["combined_features"] =
Tweets_df["combined_features"].str.lower().str.split()
stop = stopwords.words('english')
Tweets_df['combined_features']=Tweets_df['combined_features'].apply(lambda x:
[item for item in x if item not in stop])
Tweets_df["combined_features"]= Tweets_df["combined_features"].str.join(" ")
#rejoining the words to text

# Rearrange columns
new_order = ['Destination', 'Username', 'amount', 'Nights', 'Days',
       'Content', 'combined_features']
Tweets_df = Tweets_df[new_order]

Tweets_df.describe()

Tweets_df.head()

Tweets_df.tail()

Tweets_df.info()
```

3. Sentiment Analysis


```
"""## Fetch Sentiments Using NTLK Sentiment analyzer(Vader)"""

!pip install vaderSentiment
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()
def sentiment_analyzer_scores(sentence):
    score = analyser.polarity_scores(sentence)
    print("{:-<40} {}".format(sentence, str(score)))

Tweets_df['scores'] = Tweets_df['Content'].apply(lambda Description:
analyser.polarity_scores(Description))
Tweets_df.head()

# Compound score evaluation and giving the sentiments labels

Tweets_df['compound'] = Tweets_df['scores'].apply(lambda score_dict:
score_dict['compound'])
Tweets_df['sentiment_type']=''
Tweets_df.loc[Tweets_df.compound>0,'sentiment_type']='Positive'
```

```
Tweets_df.loc[Tweets_df.compound==0,'sentiment_type']='Neutral'
Tweets_df.loc[Tweets_df.compound<0,'sentiment_type']='Negative'

Tweets_df.tail(50)

# Displaying the sentiments on a countplot

import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x = 'sentiment_type',data = Tweets_df,palette='Set1' )
plt.title('Countplot of Sentiments in data')
plt.figure(figsize=(10,10),dpi=100)
plt.show()

# Frequency table
Tweets_df.sentiment_type.value_counts()

# Group the dataset by sentiment type
grouped_df = Tweets_df.groupby('sentiment_type')

# Initialize an empty dataframe to store the randomly selected rows
random_df = pd.DataFrame()

# Iterate through each sentiment type
for name, group in grouped_df:
    # Select 13 rows randomly
    sample = group.sample(n=12, random_state=42)
    # Append the selected rows to the random_df dataframe
    random_df = pd.concat([random_df, sample])

# Save the randomly selected rows as a separate dataset
random_df.to_csv('random_dataset.csv', index=False)

random_df.shape

random_df.to_csv('random.csv')

random_df.head(10)
```

4. Machine Learning Algorithms

```
"""##Data Modelling

##Naive Bayes
"""

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score


# Split the data into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(Tweets_df['Content'],
Tweets_df['sentiment_type'], test_size=0.2, random_state=42)

# Initialize the TF-IDF vectorizer
vectorizer = TfidfVectorizer()

# Fit and transform the training data
X_train_vectorized = vectorizer.fit_transform(X_train)

# Transform the testing data
X_test_vectorized = vectorizer.transform(X_test)

# Define the parameter grid for hyperparameter tuning
param_grid = {
    'alpha': [0.1, 0.5, 1.0],  # Smoothing parameter
    'fit_prior': [True, False]  # Whether to learn class prior
probabilities or not
}

# Initialize the Naive Bayes classifier
naive_bayes = MultinomialNB()

# Perform grid search for hyperparameter tuning
grid_search = GridSearchCV(naive_bayes, param_grid, cv=5,
scoring='accuracy')
grid_search.fit(X_train_vectorized, y_train)

# Get the best hyperparameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

# Predict the sentiment of the testing data using the best model
y_pred = best_model.predict(X_test_vectorized)

# Evaluate the accuracy of the best model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

X_test_vectorized

import matplotlib.pyplot as plt
import numpy
from sklearn import metrics

actual = numpy.random.binomial(1,.9,size = 1000)
predicted = numpy.random.binomial(1,.9,size = 1000)

confusion_matrix = metrics.confusion_matrix(actual, predicted)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
from sklearn.metrics import f1_score
```

```
#calculate F1 score
f1_score(actual, predicted)

from sklearn.metrics import f1_score

#calculate F1 score
f1_score(actual, predicted)

