NOVA IMS Information Management School

MGI

Master Degree Program in

Information Management

Chatbot for Lisbon Tourism

A Proof of Concept Using LLMs and Retrieval-Augmented Generation to Assist Travelers in Choosing Airbnb Accommodations

Vasco Lhansol Souto Massapina

Project Work

presented as partial requirement for obtaining a Master's Degree in Information Management

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

A Chatbot for Lisbon Tourism: A Proof of Concept Using LLMs and Retrieval-Augmented Generation to Assist Travelers in Choosing Airbnb Accommodations

A Proof of Concept Using LLMs and Retrieval-Augmented Generation to Assist on choosing Airbnb

by

Vasco Lhansol Souto Massapina

Project Work presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Business Intelligence and Knowledge Management.

Supervised by

Fernando José Ferreira Lucas Bação, PhD, NOVA Information Management School

July, 2025

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism, any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

[Lisbon, 4th of July]

Vasco Lhansol Souto Massapina

DEDICATION

To those who believed in me when I did not believe in myself, to those who pushed me forward when I felt like giving up and to those who stood by me after all this time.

ACKNOWLEDGEMENTS

I would like to thank my final supervisor, Professor Fernando Bação, who guided me and helped me finish my thesis. I would like to thank my girlfriend Joana, who kept encouraging me and making sure I kept moving forward. I would also like to thank João Sanches, who helped me start my thesis and gave me ideas that would prove essential in my development. I would also like to thank everyone who took the time to test my chatbot and return their invaluable feedback.

ABSTRACT

This work focuses on exploring chatbot technologies, proceed by developing, evaluating and deploying a prototype solution of a chatbot to help tourists in Lisbon, with the initial focus on helping tourists choosing an Airbnb to stay in given a set number of properties provided by our chosen dataset. The two main technologies we will investigate to power our chatbot with a LLM are finetuning and Retrieval Augmented Generation (RAG) in which we explore their uses and practical scenarios, but the version we developed will be using RAG with the help of LangFlow and Vector databases to deploy the chatbot. The chatbot was tested through a structured user survey, and feedback was collected to evaluate its usability, accuracy, and overall utility. Results indicate that users appreciated the ability to ask natural language questions and receive context-aware answers, although limitations in understanding complex or multi-step prompts were noted. This work contributes to the continuous growth of the AI and Chatbots world, and studies its potential uses in Tourism, demonstrating a practical case scenario of the use of RAG to implement a chatbot, while also providing insight into differences and benefits of using RAG or Fine-tuning in each scenario for LLM deployment in other works.

KEYWORDS

Chatbot; Tourism; Lisbon; Retrieval Augmented Generation; LangFlow; Finetuning; Large Language Models; Proof of Concept

Sustainable Development Goals (SDG):

TABLE OF CONTENTS

Statement of Integrity 1
Dedication 2
Acknowledgements 3
Abstract 4
List of Figures 7
List of Tables 8
List of Abbreviations and Acronyms 9
1 Introduction 10
2 Technical aspects of chatbots 11
2.1 History of chatbots 11
2.2 Why Use Chatbots 11
2.3 Essential Concepts 12
2.4 Classification of Chatbots 13
2.5 Adapting a chatbot to our needs – Finetuning, RAG and Vector Databases 14
2.6 Chatbots in Business and Tourism - Benefits and Disadvantages 17
3 State of art 20
3.1 Tourism in Vietnam Chatbot. by Nguyen et Al. 20
3.2 Smart Guidance, by Alotaibi et Al. 20
4 Methodology 21
4.1 Our goal 21
4.2 Available resources 22
4.3 Understanding our data 25
4.4 Exploring and preparing our data 26
4.5 Modelling 38
4.6 Evaluating our chatbot 40
5 Results and Discussion 42
5.1 Ease of Use 42
5.2 Response Quality 43
5.3 Interpretability 45
5.4 Context Awareness 45
5.5 Improvements 46
5.6 Overall User Perception and Satisfaction 47

6 Conclusions and Future Research 51
7 Limitations and future projects 52
8 Bibliographical References 53
9 Appendix A 58

LIST OF FIGURES

Figure 1 - Comparison between different models' performance. Taken from https://lmsys.org/blog/2024-05-08-llama3/?utm_source=chatgpt.com 23
Figure 2 - First 5 rows if Lisbon_Weekends.csv 27
Figure 3 - First 5 rows of the Lisbon_weekends.csv (Continuation) 27
Figure 4 - First 5 rows of the Lisbon_weekdays.csv 27
Figure 5 – First 5 rows of the Lisbon_weekdays.csv (Continuation) 27
Figure 6 - Variable attributes of each csvs (Left - Weekend, Right - Weekday) 28
Figure 7 - Number of distinct value on each csvs (Left - Weekend, Right - Weekday) 28
Figure 8 - Room Type Distribution for duplicate entries 30
Figure 9 - Percentage of Variables that match 31
Figure 10 - Heatmap displaying combination of room_private and room_shared depending on the room_type, during weekdays. 33
Figure 11 - Heatmap displaying combination of room_private and room_shared depending on the room_type, during the weekends. 33
Figure 12 - Room Type Distribution for Weekdays and Weekends 34
Figure 13 - Distribution of indexes for both weekends and weekdays 35
Figure 14 - Distribution of normalized indexes for both weekends and weekdays 35
Figure 15 - New distribution of listings for weekends 37
Figure 16 - New distribution of listings for weekdays 37
Figure 17 - RAG Diagram, Taken from https://netraneupane.medium.com/retrieval-augmented-generation-rag-26c924ad8181 39
Figure 18 - Sample of Chatbot UI 40
Figure 19 - Difficulty of Interaction by Device Type 42
Figure 20 - Chatbot Understanding Accuraccy 43
Figure 21 - User Satisfaction with Chatbot Answers 44
Figure 22 - User's willingness to use the chatbot again 48
Figure 23 - User Satisfaction Score 49
Figure 24 - User Recommendation Chance 49

LIST OF TABLES

Table 1 - Scale for new categorical variable representing the Attraction Index 36
Table 2 - Scale for new categorical variable representing the Restaurant Index 37

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intellience
AGI	Artificial General Intelligence
AIML	Artificial Intelligence Markup Language
API	Application Programming Interface
CSV	Comma Separated Values
GPU	Graphical Processing Unit
LLM	Large Language Model
LSA	Latent Semantic Analysis
ML	Machine Learning
NLP	Natural Language Processing
NLU	Natural Language Understanding
PEFT	Parameter Efficient Finetuning
RAG	Retrieval Augmented Generation
PoC	Proof of Concept

1 INTRODUCTION

Within the past couple of years, the rapid spike in interest on artificial intelligence has been the driving force behind the adoption of the new trending tech by various industries, particularly the utilization of chatbots. Models such as Chatgpt, an AI Chatbot developed from OpenAI's GPT large-language models, that is designed to assist with everyday jobs, has become broadly famous for its great ability to converse with people on various subjects (Rather, 2024). They excel in most general tasks, but this property of theirs often means that when a task needs more precision and deep knowledge, their performance can be lacking, since they may not have updated information about these specific topics and may not be trained to handle them.

This constrain opens the door to another kind of chatbot, Closed Domain Chatbots designed to be knowledgeable in one specific domain. Unlike LLMs, these kinds of models could be fine-tuned to answer questions with higher precision and consistency for the desired domains and deliver the best experience for users with specialized needs. However, they typically lack flexibility and will likely perform poorly outside their domain or avoid unrelated queries altogether. This trade-off is not necessarily a disadvantage as what we want to do is create an efficient system that is intelligent on the context it was trained for.

This project's goal is to create a proof of concept chatbot and develop it to be used by tourists coming to Lisbon. Lisbon is a is one of the most visited cities in European with a growing need for easily available information for people looking to make trip in it. What's hoped for here is demonstrating how an assistant built for a specific sector, trained on carefully selected information, can benefit individuals by providing support on deciding on how to make their trip. Chatbots are very impressive for communication and knowing a little bit of everything, but a LLM's effectiveness for tasks specific to tourism is normally impaired because of incomplete, outdated information, or sometimes both, especially when local details and accuracy are most critical (Carvalho & Ivanov, 2022). Though the project we will develop here is limited to a proof of concept, its ultimate goal would be to create a digital assistant capable of aiding tourists planning an entire trip from transport and lodging to local attractions and travel bureaucracies.

To create our chatbot, we will investigate technologies like retrieval-augmented generation (RAG), structured prompting and finetuning to build our chatbot and utilize low-code platforms where we can. Rather than utilizing typical benchmarks comparisons versus LLMs like ChatGPT models, the system will be evaluated with realistic end-user interactions and feedback collection. This approach provides us with more realistic insight into the usability of the chatbot, its performance, and where we should improve.

2 TECHNICAL ASPECTS OF CHATBOTS

2.1 HISTORY OF CHATBOTS

Chatbots are a technology that is currently very sought after, but chatbots weren't invented in the last decade. One of the oldest ancestors of chatbots is "ELIZA", created in the 1966 by Joseph Weizenbaum, with the objective to mimic a psychotherapist. It had the knowledge of a psychotherapist and was taught many techniques to avoid reaching dead ends in a conversation (Calvaresi et al., 2021).

There were many other chatbots developed in the last century that all had a meaningful impact in the progress of chatbot technology, such as PARRY, which was developed in 1972 with the objective of simulating artificial paranoia (Artificial Paranoia' Kenneth Mark Colby), and a more important chatbot which was A.L.I.C.E., a chatbot which many considered a successor of ELIZA (The Anatomy of A.L.I.C.E. Richard S. Wallace), created by Richard Wallace, which won the Loebner Prize, an annual competition of different artificial "intelligences" . The most recent chatbots that we must interact with nowadays include the assistant present on our phones, like Google Assistant, Cortana and Siri. One of the greatest chatbots of this decade is ChatGPT, which we will discuss a lot in our project (Adamopoulou & Moussiades, 2020).

2.2 WHY USE CHATBOTS

Understanding why people consider using chatbots is key to creating and developing systems that meet the requirements of the situation at hand. Users are often motivated by simple goals, such as efficiency, convenience, and help completing a task (Brandtzaeg & Følstad, 2017).

Research by Brandtzaeg and Følstad points out that the one of the most common reasons users turn to chatbots is the need for immediate answers, availability outside work hours, and the possible simplicity of the UI associated to the chatbot. These motivations are often reinforced by the chatbot's perceived usefulness and facility of usage, which are closely linked to customer satisfaction and willingness to utilize the bot again. Besides their reliability and efficiency, chatbots are also commonly used due to their ability to provide entertainment and being the newest technology but are also viewed by users as friends or casual conversation partners, most frequently by younger groups. The article also concludes that expectations play a significant role in one's opinion about the chatbot, users who go in already expecting it to perform well are more likely to report positive experiences. From these points we can say that chatbots' appeal extends beyond automation; people also use them as easily accessible and reliable tools that can help accomplish practical needs but also as entertainment (Brandtzaeg & Følstad, 2017).

We can take a historical look at the development of chatbots and their associated technologies in the 2020 article “An Overview of Chatbot” by Adamopoulou and Moussiades. Their work synthesizes insights from numerous scientific publications to clarify the evolution of chatbot systems, followed by a detailed explanation of relevant terminologies in this area. One of the key conclusions drawn from their review is that the widespread adoption of chatbots is derived from a combination of benefits, both for users and for developers. From the user’s side, chatbots offer immediate access to information without requiring the installation of specific platforms and/or applications, they can support multiple simultaneous conversations, and their convenience of interacting through a familiar interface, like that of a messaging app. On the developer side, the ability to reuse prompt-engineering logic across multiple systems, the prebuilt UI for user interaction, and the absence of version differences considerably facilitate the work of the developer. The authors emphasize that productivity continues to be the main reason as to why a chatbots are used, however, entertainment, having someone to talk to, and the recency of the technology also play important roles in attracting more people to this technology (Adamopoulou & Moussiades, 2020).

2.3 Essential Concepts

Chatbots are mostly built using machine learning, a subfield of artificial intelligence that allows systems to learn from data and improve performance based on the data they were given to learn from (Akma et al., 2018). Although a wide range of technologies can support the development of chatbots, we will focus on the most relevant and foundational components.

Natural Language Processing, or NLP for short, is a branch of artificial intelligence focusing on making bots better able to comprehend, analyse, interpret and respond to user queries with spoken and written natural human language (Akma et al., 2018). Most modern NLP methods often rely on different machine learning techniques to enhance their ability to process a multitude of different linguistic patterns. Another branch of Machine Learning, often paired with NLP, is Natural Language Understanding (NLU), which has the important role of making chatbots able to interpret the objective behind one user’s input. As mentioned in Semantic Vector Learning for Natural Language Understanding, NLU is essential for interpreting the user's intention and the entities present on its queries (Akma et al., 2018).

The term Intent describes the objective or purpose of a user's prompt, or what the user wants the chatbot to do. Entities, on the other hand, stand for pieces of information that are retrived from the user’s input and help placing in the purpose of the context (Akma et al., 2018). A Comparison and Critique of Natural Language Understanding Tools highlights that entities enable the chatbot to identify and classify important information like names, locations, and dates. One example of these two aspects of NLU is, in the question “What is the temperature in Portugal?”, the intent is to request weather information, while “Portugal” is considered an entity within the “Country” category.

Combining Latent Semantic Analysis (LSA) with Artificial Intelligence Markup Language (AIML) is another technological foundation on which some chatbots have been developed. AIML is a programming language that works by following rules created by the developers of the chatbots to make “Blueprints” or presets of possible inputs to handle common and frequently requested prompts (Thomas, 2016). Most AIML templates work similarly, they create response guidelines for common questions like "How are you?" or "What's up?" Essentially, these templates help chatbots know what to say whenever they recognize certain patterns or phrases (Thomas, 2016).

AIML has significant drawbacks even if it is straightforward and efficient for everyday usage. The amount of effort involved in writing and coding every potential user query is one of the biggest drawbacks. As the complexity of the chatbot's domain increases, developers must increasingly describe more and more possible input patterns, and this can quickly become a tedious task and unsustainable on a long-term basis, as there can be a large number of unpredictable inputs. To solve this problem, Latent Semantic Analysis (LSA) is often paired with AIML systems to improve them. LSA is a statistical method that uses vectors to represent words in a high-dimensional space to determine their semantic links. LSA can be used to search a database for frequently asked questions or knowledge base items and identify with their semantically closest match when a user input does not exactly fit the predetermined AIML patterns. This, in turn, helps reduce the time spent creating all patterns and turn this task into a more sustainable one, improving a chatbot’s flexibility and usefulness, by making it able to gather more appropriate responses to more situations, even if they don’t exactly match provided AIML patterns (Thomas, 2016).

2.4 Classification of Chatbots

Chatbots can be classified in different ways depending on their structure, purpose, and interaction model. The four most predominant ways of classifying a chabot refer to their knowledge domain, service type, their goal, and Input/Output generation method. This section will focus on the most relevant categories for this project (Ketakee Nimavat, & Tushar Champaneria. (2017)).

First, the Knowledge domain, is responsible for classifying a chatbot considering the information it is capable of handling. Two main types are distinguished in this domain, Open Domain Chatbots and Closed Domain chatbots. Open Domain Chatbots are chatbots designed for general usage, unrestricted to a specific topic. They mostly rely on Large Language Models (LLM) and utilize substantially large training datasets to generate coherent and natural sounding, or human sounding, responses. An example of these kinds of systems is the widely known ChatGPT. In the other end of the spectrum, Closed Domain Chatbots are chatbots designed to be used for a specific topic, such as IT-HelpDesks or Hotel Bookings. These chatbots are optimized to answer questions specific to their topic and will often fail or attempt to divert questions unrelated to said topic. This type of chatbot is the one we are aiming to demonstrate on this project (Ketakee Nimavat, & Tushar Champaneria. (2017)).

Secondly, there is the type of service provided, which classifies chatbots based on how they interact with the user, being split into three categories, Intrapersonal, Interpersonal and Inter-agent chatbots. Intrapersonal chatbots directly interact with the user, and with the user's environment. They can do things like Set up alarms, schedule events, or set up reminders. They often attempt to simulate human conversations and gather information about the user to build a user-profile for future reference. Common examples are Google's Google Assistant and Apple's Siri. Chatbots that function within the social or personal distance range are known as interpersonal chatbots. These consist of systems made to handle frequently asked questions or to help with services like booking reservations for restaurants or flights. These bots are designed to be useful agents that retrieve and provide customers with information and are not designed to attempt to be companions or form a relationship with the costumer. The chatbots might take on a friendly behaviour or save certain user data for later, but they are not expected to. This type of service is the one our chatbot will provide. Lastly, Inter-agent chatbots are less common chatbots that are used by other bots as a mediator of information. They coordinate multiple services and systems to help coordinate tasks that require multitasking and are normally not accessible to a user (Ketakee Nimavat, & Tushar Champaneria. (2017)).

Thirdly, the goal of a chatbot is a classification based on a chatbots' purpose, which in turn alters how they are created and designed, and how users get to interact with them. These can be classified informative, conversational, or task based chatbots. Informative chatbots' purpose is to focus on delivering the information requested to it. They are often used to support customer service and often rely on a frequently asked questions database to answer costumer questions. For now, our chatbot will have this kind of goal, but in a future version, it could be rebuilt as a Task based chatbot. These chatbots are often made for specific tasks, like helping customers navigate an online store or making a travel reservation. They are intelligent chatbots that can make the necessary questions to the user to help better achieve their task. Conversational chatbots are made to engage in conversations and interactions with a user. Their goal is simply to respond correctly to a prompt they are given with natural dialogue and to attempt to continue a conversation with the user (Ketakee Nimavat, & Tushar Champaneria. (2017)).

Chatbots can also be classified by how one is able to interact with them. Chatbots can be text based, voice based or even have multiple input methods. These choices affect the system's accessibility (Ketakee Nimavat, & Tushar Champaneria. (2017)).

2.5 Adapting a chatbot to our needs – Finetuning, RAG and Vector Databases

There are 2 main technologies that we are going to look through to decide how we are going to create our chatbot, finetuning and Retrieval Augmented Generation (RAG).

Natural language text generation has proven to be one of the strong points of large language models (LLMs), such as GPT-3 (175 billion parameters) and GPT-4 (reportedly over 1 trillion parameters), however, LLMs operating at this scale come with elevated costs, both to

maintain and further develop. The resources associated to running these LLMs demand exponencial GPU power, huge memory utilization, and extended training cycles to fully fine-tune such models for particular services or domains. For example, it may take up to 40 million GPU hours to train a 405B parameter model like LLaMA-3.1 (Tay et al., 2024). Additionally, for most organizations it is not doable to retrain some LLMs from scratch due to financial limitations or being unable to access the dataset originally used to train the dataset. For many real-world applications, particularly in settings with limited resources, such as ours, this renders conventional fine-tuning techniques impracticable (Buehler et al., 2025).

In order to make finetuning more practical for more users, Parameter-Efficient Fine-Tuning (PEFT) was created as a way of adapting LLMs to specific tasks or domains. PEFT works by instead of retraining the entirety of the LLM, only aiming to retrain a very small subset of its parameters, usually 2-3%, leaving the rest of the model as it originally was. Since we are training a much smaller number of parameters, the amount of time and resources necessary will decrease accordingly (Tay et al., 2024). Besides being more efficient with its resources and time available, it also consequently consumes less energy and lowers carbon emissions (Buehler et al., 2025), a problem that is starting to rise with big LLM companies like OpenAI.

There are different ways of approaching finetuning, Continued Pretraining (CPT) expands an LLM's knowledge and proficiency on a specific topic by training it with raw informative text files. Using labelled datasets, Supervised Fine-Tuning (SFT) teaches the LLM how to handle tasks a user can request, such as summarizing content or answering questions. More recent developments, such as Direct Preference Optimization (DPO) and Odds Ratio Preference Optimization (ORPO), train models using input to assist match outputs to user expectations. (Tay et al, 2024).

PEFT can also be split into 6 different types. Additive PEFT adds small prompts to the training data, to help guide the LLM into specific responses, Reparametrized PEFT modifies the weight of some parts of the data to give it more or less importance, using things like Low-Rank Adaptation (LoRA). Selective PEFT applies to when we target only a smaller number of parameters of the LLM, hybrid PEFT combines different PEFT types to cover more aspect and in turn make the LLM more flexible, Quantization PEFT uses layers to improve speed and memory usage, and Multi-task PEFT supports parameter sharing across tasks to provide the LLMs with multi-task learning (Tay et al., 2024)..

Another direction fine-tuning is currently heading towards is model merging, where separately fine-tuned models are merged into a new model with enhanced or emergent capabilities, supposedly retaining the best of each LLM, but not exclusively, as this approach is not simply additive, it can result in features not present in either source model alone. One method of merging frequently discussed is Spherical Linear Interpolation (SLERP), which attempts to merge two models evenly, helping preserve how the original models were structured (Buehler et al., 2025). Compared to smaller models, larger models such as LLaMA-2-7B or Mistral-8B show more benefits from merging (Tay et al., 2024).

The quality of the training data will always have the most significant impact on how well any fine-tuning approach works. While clean, organized and structured data, like Q&A examples or data tables indubitably improve results of finetuning efforts, unprocessed data will always produce worse results. Having a good database to start with is essential for any finetuned LLMs to work properly, specially when talking about LLMs that are going to be specialized in specific domains, where finetuned models are supposed to analyze domain specific queries like scholarly articles, court records, or user interactions unique to a given business (Buehler et al., 2025).

In recent years, Retrieval-Augmented Generation (RAG) has gained popularity due to boosting the performance and reliability of LLM's, most predominantly in applications that depend on LLM's to provide lots of factual information on specific topics (Jeong, 2023)(Lewis et al., 2020). One of LLM's like ChatGPT's biggest weakness is a phenomenon known as hallucination. Hallucination is a phenomenon best describe as when the Chatbot is requested information that it possesses scarcely, resulting in a response that despite sounding accurate and coherent, is made of factually incorrect information due to not having the necessary documents and information (Jeong, 2023; Lewis et al., 2020). RAG overcomes this by giving the LLM's the knowledge they require through documents and data sources provided to it, preventing hallucination on the given topics and boosting its performance (Jeonb, 2023).

Vector databases are a more recent new type of databases that, instead of storing information in text or number format, they are built to store information as embeddings, which are numerical representations of the data, be the original data text or numbers. These embeddings are capable of storing the data's original meaning. These vector databases are a key component of RAG, as by capturing the context of the data, it allows the RAG system to find data that is semantically similar, even if it doesn't use the same exact words (DataStax, 2025)

When a user asks a question in a chatbot powered by RAG, their input is turned into a vector and compared to a collection of stored vectors representing pieces of documents or data. The database then returns the most similar matches. The large language model receives the returned chunks, which increases the response's precision and reliability. This procedure helps prevent the hallucinatory effects that a chatbot may have and enables the model to give more accurate responses without having to keep everything in its own memory (DataStax, 2025). Because they are optimized for speed and large-scale searches, vector databases are widely used in modern AI systems to support features like personalized recommendations, semantic search, and conversational agents.

LLMs often face constraints like limited adaptability to new data or outdated information beyond their training, for example, the GPT 3.5 model lacking data beyond September 2021 (Jeong, 2023).

One of LLMs most frequent difficulties is their data. Their data is often outdated and when requested information outside their training they can struggle. One case is the famous GPT 3.5, whose data only reaches September 2021 (Jeong, 2023).