"""##Support Vector Machine"""

from sklearn.svm import SVC

# Define the parameter grid for hyperparameter tuning
param_grid = {
    'C': [0.1, 1, 10],  # Regularization parameter
    'kernel': ['linear', 'rbf'],  # Kernel type
    'gamma': ['scale', 'auto']  # Kernel coefficient
}

# Initialize the SVM classifier
svm = SVC()

# Perform grid search for hyperparameter tuning
grid_search = GridSearchCV(svm, param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train_vectorized, y_train)

# Get the best hyperparameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

# Predict the sentiment of the testing data using the best model
y_pred = best_model.predict(X_test_vectorized)

# Evaluate the accuracy of the best model
accuracy = accuracy_score(y_test, y_pred)

# Train the SVM classifier
classifier = SVC(kernel='linear')
classifier.fit(X_train_vectorized, y_train)

# Transform the testing data
X_test_vectorized = vectorizer.transform(X_test)

# Predict the sentiment of the testing data
y_pred = classifier.predict(X_test_vectorized)

# Evaluate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

import matplotlib.pyplot as plt
import numpy
from sklearn import metrics

actual = numpy.random.binomial(1,.9,size = 1000)
predicted = numpy.random.binomial(1,.9,size = 1000)
```

```
confusion_matrix = metrics.confusion_matrix(actual, predicted)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
from sklearn.metrics import f1_score

#calculate F1 score
f1_score(actual, predicted)

"""## K-Nearest Neighbors Algorithm"""

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier


# Initialize the TF-IDF vectorizer
vectorizer = TfidfVectorizer()

# Fit and transform the training data
X_train_vectorized = vectorizer.fit_transform(X_train)

# Transform the testing data
X_test_vectorized = vectorizer.transform(X_test)

# Define the parameter grid for hyperparameter tuning
param_grid = {
    'n_neighbors': [3, 5, 7],  # Number of neighbors
    'weights': ['uniform', 'distance'],  # Weighting scheme
    'p': [1, 2]  # Distance metric
}

# Initialize the k-NN classifier
knn = KNeighborsClassifier()

# Perform grid search for hyperparameter tuning
grid_search = GridSearchCV(knn, param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train_vectorized, y_train)

# Get the best hyperparameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

# Predict the sentiment of the testing data using the best model
y_pred = best_model.predict(X_test_vectorized)

# Evaluate the accuracy of the best model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

import matplotlib.pyplot as plt
import numpy
from sklearn import metrics
```

```
actual = numpy.random.binomial(1,.9,size = 1000)
predicted = numpy.random.binomial(1,.9,size = 1000)

confusion_matrix = metrics.confusion_matrix(actual, predicted)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
from sklearn.metrics import f1_score

#calculate F1 score
f1_score(actual, predicted)
```

5. Cosine similarity

```
"""##Applying the Cosine Similarity"""

#import modules
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

#data.replace('', np.nan, inplace=True)
cv = CountVectorizer() #creating new CountVectorizer() object
count_matrix =
cv.fit_transform(Tweets_df['combined_features'].values.astype('U'))
#fitting cv object to combine features column
cosine_sim = cosine_similarity(count_matrix) #calculating cosine
similarity

cosine_sim

#creating new column to merge place and package_name which serves as
user input identifier
Tweets_df.head()
features2 = ['Destination','Username']
def combine_features2(row):
 return row['Destination']+" "+row['Username']
for feature in features2:
    Tweets_df[feature] = Tweets_df[feature].fillna('') #filling all
NaNs with blank string
Tweets_df["place_names"] = Tweets_df.apply(combine_features,axis=1)
Tweets_df["place_names"] = Tweets_df["place_names"].str.lower()

Tweets_df.head()

#function to search user input on the newly created column and return
the index
import re
def get_index_from_title(Destination):
    for ind in Tweets_df.index:
        mond=Tweets_df.iloc[ind]['place_names']
        if re.search(Destination,mond):
            return(ind)
```

```
ind

#finding similar places using cosine similarity
place_selec = "Diani"
place_index = get_index_from_title(place_selec)
#print(place_index)

similar_places = list(enumerate(cosine_sim[place_index])) #accessing
the row corresponding to given place name to find all the similarity
scores for that place name and then enumerating over it

#sorting those packages in descending order
sorted_similar_places = sorted(similar_places,key=lambda
x:x[1],reverse=True)[1:]
#print(sorted_similar_places)

sorted_similar_places

df = pd.DataFrame(Tweets_df)

def get_recommendation_info(destination):
    # Initialize the CountVectorizer
    cv = CountVectorizer()
    count_matrix = cv.fit_transform(df['combined_features'])