RAG can a dress this frequent issue by feeding the LLM with the required data, updating its knowledge base without necessitating expensive retraining of the entire LLM (Jeong, 2023; Lewis et Al., 2020). This is further reinforced by RAG's ability to simply swap in and out the necessary files by simply updating the connected database, meaning new information is easily added and old information that was added this way can equally easily be removed from the vector database, making RAG an extremely practical solution for frequently changing domains (Lewis et Al., 2020)

RAG enhances interpretability by including sources for its answers, making the retrieved external knowledge easy to access and navigate whenever needed (Lewis et al., 2020). This combination of retrieval and generation allows chatbots and AI systems to deliver more accurate and contextually relevant responses without the need for time-consuming retraining, making RAG a more practical approach then finetuning for enterprise-level use (Jeong, 2023; Lewis et al., 2020). Additionally, chatbots that use RAG tend to perform significantly better when it comes to question-answering tasks (Lewis et al., 2020).

2.6 Chatbots in Business and Tourism - Benefits and Disadvantages

While there are clear obstacles as well as benefits both for the operational logistics and costumer usage, the use of artificial intelligence and chatbots in tourism and hospitality services has completely shifted how companies interact with their clientele. Chatbots, even more those made with LLM's, have gained popularity as almost essential tools for enabling multi-language, scalable and live assistants to interact with clients and help them on their journey as a digital tourist guide as smart service platform (Carvalho & Ivanov, 2023; Rather, 2024).

One of the most evident advantages of using chatbots is the possibility of them working all the time, making sure they are available 24/7, despite the user's time zone or staff limitations of the deployer company (Carvalho & Ivanov, 2023; Jiang et Al. 2022; Rather, 2024). This uninterrupted availability to fast and efficient support is extremely valuable in tourism, where travelers might need help at anytime for informations like directions, travel information, personalised recommendations, helping define an itinerary or even making bookings (Carvalho & Ivanov, 2023; Rather, 2024). Together with chatbot's capability for infinite grow, which allows companies to handle high quantities of data and simultaneously interact with customers efficiently, the use of chatbots allows the increase of capabilities of a company's service and also frees human resources to handle other tasks without disrupting speed and quality of responses to a client, which is essential, especially during peak tourism seasons (Carvalho & Ivanov, 2023; Rather, 2024).

Chatbots also allow the developers to make “blueprints” of how they should handle and respond to customers by recommending them specific solutions to their situations. This customization allows the developers to better target their customer’s needs, but I can also increase brand “loyalty” and the user’s intent to return to a service (Rather, 2024). AI agents can also help boost efficiency by being responsible for repetitive tasks, like answering commonly asked questions (Q&A), managing reservations or providing extra details about certain itineraries or specific places, which in turn allows human resources to be freed, and consequently allocated to tasks that require more complex reasoning or simply higher-stake responsibilities (Jiang et Al., 2022). Their extra ability of being able to collect data from interaction with users, which in turn allows developers to analyse it, also helps to further develop the chatbot and to better understand what customers expect and need out of their business (Carvalho & Ivanov, 2023; Rather, 2024).

Despite these advantages, chatbots still process some deployment constraints in the tourism industry (Rather, 2024). One frequent setback is the absence of the human touch when interacting with chatbots, even thought some LLM’s have become incredibly good at simulating dialogue with a real human (Carvalho & Ivanov, 2023). Chatbots often fail to provide emotional nuance or empathic responses to users, which are often necessary to make users more comfortable and, in turn, trust the chatbot more on its decisions (Rather, 2024). Some tourists that appreciate a human’s touch might be repelled from a company if the chatbot fails its task and might leave them frustrated (Jiang et Al, 2022).

Another limitation is that LLM’s often aren’t train to handle complex or ambiguous questions. Questions that might have multiple steps to a reasoning, that might require some cultural knowledge, or even questions that have some emotional background may not be properly interpreted by the chatbot and its answers may be inadequate or inaccurate, providing more generic responses (Carvalho & Ivanov, 2023). These constraints might become problematic when encountered by users, especially in high-stakes scenarios, where emergency assistance might be required (Jiang et Al., 2022)

Besides the conversational aspect of chatbots, there are also some technological constraints. Many chatbots and LLM’s struggle with intent that is meant to be sarcastic, or with question that have regional dialect variants a Chatbot is not accustomed to. The performance of a chatbot is heavily reliant on the quality, accuracy and recency of the data that was used to train the chatbot. Reliance on outdated/incomplete data is always going to severely impact the performance of a chatbot (Rather, 2024).

Additionally, there is a substantial initial investment required that is associated with the deployment and maintenance of chatbot systems. Even if they provide long-term benefits, smaller businesses and companies may find the required initial investment, be it on actual server hardware or a cloud platform to host the chatbot on, too high (Carvalho and Ivanov, 2023). There are also a few ethical difficulties. Chatbots actively collect data shared by the

user, and this data can be very sensitive. Without proper data governance measures, this data can turn into safety breaches, which can raise ethical and legal issues (Jiang et Al., 2022).

Cultural and linguistic diversity poses another challenge. Although many chatbots support multiple languages, their proficiency may vary, and they often fail to account for regional variations or cultural subtleties in communication. This is particularly relevant in international tourism, where customer expectations differ widely (Carvalho & Ivanov, 2023; Gursoy et al., 2023). Lastly, over-reliance on automation can lead to a reduction in human staff, potentially eroding the personalized, high-touch service that defines many hospitality experiences (Gursoy et al., 2023).

Chatbots also have some language and cultural barriers, mostly due to the data they are trained on. Many chatbots support multiple language inputs, but their proficiency in each language is never the same, and they can fail to respond to the same question in a different language, or even in the same language, but with regional variants. This is particularly troublesome in international scenarios, where customer expectations and inputs differ widely (Carvalho & Ivanov, 2023; Rather, 2024).

It is important to note that the over reliance on AI can lead to reduction in human staff, which often erodes a company's personal identity, and is often seen as "bad" by the public in today's standards, often detering someone's experience (Rather, 2024).

3 STATE OF ART

3.1 Tourism in Vietnam Chatbot. by Nguyen et Al.

Nguyen et Al. (2023) created a Chatbot aimed for tourist recommendations in Vietnam, in hopes of supporting the local tourism. The paper suggests using machine learning techniques to build a Chatbot using LLM's like BERT and RoBERTa to handle questions, identify a user's intent and train conversation scenarios.

The dataset used to test this chatbot had 7319 samples, which after finetuned into the used LLM, resulted in a 90% accuracy and a 90.1% F1-Score on the test set after 200 epochs utilizing BERT, which outperformed other models in this test.
Using the Flutter framework, the chatbot was developed as an Android mobile application. The application offers recommendations to users and has a user interface (UI) built with the intent of simplifying usage. While administrators can control content and train the chatbot model to increase its knowledge on touristic attractions in Vietnam, and users can read articles, check hotel rates, obtain information about tourist locations, and ask location-related inquiries. The research highlights how easy it is to use AI in the tourism field, promoting tourism development, enhancing the experience for international tourists, and also highlights potential advantages of using AI in tourism.

3.2 Smart Guidance, by Alotaibi et Al.

Another example is the 2020 study by Alotaibi et al. that presented "Smart Guidance", a text chatbot that was suggested and created as a mobile application for the Jeddah, Saudi Arabia, tourism industry. Being the second biggest city in Saudi Arabia, Jeddah has multiple activities that visitors can do to explore.This Chatbot was made with the final objective of providing virtual assistance, chatting with users in natural language, at the time exclusively Arabic, and acting as a single point for all information a tourist might want. It is available 24/7, with the planned addition of english support and voice chat in the future.

"Smart Guidance" provides real-time responses to user queries. It can answer questions related to desired destinations, recommend offers, and highlight real-time events to satisfy tourist needs and save time and effort. Users can inquire about restaurants, malls, hotels, coffee shops, and request recommendations for places near their current location. The system can display details of places and their locations via a map. Additionally, it can provide current weather states by calling external APIs. The chatbot uses Natural Language Processing (NLP) and Machine Learning (ML), with Rasa.ai serving as its open-source framework for language understanding and dialogue management. Its Natural Language Understanding (NLU) model is trained to comprehend messages written in Arabic, performing pre-processing for Arabic text such as entity recognition, extraction, and tokenization. Users found the interaction effective, and the responses were provided immediately. Performance tests also concluded that the bot was relatively quick, taking on average 5 seconds to respond to most questions.

4 METHODOLOGY

In this phase, our objective is to lay out the plan of our project, and to determine our needs and objectives, from a technical perspective, develop our project and attain a result of which we can take conclusions. We will be doing this by adopting a methodology commonly known as the CRISP-DM, which has been for years the most well adopted way of working with projects that handle data, and adding our own twist to it.

4.1 OUR GOAL

As we have discussed, the objective of our project is to create a Proof of Concept (PoC) of a chatbot that could be used by the *Câmara Municipal de Lisboa* to support tourists with the exploration of Lisbon. Our PoC will focus on the airbnbs of Lisbon, making our chatbot able to provide tourists with the best possible accommodations given their needs. We chose to focus on this particular attribute since it often is the most important aspect of a trip, as a well rested tourist is a happy tourist, and those spend more.

When planning a trip, one can struggle to find, as it is rarely ever in one place and requires people to invest a lot of time into planning their trips. Tourists increasingly rely on technology to support their searches, and when the information one needs is hard to access and navigate, it can quickly lead to frustration. In some cases, there's too little information and the tourists can become demotivated by the lack of access to information and the feeling of uncertainty. In others, the opposite can happen.

There is an overwhelming abundance of information on specific subjects, but often different sources contradict each other, which can make things like finding the Airbnb closest to the city center unnecessarily confusing and frustrating. There's also the linguistical barrier, where the information someone needs may be in a language they aren't familiar with. In such cases, tourists might rely on poor translations, which can lead to misinformation.

Aware of these challenges, the aim of our project was to ease this process, by creating an official source of information, that can be active 24 hours a day and 7 days a week, user-friendly, and capable of keeping track of all of the information seeker's requests on one place with all the benefits of a chatbot that we previously mentioned. This is the reasoning behind our project, and while the current scope of our project doesn't include every single type of information one tourist could need, it is our first step into a complete chatbot and, completed successfully, it could be expanded with the assistance of a reliable accurate up-to-date information source, like the *Câmara Municipal de Lisboa*, making it possible to create a complete database with all information a tourist could need

Our project will be developed with a few limitations, as we will not be using an official source of information yet.

We will consider our project successful if we can develop said chatbot, correctly responding to a high percentage of our question about the database it is provide with, based on a test set we will create, returning context-aware answers based on the user's requests, and if the chatbot is easy to use, access and maintain.

To achieve our goal, we have at our disposal some tools that we have previously discussed, and will need to find others to complete our project.

4.2 AVAILABLE RESOURCES

To develop our chatbot, we had two main tools at our disposal: fine-tuning and Retrieval-Augmented Generation (RAG). We have previously discussed the strengths and limitations of each of these approaches, and based on our findings, we decided to follow the RAG architecture and to discard fine-tuning our model, due to the intensive resources needed to be able to properly train our chatbot, consequence of this project being developed entirely on a local machine, including hosting the chatbot itself. To implement our RAG architecture, we will use LangFlow.

LangFlow is a visual interface built on top of LangChain, which allows users to design and deploy LLM-based workflows using a low-code interface. It supports components such as document loaders, retrievers, prompts, and large language models, making it easier for anyone to start developing their own RAG pipelines. LangFlow's modular design allows us to explore different ways to load our data into the model and decide how to proceed.

Having decided our approach, we also needed to decided which models we were going to use. We had 2 main LLM providers that we decided to choose from:

Meta's LLAMA models

LLAMA, which stands for Large Language Model Meta AI, is one of Meta's most recent attempts at creating an LLM. There have been multiple public versions released, ranging from as low as 1B parameters to 70B parameters, and these models have been trained with over 15 trillion tokens with the goal of improving reasoning, instructions following and multilingual prompts (Meta AI, 2024). Unlike proprietary systems like those from openAI, Meta's llama models are available for free under an easy access permissive license, making them attractive to small scales projects like academic researchs and deployment in local or private environments. It's a family of models designed to compete with openAI's GPT models.

Despite strong performance, llama models require significant setup and infrastructure and often trail behind other flagship models like OpenAI's ChatGPT.

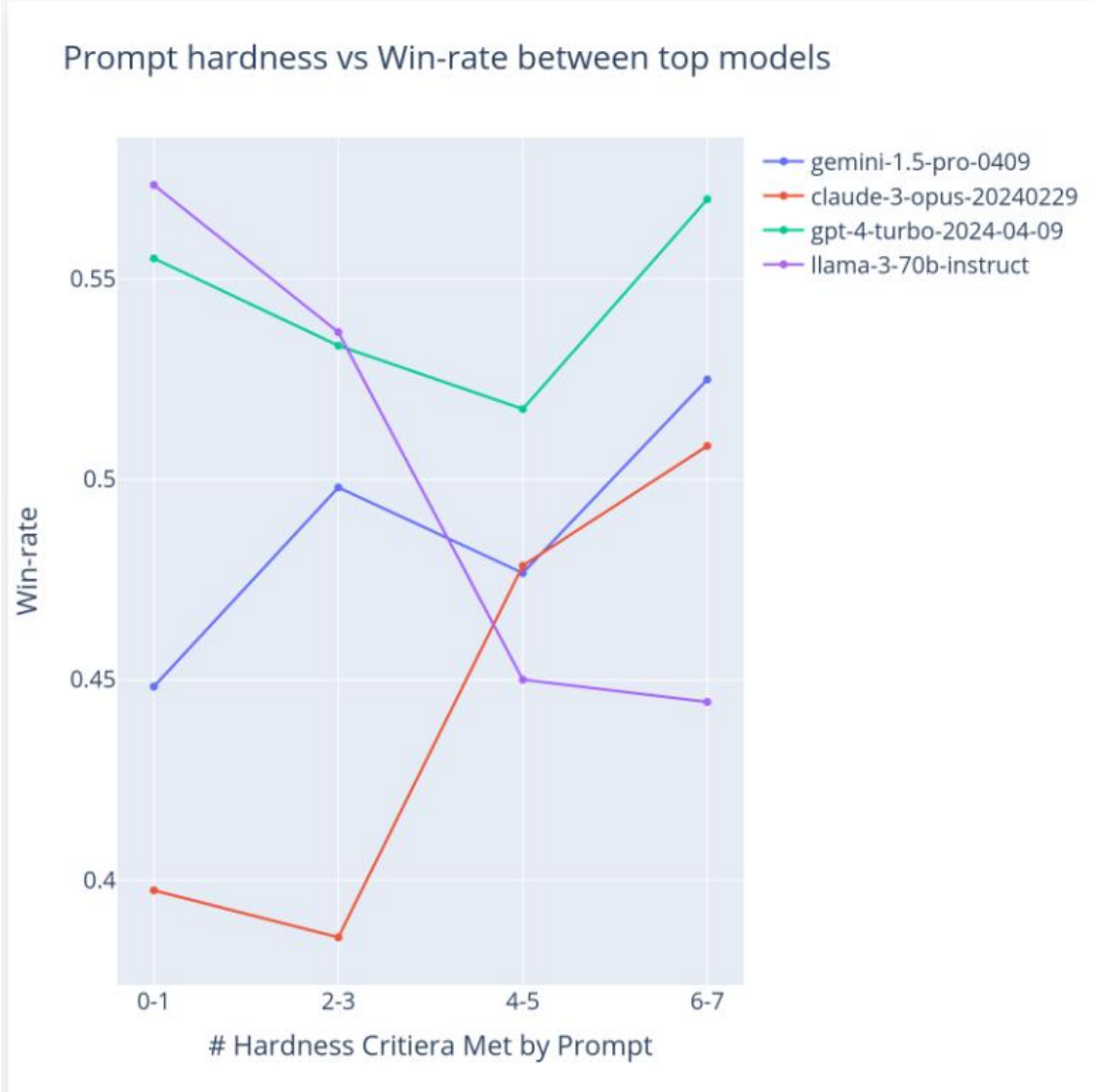

Figure 1 - Comparison between different models' performance. Taken from https://lmsys.org/blog/2024-05-08-llama3/?utm_source=chatgpt.com

They have limited, almost inexistent built-in API access, safety layers, and don't process chat memory unless manually added to the infrastructure. Llama's flexibility also means developers have to handle more tedious tasks like tokenisation, RAG integration and prompt tuning (LMSYS, 2024). These types of tasks are also normally more associated with finetuning.

OpenAI's GPT models

The famous ChatGPT models were created by OpenAI, a company founded on 2015, initially established as a non-profit organization, whose main objective was to create Artificial General Intelligence (AGI) capable of outperforming humans in many tasks with the objective of benefiting all humankind (Palihapitiya, 2023).

OpenAI has made lots of advancements on the field of AI with models like ChatGPT 3 and 4, and image generation models like DALL-E and SORA, with these technologies being integrated in multiple business (The Wall Street Journal, 2025).

The ChatGPT LLMs are a closed-source cloud based LLMs, which means users rely on OpenAI's infrastructure and terms of service. It is a paid service, with a free tier allowing limited usage of the ChatGPT 3.5, while the paid tier allows full access to public models shared by OpenAI. These models can be accessed through the ChatGPT website our with the use of APIs. Despite easy access to APIs, due to being close sourced, it does not allow local deployment or full costumization, but it is easier to use, often being ready to implement in different cloud based environments with minimal tweaks needed, excluding the prompting rules we setup based on the context of the RAG model we are building.

ChatGPT models are often seen as the most reliable and easiest to use, and this was one of the key factors that made us choose to use these models over other reliable options.

OpenAI provides different LLMs with different performances and costs, as a results of both its advancements and different objectives. The weakest of its models is GPT 3.5 turbo, which is what the free tier of ChatGPT has access to. It is well suited for simple tasks and has very low resource cost, with fast response times (LMSYS, 2023).

GPT-4, released in 2023, offers stronger performance and scores significantly better results across different benchmarks (OpenAI, 2023), but its slower response times and expensive costs made it less practical for different applications.

GPT-4-Turbo was a more efficient variant of the GPT-4 model, able to retain more context then GPT-4 across longer conversations, making it more reliable for bigger projects (Vellum AI, 2024).

In 2024, GPT-4o ("o" for omni) was released. It had GPT-4 intelligence, but improved on its latency, added multi model input (Text, images and audio), all at a significantly lower cost (OpenAI, 2023, 2024). Because of this, GPT-4o is the most used GPT model. With its released, OpenAI also released the GPT-4omini version, a much smaller but faster variant made specifically for environments where answer speed is crucial such us LangChain or LangFlow agents. While more cost effective, it often underperforms on most benchmarks, mainly failing when reasonings become increasingly complex (OpenAI, 2024).

GPT-4.1 was also released by OpenAI in 2024, which was an upgrade of the previous flagship models GPT-4. It outperformed even the GPT-4 Turbo in terms of speed and reasoning power and, additionally, it featured a very useful feature for us, in the form of an extended context window of up to one million tokens, which let it access a bigger chunk of our database whenever it needed. Furthermore, it featured a GPT 4.1 Mini version, which was a lighter model that had less reasoning power but was faster, less expensive per token, and required less memory during deployments. This made it suitable for deployments with limited

resources like ours (OpenAI, 2024). In the end we decided to choose GPT 4.1 mini, as it offers the best balance of attributes beneficial for our deployment in LangFlow, where lightweight models are critical, while trading off, trading off others we don't particularly need, like the stronger reasoning.

Having decided what resources we were going to use, we now need to find a suitable data source for our project.

4.3 UNDERSTANDING OUR DATA

This project will focus on being a proof of concept that a chatbot with information about Lisbon specifically is possible and a worthwhile investment to better inform tourists before they come to Lisbon.

To prove said concept, we looked for a dataset with information about a specific aspect of Lisbon that would no doubt help tourists plan their trip, which led me to search for datasets in multiple repositories. Eventually, we found a dataset about Airbnb's price calculations shared by hugging face but initially published on Zenodo.org, called “Determinants of Airbnb prices in European cities: A spatial econometrics approach (Supplementary Material)”. This dataset contains information about Airbnb’s through the world, with relevant information, such as the location of the Airbnb’s, and it contains 5.17 thousand entries.

A key aspect of a tourist's trip is to have a proper accommodation, as the quality of rest of a tourist will allow him to explore more actively, and the location of an accommodation is also a key factor on which attractions a tourist may be able to visit or how much dislocation one might have to do to visit all his desired places.

Besides their location in geological coordinates, this dataset also contains information regarding other crucial factors like the price, guest satisfaction and the ratings of the airbnbs, which we will explore further ahead.

This dataset also comes with an already structured format that makes it easier to process and transform into the information we may need from it, making us spend less efforts in preprocessing the data and more into exploring and analyzing it. It also makes it easier for a chatbot to read the data and filter it for specifics attributes.

We also choose this dataset due to its origin, it comes from two very well-known and reliable open sources. Hugging face is well known for its educational strides in the LLM world and Zenodo is a data source backed by the European Commission’s OpenAIRE project, which supports multiple different research projects.

One limitation present in this dataset is it not being up to date and not being regularly updated, but since we are only trying to create an initial version of a chatbot, this will not pose a challenge in the time being.

We will be using this dataset to feed the chatbot and improve its Q&A capabilities, and to provide better recommendations to the tourists.

The dataset we are using comes in the format of multiple csvs, with the information about each city split into 2 csvs, one with information gathered for weekdays and one for weekends. The dataset comes with a short description of each variable, that allows us to more quickly understand what each variable means, and potential uses each one may have. We are only using the csvs relating to the city of Lisbon and, upon a first look, we can see they have a different number of entries, 2859 for weekdays and 2908 for weekend, which makes us conclude that not all entries have information for both weekdays and weekends, we will have to decide what to do with this situation further ahead.

The dataset's variable can be split into 6 categories, Listing location, Accommodation Characteristics, Host information, Pricing, Quality and Satisfaction and Location-Based Metrics.

Listing location includes the city of the Airbnb (city) and its geographic location (latitude, longitude), Accommodation Characteristics includes the type of accommodation (room_type), whether the room is shared or not (room_shared, room_private), the number of bedrooms (bedrooms) and the number of guests it can house (person_capacity).

Host information includes whether the host is a super host or not (host_is_superhost), and how many listings the host has (multi, biz)

In Pricing, we get the total price for a 2 night stay (realSum) and if its weekend or weekday (day_type).

Quality Satisfaction includes how clean the place is (cleanliness_rating) and the satisfaction of the guest (guest_satisfaction_overall).

In Location-based Metrics we have the distance from the city center (dist), the distance from the closest subway (metro_dist), and some information about how good the surrounding area is (attr_index, attr_index_norm, rest_index, rest_index_norm).

Throught our analysis, we will assume both csvs were gathered at the same time, to avoid discrepancies we might encounter ahead.