    # Compute the cosine similarity based on the count matrix
    cosine_sim = cosine_similarity(count_matrix)

    # Get the index of the input destination
    destination_index = df[df['Destination'] ==
destination]['index'].values[0]

    # Get the similarity scores for the input destination
    sim_scores = list(enumerate(cosine_sim[destination_index]))

    # Sort the destinations based on similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    # Exclude the input destination itself
    sim_scores = [x for x in sim_scores if x[0] != destination_index]

    # Get the top recommended destination
    top_index = sim_scores[0][0]
    best_username = df.loc[top_index]['Username']
    amount = df.loc[top_index]['amount']
    nights = df.loc[top_index]['Nights']
    days = df.loc[top_index]['Days']

    return best_username, amount, nights, days

# Example usage
destination_input = input("Enter the destination: ")
best_username_output, amount_output, nights_output, days_output =
get_recommendation_info(destination_input)
```

```
    if best_username_output:
        print(f"Best Username: {best_username_output}")
        print(f"Amount: {amount_output}")
        print(f"Number of Nights: {nights_output}")
        print(f"Number of Days: {days_output}")
    else:
        print(f"No recommendations found for {destination_input}.")
```

6 Hybrid Filtering Approach

```
df = pd.DataFrame(Tweets_df)
#Collaborative Filtering
collab_matrix = df.pivot_table(index='Username', columns='Destination',
values='amount', aggfunc='first').fillna(0)
cosine_sim_collab = cosine_similarity(collab_matrix)

# Content-Based Filtering
cv = CountVectorizer()
count_matrix = cv.fit_transform(df['combined_features'])
cosine_sim_content = cosine_similarity(count_matrix)
# Hybrid Recommender Function
def get_recommendation_info(destination):
    # Collaborative Filtering
    destination_index = df[df['Destination'] ==
destination]['index'].values[0]
    sim_scores_collab =
list(enumerate(cosine_sim_collab[destination_index]))
    sim_scores_collab = sorted(sim_scores_collab, key=lambda x: x[1],
reverse=True)

    # Content-Based Filtering
    sim_scores_content =
list(enumerate(cosine_sim_content[destination_index]))
    sim_scores_content = sorted(sim_scores_content, key=lambda x: x[1],
reverse=True)

    # Combine the scores using weighted average
    hybrid_scores = [(idx, (0.6 * collab_score + 0.4 * content_score))
                     for (idx, collab_score), (_, content_score) in
zip(sim_scores_collab, sim_scores_content)]

    # Sort the hybrid scores
    hybrid_scores = sorted(hybrid_scores, key=lambda x: x[1],
reverse=True)

    # Exclude the input destination itself
    hybrid_scores = [x for x in hybrid_scores if x[0] !=
destination_index]

    # Get the top recommended destination
    top_index = hybrid_scores[0][0]
    best_username = df.loc[top_index]['Username']
    amount = df.loc[top_index]['amount']
    nights = df.loc[top_index]['Nights']
    days = df.loc[top_index]['Days']
```

```
    return best_username, amount, nights, days
```

7 Web Interface on Streamlit

```
from PIL import Image
import streamlit as st
# Set the page title
# st.set_page_config(page_title='Travel Agency Recommender')

# Define the title and description
icon_image = Image.open("header-image.jpg")
st.image(icon_image, use_column_width=True, width=100)
st.title('Travel Agency Recommender')
st.markdown('Welcome to our travel agency recommender system! ‘ʏThis app
helps in the suggesting a preferred travel agency based on the selected
destination.')
# Create an input field for the destination
destination_input = st.text_input('Destination')

# Add a button to trigger the recommendation
if st.button('Recommend'):
    # Perform recommendation based on the input destination
    best_username_output, amount_output, nights_output, days_output =
get_recommendation_info(destination_input)

    if best_username_output:
        # Display the recommendation results
        st.subheader('Recommendation Results')
        st.write(f"Destination: {destination_input}")
        st.write(f"Best Username: {best_username_output}")
        st.write(f"Amount: {amount_output}")
        st.write(f"Number of Nights: {nights_output}")
        st.write(f"Number of Days: {days_output}")
    else:
        st.warning(f"No recommendations found for {destination_input}.")
```