4.4 EXPLORING AND PREPARING OUR DATA

Now that we better understand our dataset's structure and variables, we can start exploring it and identify preprocessing steps we might need to take, by starting with an overlook of the first few entries

	Unnamed: 0	realSum	room_type	room_shared	room_private	person_capacity	host_is_superhost	multi	biz	cleanliness_rating
0	0	137.664165	Private room	False	True	2.0	True	1	0	10.0
1	1	123.827392	Private room	False	True	2.0	True	1	0	10.0
2	2	193.011257	Private room	False	True	4.0	True	1	0	10.0
3	3	326.219512	Entire home/apt	False	False	6.0	False	1	0	9.0
4	4	174.484053	Private room	False	True	3.0	False	0	1	10.0

Figure 2 - First 5 rows if Lisbon_Weekends.csv

guest_satisfaction_overall	bedrooms	dist	metro_dist	attr_index	attr_index_norm	rest_index	rest_index_norm	lng	lat
98.0	1	4.328041	0.298493	74.230019	2.448349	176.867933	9.940446	-9.14034	38.75137
97.0	1	4.465486	0.293602	72.571059	2.393631	173.909831	9.774192	-9.14092	38.75260
87.0	2	4.475239	0.167851	72.517817	2.391875	177.996433	10.003870	-9.14245	38.75264
93.0	2	0.667018	0.530362	537.049195	17.713637	775.734765	43.598345	-9.13200	38.71300
96.0	1	4.888606	0.466018	67.971330	2.241917	160.215069	9.004511	-9.14053	38.75642

Figure 3 - First 5 rows of the Lisbon_weekends.csv (Continuation)

	Unnamed: 0	realSum	room_type	room_shared	room_private	person_capacity	host_is_superhost	multi	biz	cleanliness_rating
0	0	138.133208	Private room	False	True	2.0	True	1	0	10.0
1	1	124.061914	Private room	False	True	2.0	True	1	0	10.0
2	2	194.183865	Private room	False	True	4.0	True	1	0	10.0
3	3	191.604128	Entire home/apt	False	False	4.0	False	0	1	9.0
4	4	327.861163	Entire home/apt	False	False	6.0	False	1	0	9.0

Figure 4 - First 5 rows of the Lisbon_weekdays.csv

guest_satisfaction_overall	bedrooms	dist	metro_dist	attr_index	attr_index_norm	rest_index	rest_index_norm	lng	lat
98.0	1	4.328029	0.298484	74.230170	2.450656	176.868292	7.910210	-9.14034	38.75137
97.0	1	4.465504	0.293603	72.570845	2.395875	173.909352	7.777875	-9.14092	38.75260
87.0	2	4.475232	0.167860	72.517895	2.394127	177.996137	7.960651	-9.14245	38.75264
89.0	1	0.850978	0.589445	409.695507	13.525805	806.463205	36.068044	-9.13000	38.71100
93.0	2	0.667029	0.530351	537.015471	17.729183	775.718380	34.693021	-9.13200	38.71300

Figure 5 – First 5 rows of the Lisbon_weekdays.csv (Continuation)

We can see that the column “Unnamed: 0” seems to be serving as ID to each entry, but we can see just by looking at these first entries that the ID’s don’t match between csvs, since ID 3’s coordinates don’t match. We will delete the column Unnamed: 0 and replace with a proper ID variable once we merge both csvs.

Before we proceed with analysing our data, we decided to merge it for convenience, ease of use, and also to simplify the querying methods our chatbot might need to use. If all the information he needs is in a single file, it will be simpler for it to use.

Before proceeding with data analysis, we chose to merge the weekday and weekend datasets into a single unified structure, for ease of use, convenience and consistency. By consolidating each Airbnb listing into a single row (based on shared geographic coordinates, as explained further on), we ensured that all relevant information would be available in one place. This structure reduces redundancy, avoids potential confusion from duplicate entries, like for when one Airbnb has weekday and weekend prices, and enables the chatbot to compare or

retrieve day type specific attributes (such as pricing or room type) without requiring cross-referencing between multiple files.

After having deleted the “Unnamed:0” column from both csvs, we can identify the best possible merging attributes by studying our data in a more numerical way.

#	Column	Non-Null Count	Dtype
0	realSum	2906 non-null	float64
1	room_type	2906 non-null	object
2	room_shared	2906 non-null	bool
3	room_private	2906 non-null	bool
4	person_capacity	2906 non-null	float64
5	host_is_superhost	2906 non-null	bool
6	multi	2906 non-null	int64
7	biz	2906 non-null	int64
8	cleanliness_rating	2906 non-null	float64
9	guest_satisfaction_overall	2906 non-null	float64
10	bedrooms	2906 non-null	int64
11	dist	2906 non-null	float64
12	metro_dist	2906 non-null	float64
13	attr_index	2906 non-null	float64
14	attr_index_norm	2906 non-null	float64
15	rest_index	2906 non-null	float64
16	rest_index_norm	2906 non-null	float64
17	lng	2906 non-null	float64
18	lat	2906 non-null	float64

#	Column	Non-Null Count	Dtype
0	realSum	2857 non-null	float64
1	room_type	2857 non-null	object
2	room_shared	2857 non-null	bool
3	room_private	2857 non-null	bool
4	person_capacity	2857 non-null	float64
5	host_is_superhost	2857 non-null	bool
6	multi	2857 non-null	int64
7	biz	2857 non-null	int64
8	cleanliness_rating	2857 non-null	float64
9	guest_satisfaction_overall	2857 non-null	float64
10	bedrooms	2857 non-null	int64
11	dist	2857 non-null	float64
12	metro_dist	2857 non-null	float64
13	attr_index	2857 non-null	float64
14	attr_index_norm	2857 non-null	float64
15	rest_index	2857 non-null	float64
16	rest_index_norm	2857 non-null	float64
17	lng	2857 non-null	float64
18	lat	2857 non-null	float64

Figure 6 - Variable attributes of each csvs (Left - Weekend, Right - Weekday)

rest_index_norm	2906
attr_index	2906
attr_index_norm	2906
metro_dist	2906
dist	2906
rest_index	2906
lng	1862
lat	1757
realSum	651
guest_satisfaction_overall	46
cleanliness_rating	9
bedrooms	7
person_capacity	5
room_type	3
room_shared	2
biz	2
room_private	2
host_is_superhost	2
multi	2

rest_index_norm	2857
attr_index	2857
attr_index_norm	2857
metro_dist	2857
dist	2857
rest_index	2857
lng	1821
lat	1754
realSum	742
guest_satisfaction_overall	44
cleanliness_rating	9
bedrooms	7
person_capacity	5
room_type	3
room_shared	2
biz	2
room_private	2
host_is_superhost	2
multi	2

Figure 7 - Number of distinct value on each csvs (Left - Weekend, Right - Weekday)

Looking at these values, we can take a few conclusions. First, we can confirm once more that there are airbnbs of which we only have values about weekends, and some of which we only have values of weekdays. When considering how to handle this situation, we took into consideration three possible solutions. Since this data will be used to feed a chatbot, the LLM we are using are advanced and might be capable of telling the user they only possess information about weekends/weekdays for a specific Airbnb, and we could simply leave the

missing values as is. However, if we prefer to use a clean dataset, we can either drop the data with missing values, if its percentage is no greater than 5% of the total data, or we can duplicate entries that only have weekends/weekdays and fill in the price value with the avarege price of that combination of attributes. This last option could be faulty since some Airbnb's do in fact only offer their services on weekends or weekdays, not necessarily offering their services in both timeframes.

In the end, rather than removing or filling these missing values, we chose to simply leave them be, since we want to have as much accurate date as possible. This way, we will ensure we have the biggest possible database. It will be our intention to make the chatbot simply inform the user when he does not possess some requested information. Knowing this, we now must decide how we will be merging the dataset.

We saw 2 potential options to perform this step. We can merge the weekdays and weekends entries into a single entry, where each row represents one Airbnb, and we then separate the weekday and weekend prices by splitting the variable that holds the price of the Airbnb (realSum) into two variables (realSum_weekday and realSum_weekend, for example). This option preserves one entry per Airbnb in our dataset, avoiding the potencial of duplicate entries and making it so there are less entries for our chatbot to look through, and can also simplify how our chatbot looks for information specific to one of the day types. It is also a clearer structure that will facilitate our following changes and decisions. However, this approach will leave missing values in our data, and will create variables with similar names, although this shouldn't be an issue later on. Another option would be to duplicate airbnbs entries for each day type, making it so we have 2 entries per Airbnb, one for weekends and one for weekdays. This approach has the benefit of more effectively preserving the original dataset's structure and also makes it easier to perform different type of tasks on one type of day specifically, like modelling or prediction tasks, although this can also be achieved with the first option, but requiring a few extra steps, and it also simplifies the merging process of the csvs, by simply "glueing" both csvs together, adding one variable to separate weekdays from weekends. This option, however, will have duplicate attributes in some columns, like the lat/lng of each entry, it will require the chatbot to filter for ID and day type, adding logical complexity, which should always be avoided to keep the chatbot fast and consistent.

The option we deemed more beneficial ended up being the first one, since it reduces duplication and simplifies the retrieval of information for the chatbot, allowing it to present information for both days from the start, unless the user specifies one of the day types, and better being able to identify when an Airbnb only has one of the day types available.

To determine whether entries from the weekday and weekend datasets referred to the same Airbnb listing, we selected latitude and longitude as the matching key to use as a temporary ID. This approach is grounded in the assumption that the geographic location of a listing remains consistent across day types. While we considered the possibility of slight coordinate variations due to data collection or platform adjustments, we chose not to implement fuzzy

spatial matching. This was to avoid the risk of incorrectly merging distinct listings — such as different apartments within the same building — and to maintain a clear and robust merging logic. As such, only listings with identical coordinates were merged. Slightly offset entries were treated as separate listings, with the understanding that minor location differences are expected in real-world datasets and do not compromise the chatbot's utility. For variables that are stagnant and don't change from weekend to weekdays, we will simply leave one variable, but for attributes that are client dependant, like the guest satisfaction or cleanness rating, we will have to decide what to do. To aid our decision, we checked whether these values change a lot from weekday to weekend, if the values change a lot, it may be safer to include these values distinctly for weekdays and weekends. If the difference is minimal, we could simply keep the values and average out the weekends with weekdays values. We rounded these values to the second decimal (except the lat/lng), since anything lower then that ends up becoming irrelevant. To properly merge the data and later on decide how to handle these variables, we first gave variables coming from the weekdays the suffix _day, and from the weekend the suffix _end.

Before proceeding with the merge, we checked how many unique Airbnb locations were in the datasets and noticed that some locations had more than one Airbnb associated with them. To try and investigate why this is the case, we decided to analyse the duplicate locations in a different dataset, to allow a better grasp of what type of Airbnb's show location duplication and came up with the following visualization.

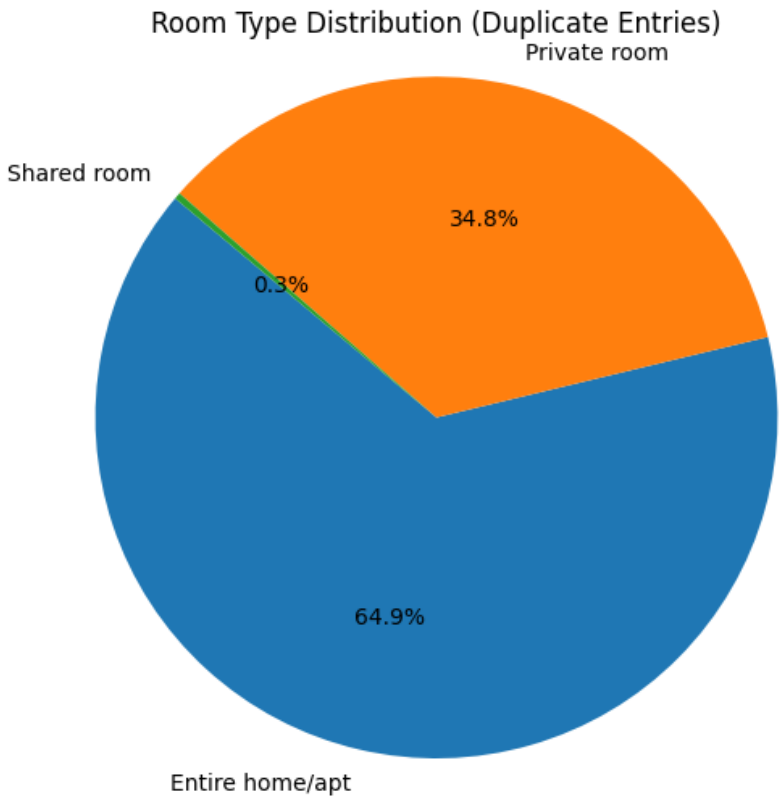

Figure 8 - Room Type Distribution for duplicate entries

For duplicated lat/lng entries that correspond to rooms, one can assume they simply referred to the same house, but each entry was a different room of said house. Since entire homes and apartments also seem to be showing duplicated locations, and we couldn't reach any conclusions as too what this case meant, we decided to only keep the first instance of each geographical location in our dataset, for both entire home/apartments and shared/private rooms, both to avoid potential issues this duplicates might make, and to simplify the development of our PoC, as our objective with this project is to show a working chatbot. Data that we would use in a official final product would go through a more robust data processing phase and would also include more information about certain things, information that is not abundant with our current database. This decision was also based on the geographical location of the Airbnb's being our best shot at providing each Airbnb with an unique ID, but if there are duplicates of this unique combo, that is the geographical location of each Airbnb, this wouldn't be possible.

Variable	Match_Percentage
room_shared	99.947862
dist	99.478624
metro_dist	99.478624
room_private	99.217935
room_type	99.217935
multi	99.113660
host_is_superhost	98.644421
biz	98.488008
bedrooms	97.288843
cleanliness_rating	96.819604
person_capacity	96.611053
guest_satisfaction_overall	91.970803
attr_index	76.746611
rest_index	57.507821
attr_index_norm	43.378519
realSum	1.772680
rest_index_norm	0.000000

Figure 9 - Percentage of Variables that match

After the merge, we were left with more listing than any of the two databases had separately. This is because of, like previously mentioned, some Airbnb's don't have a listing for both weekends and weekdays. We know must decide what to do to our variables, in order to simplify our database. First, we decided to make a script that would tell us the percentage of values that match between the weekday values and weekend values, and our results were:

Considering these values, our initial approach will be to merge the values of the variables dist and metro_dist using their average, which reflect geographic distance, and since it doesn't make sense that one Airbnb has a different distance from a point depending on the weekday, we used their average and merged them, since when looking into these, they often deviated very minimally, which was probably simply due to minimal discrepancies on data collection. There were, however, several variables that exhibited differences between weekday and weekend entries that could be explained, including room_type, person_capacity, cleanliness_rating, and guest_satisfaction_overall. This could be because some hosts might want to take their airbnbs in different directions during weekends/weekdays and, as such, these can understandably be different. These were retained in both versions (room_type_weekday, room_type_weekend).

The same cannot be said to the host-related attributes host_is_superhost, multi, and biz. Although these occasionally varied, they represent fixed characteristics of the host rather than the listing itself that don't make sense to be different just because it is a different day of the week. For these fields, we handled discrepancies by standardizing them, prioritizing the value True when inconsistencies arose, under the assumption that host classification is more likely to be added than removed.

For the variables attr_index and rest_index, the inconsistency noticed between weekdays and weekends could be explained by the different stores, restaurants and touristic attractions that are open during these times being different, and as such, we decided to also leave them separated.

After performing the merge, we investigated simplifying our data and correcting discrepancies unrelated to the merging of our data.

It is noticeable that the existence of both a dummy variable to identify wether a room is shared or not (room_shared) and whether a room is private or not (room_private) seems redundant, we can take a look at these variables and compare it to room_type to be able to determine how we can merge these 3 variables into a less redundant single dummy variable.

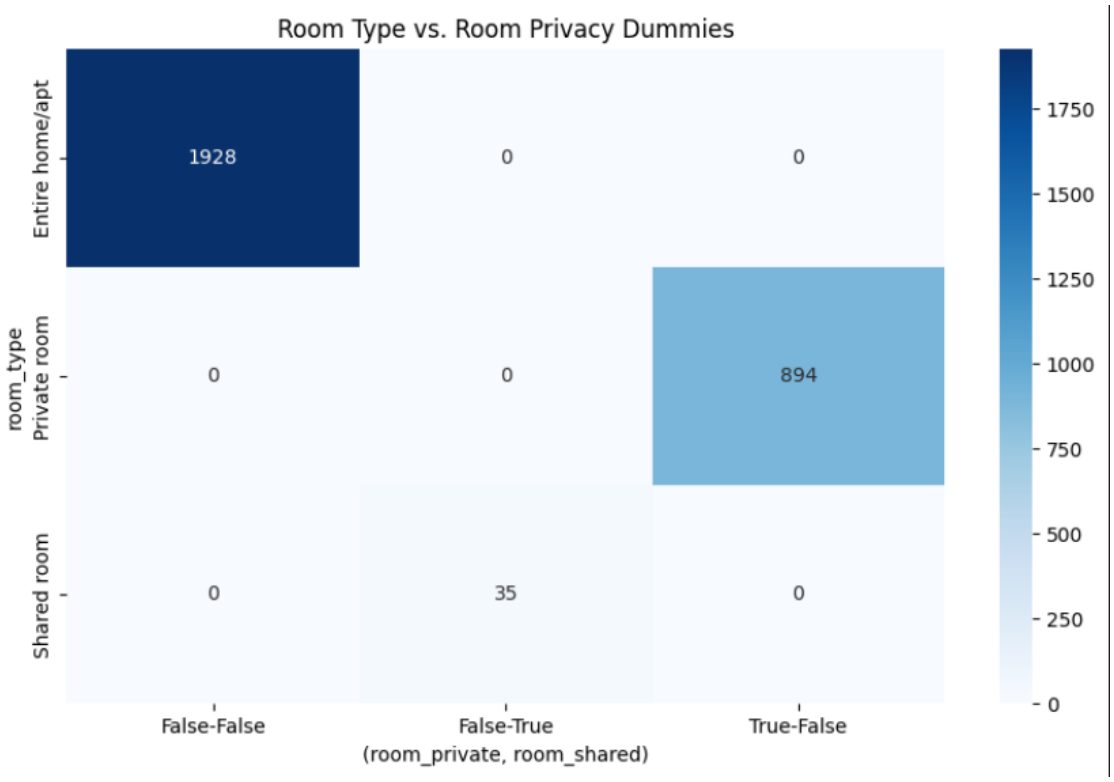

Figure 10 - Heatmap displaying combination of room_private and room_shared depending on the room_type, during weekdays.

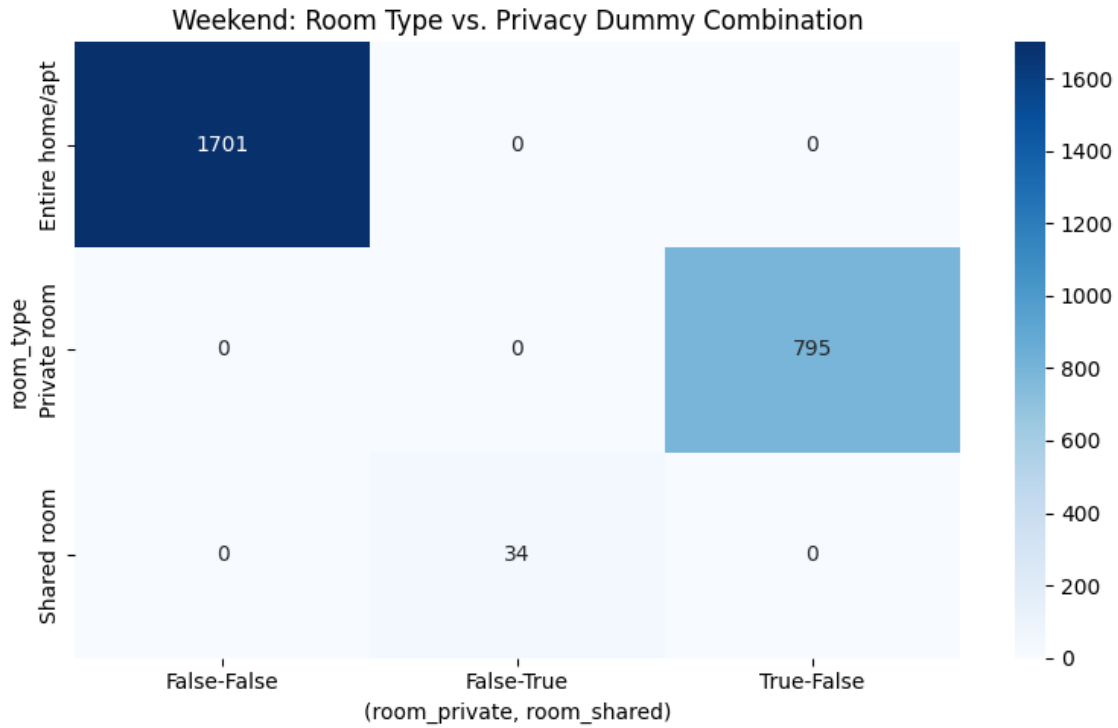

Figure 11 - Heatmap displaying combination of room_private and room_shared depending on the room_type, during the weekends.

Looking at these visualizations, we see that each combination of these variables always has the same outcome, both for the weekdays and weekends. When looking at entire home/apartments, they always returned False to both the room being private and the room being shared, likely due to the Airbnb not being a single room but a house/apartment, which includes more than one room that the user will probably share with his friends/family coming with him.

For private rooms, the room_private variable and the room_shared variable always returned True-False, as it was expected, and the shared_room variable False-True, also to be expected. Since these 3 variables always have these 3 combos of values, we can merge the Room type variable with the 2 dummy variables, and make a single variable with 3 possible values, “Entire home/apartment”, “Private Room” and “Shared Room”.

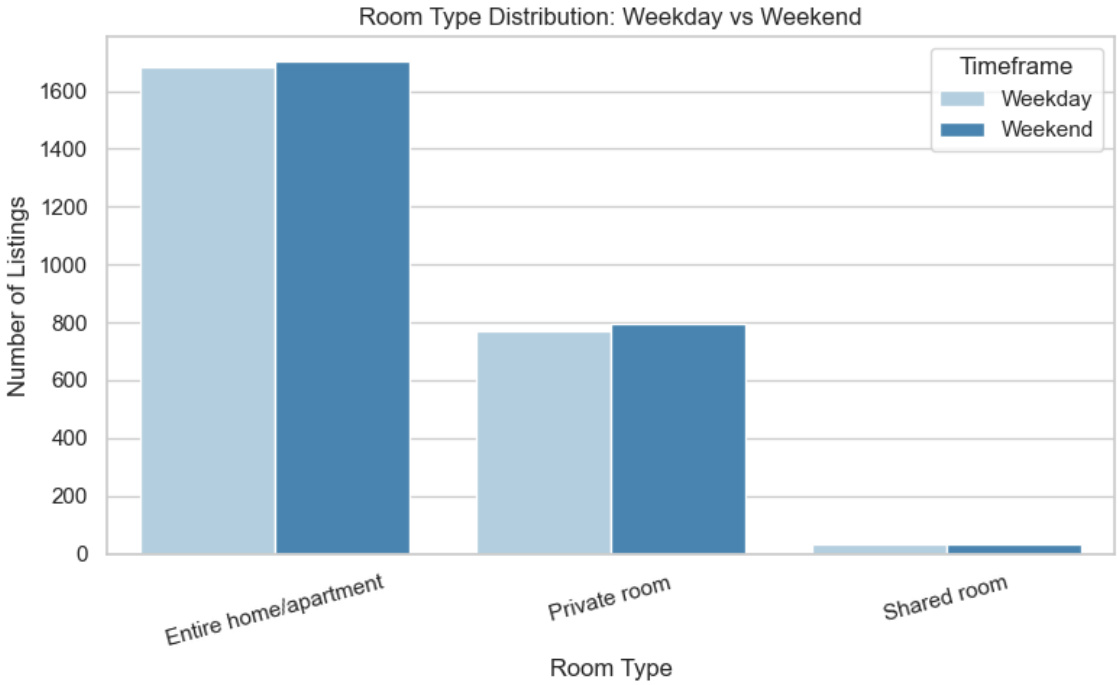

Figure 12 - Room Type Distribution for Weekdays and Weekends

As we previously noted, an Airbnb’s room type can change from weekday to weekend, so we will continue to keep these values separated, into the variables “Accomodation_type_weekday” and “Accomodation_type_weekend”. We can also see how many of each time of accommodation there is more effectively with this change.

We can conclude that the majority of the airbnbs are “Entire home/apartments, and that shared rooms are uncommon.

Last but not least, we are looking at the variables attr_index, attr_index_norm, rest_index and rest_index_norm. The attr_index and rest_index variables seem to represent a score of how good the surrounding attractions and restaurants are, and the higher this value, the better attractions and restaurants, respectively, are in the area near the Airbnb. The

attr_index_norm and rest_index_norm variables represent the same value but normalized on a scale from 0 to 100. To get a better visualization of our data, we started by creating a histogram of these values, to better guide our next decisions.

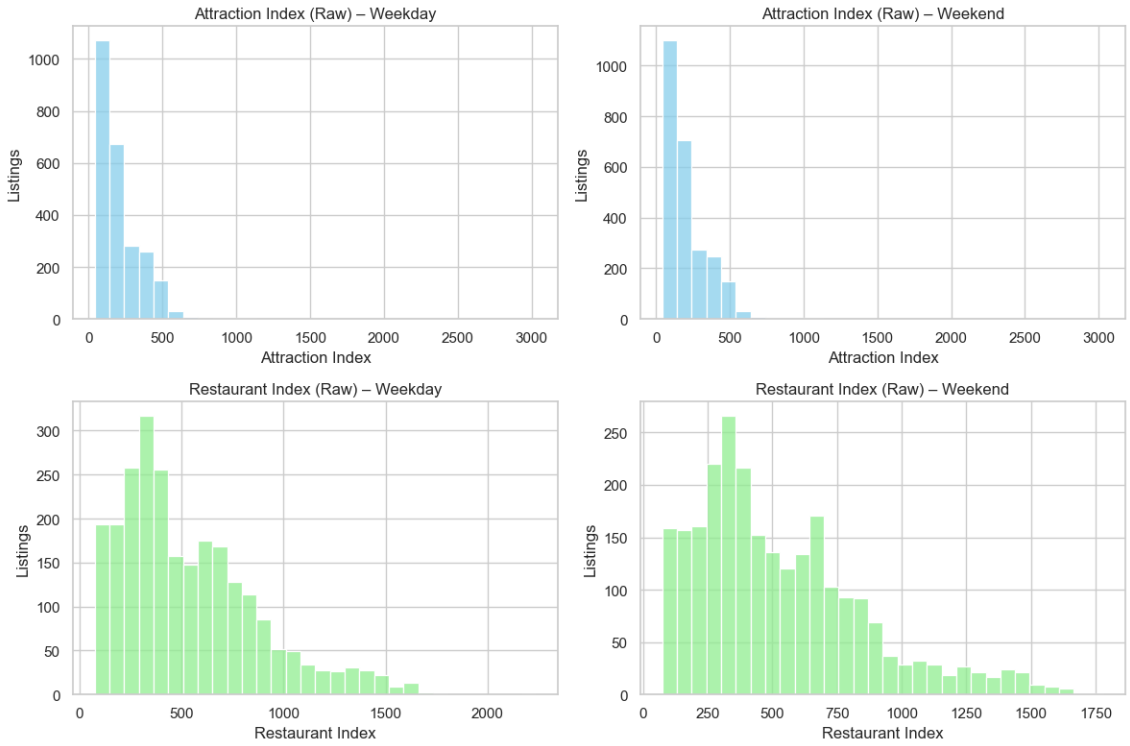

Figure 13 - Distribution of indexes for both weekends and weekdays

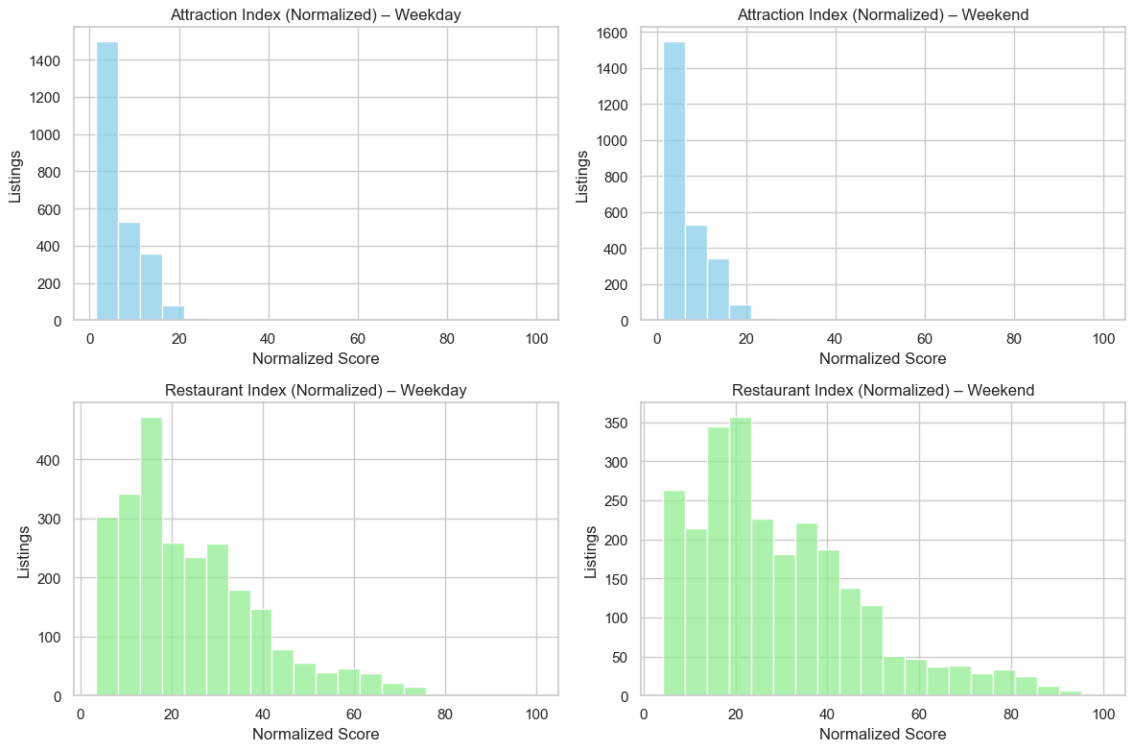

Figure 14 - Distribution of normalized indexes for both weekends and weekdays

Taking a first look at these histograms, we can conclude that these scores are heavily right skewed, making them difficult to interpret and possibly passing the wrong idea to the chatbot. This is to be expected in our data context, since there are urban areas that have way more dense attraction and restaurant population then others, for example, residential areas have way lower values in these scores.

Since these values are too specific and someone without context might struggle to understand what the values mean, we decided to convert this variable into a categorical variables and split the data into 5 different values. This categorization was created manually through an interpretation process based on visual study of the data distributions rather than using rigid percentiles. The objective was to ensure that each class contained enough listings to be analytically valuable while forming meaningful groupings that corresponded with the data's natural breakpoints. We decided to base our new scale on the raw values instead of normalized, so we can better control which values to include, and we settled upon the following scale:

Table 1 - Scale for new categorical variable representing the Attraction Index

Category	Range
Very low	< 100
Low	100 - 200
Average	200 - 350
Good	350 - 600
Excelent	> 600

Table 2 - Scale for new categorical variable representing the Restaurant Index

Category	Range
Very low	< 100
Low	100 - 400
Average	400 - 700
Good	700 - 1100
Excelent	> 1100

After this change, we can look at the following graphs to understand the new distribution of listings across the 5 categories.

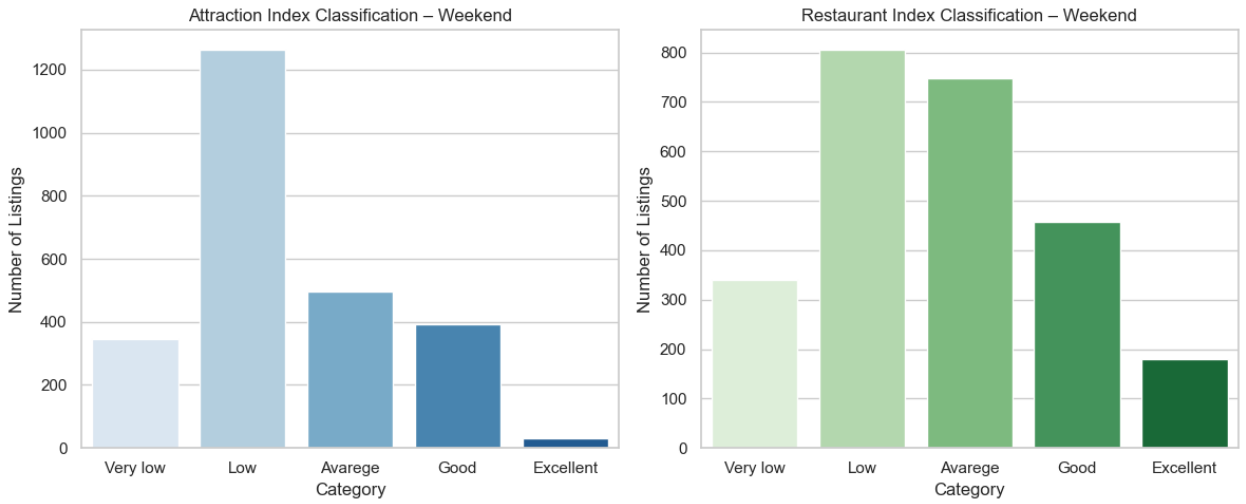

Figure 15 - New distribution of listings for weekends

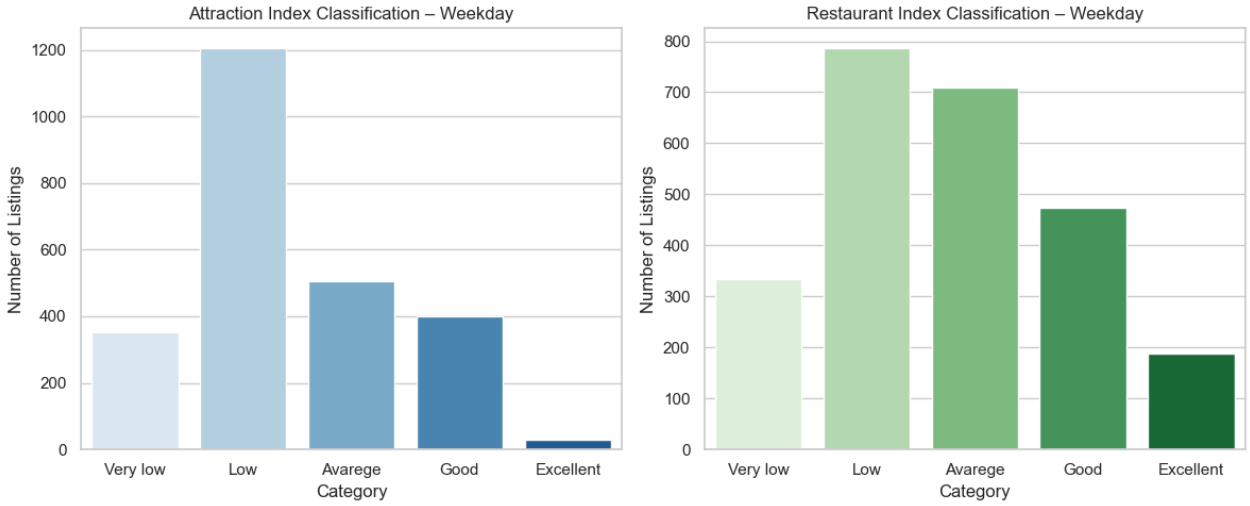

Figure 16 - New distribution of listings for weekdays

We can see that only a small percentage of listings make it to the Good or Excellent, classes for the attraction index, with the great majority falling into the Low to Average categories. This suggests that most listings are found in more residential or quieter areas, and that being close to tourist attractions is relatively rare.

For the restaurants index, the results are relatively similar, with the good and Average category having significantly having more listings but still falling short of the Low category. This suggests that restaurants are spread more evenly over the city, and that most Airbnb's have some restaurants nearby.

To finish preparing our data, we created a new column named ID so that we have a unique identifier for each listing that doesn't rely on 2 variables.

4.5 MODELLING

To prepare the Airbnb dataset for use in a RAG system, each listing was transformed into a standalone, structured natural-language description. These descriptions were uniform in tone and format, clearly distinguishing between weekday and weekend availability and including fields such as room type, price, guest capacity, cleanliness, satisfaction rating, and distance to the metro and city center. The listings were stored in a .txt file through a python script that uniformed all entries, each Airbnb listing was written as a complete paragraph with clear internal structure and consistent field ordering, separating them with specific indentation for easier chunking, and making them compatible with document loaders that perform chunking and embedding in langflow's UI. Each listing became a semantic unit and chunk for rag to look through.

To support semantic search, the system required a vector database capable of storing and querying vector databases. AstraDB was chosen for this purpose due to its ease of integration with LangChain and its managed infrastructure. Listings were embedded using OpenAI's embedding model, then stored in AstraDB's vector database. These databases allow cosine similarity search over text embeddings, meaning user queries can be matched semantically to relevant chunks of the PDF, even if no exact keyword match exists.

The chatbot follows a classic RAG pipeline:

1. User submits a natural language query.
2. The query is embedded and used to search the vector database (AstraDB).
3. The top k matching chunks are returned as "context".
4. A prompt template combines the user query and retrieved context.
5. The language model (GPT-4.1 mini) generates a final answer.

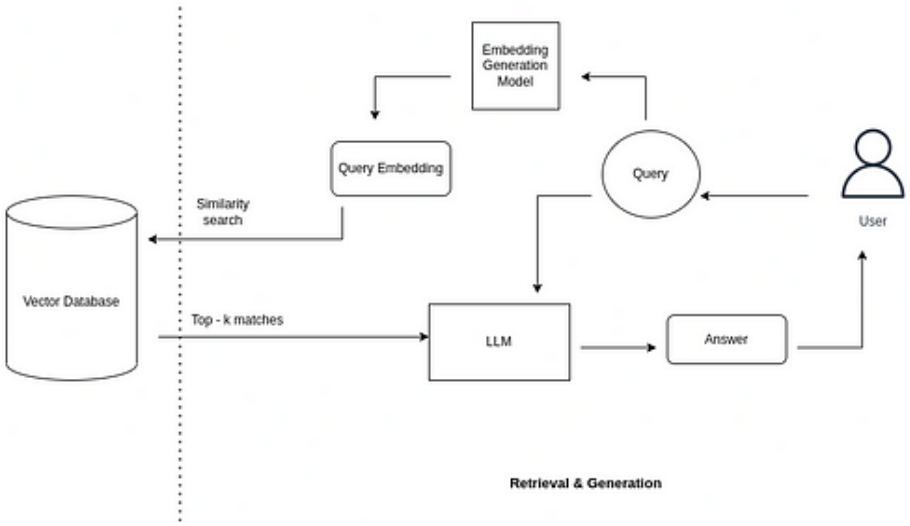

Figure 17 - RAG Diagram, Taken from https://netraneupane.medium.com/retrieval-augmented-generation-rag-26c924ad8181

A prompt can be built and given to our LLM to instruct it on how to handle user requests and how to respond to them. We made this prompt to instruct the chatbot on how to format responses, how to filter listings based on user requested attributes, how to handle vague or incomplete queries, how to explain when info we didn't have was requested, and to divert questions unrelated to the purpose of our chatbot. We made this prompt throught multiple tests and assistance of users to try and discover flaws on our chatbot.

To make the chatbot accessible to external users, it was deployed using Ngrok, a tunneling service that exposes a local server to the internet via a secure public URL, which allowed hosting the chatbot locally while making it available for real-world testing.

A Webpage was made available to contacted volunteers that once accessed, would display a form created to gather feedback from said users and where they could also interact with the chatbot.

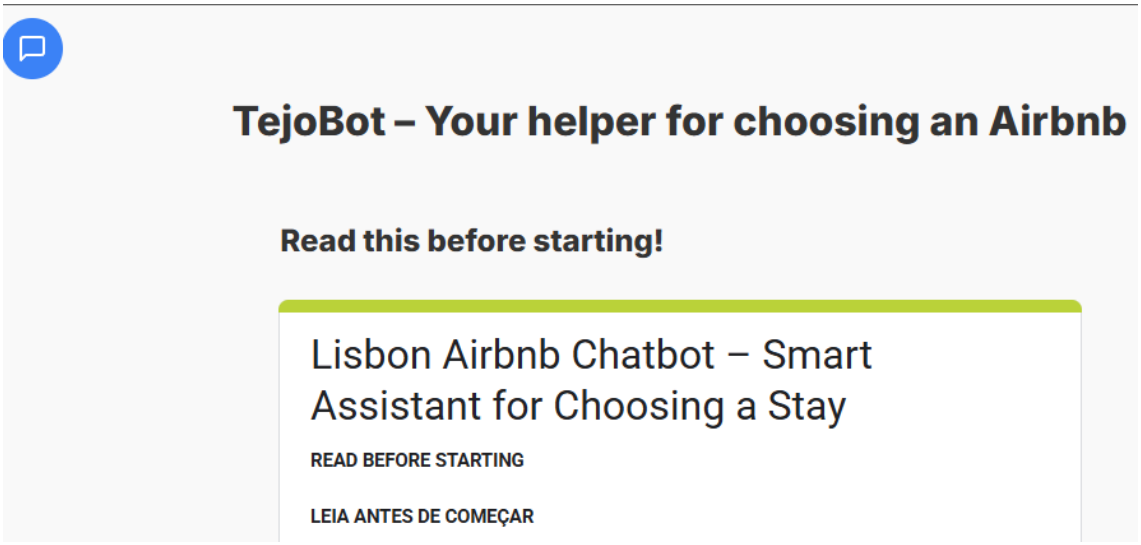

Figure 18 - Sample of Chatbot UI

Our system still has some constraints due to our limited software selection and budget, and the langflow environment having no option to edit its UI on the embedded website. Another matter is that our chatbot cannot access the entirety of the provided database in one go, usually being limited to 400 specific listings per session chosen by the LLM without criteria. This is due to our limited software and inability to edit certain aspects of the chatbot's structure.

4.6 Evaluating our chatbot

To evaluate our system's usability, effectiveness and interest it could generate as a PoC, a google form was created to gather feedback

Our form's questions can be divided into 6 main categories: Ease of use, questioning how easy it was for users to access and interact with the chatbot, the chatbot's comprehensive skills and response accuracy, evaluating whether the chatbot understood the questions correctly and returned relevant or coherent answers. Helpfulness and Informational value, where we asked users if they thought the current information the chatbot had access to was meaningful and if it was sufficient to help decide, Interpretability of the chatbot, where we asked how clear and fluent the language was used by the chatbot, and if it was a "user-friendly" language. We also asked for feedback on how Errors were handled and any limitations the chatbot demonstrated, asking about any confusions made by it or if any interactions were frustrating, in hopes of using this feedback to improve the bot. Lastly, we also asked for suggestions and interest in the continuous use of the chatbot, hoping to understand what users thought a chatbot of this kind should add as a priority and also to evaluate how much interest it generated.

The feedback is essential for assessing both the performance of the current prototype and identifying the most valuable directions for future development. The survey was offered in both Portuguese and English.

The Google Form had 25 volunteers interacted with the chatbot via a public webpage and as previously described, the system was made accessible using Ngrok to expose a LangFlow hosted chatbot instance. Before accessing the chatbot, users were presented with a brief introductory message explaining that the bot was a proof of concept created for a Master's project, along with a description of the available data (price, location, cleanliness, guest satisfaction, etc.) and example prompts to try. They were informed that the bot could take around 10 seconds to respond and were encouraged to return to the form afterward to provide feedback.

The form was shared with people of different nationalities and academic backgrounds, hoping to get a more varied audience, and included both multiple choice questions evaluating certain aspects of the bot and open text qualitative questions in hopes of gathering insight about the bot.

5 RESULTS AND DISCUSSION

5.1 EASE OF USE

To access and interact with our chatbot and to answer the questions of the form, users would have to access a link given to them that would display a webpage with the google form in question, and a small chat button on the top left corner of the website. This is where they would interact with the chatbot.

With these first questions, our main goal was to see if this UI approach would be one that made the chatbot easy to spot and Interact with, to understand if it was intuitive or not.

Across computer, tablets and phone users, the experience was generally positive, most people using their mobile phones to interact with the chatbot, some using their computers, but none using tablets, likely due to the fall of their popularity. However, looking at the few people that did mention any kind of struggle, they were using exclusively phones, which suggests that even though the mobile UI does what it is intended to do, it could use improvements, as some users struggled with it. This is likely related to the limited langflow UI, that doesn't allow us to edit simple stuff like to which side the chatbot chat box opens, or what size it takes, like previously mentioned in deployment constrains.

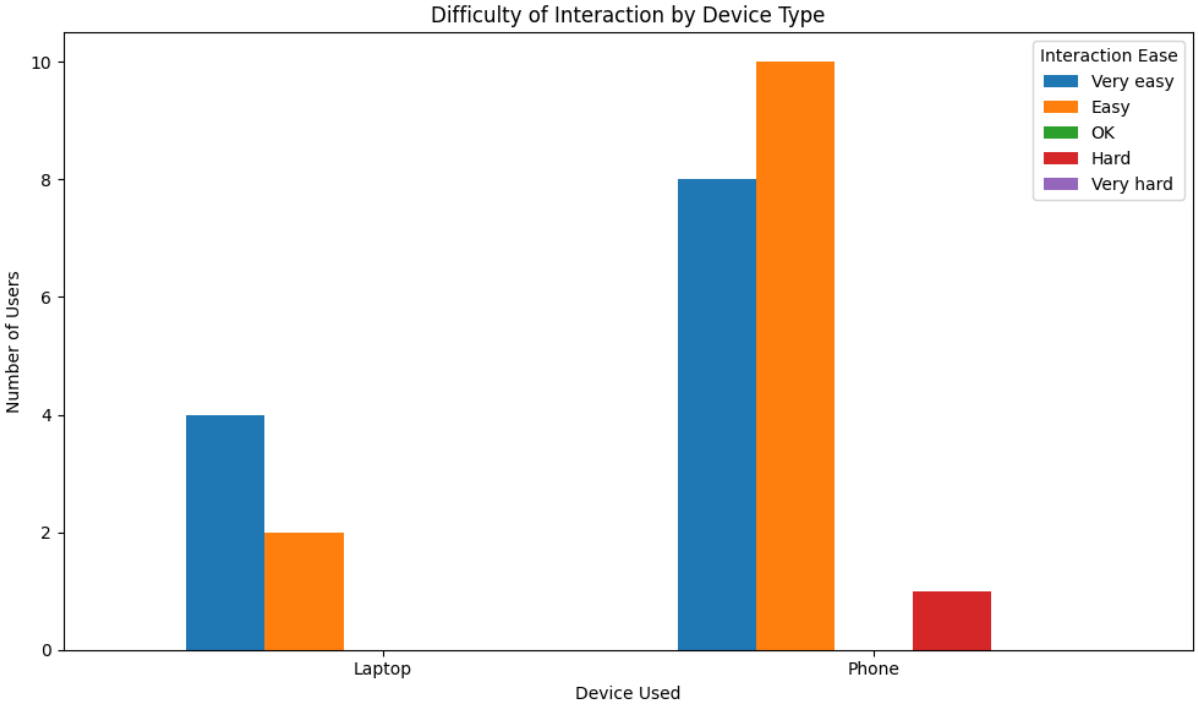

Figure 19 - Difficulty of Interaction by Device Type

5.2 Response Quality

This section assesses how well the chatbot understood user questions and how satisfied users were with the responses provided.

More than a third of the users reported that the chatbot understood all of their questions, but half the users responded that it understood almost all their questions. The remaining small percentage of users responded that the bot understood what they meant “Sometimes”. This could be because the users asked questions that the chatbot was not trained to answer, or simply scenarios we failed to anticipate. However, from this feedback, we can conclude that the chatbot is headed the right way and potentially needs only minor tweaks to its prompt template and updated information.

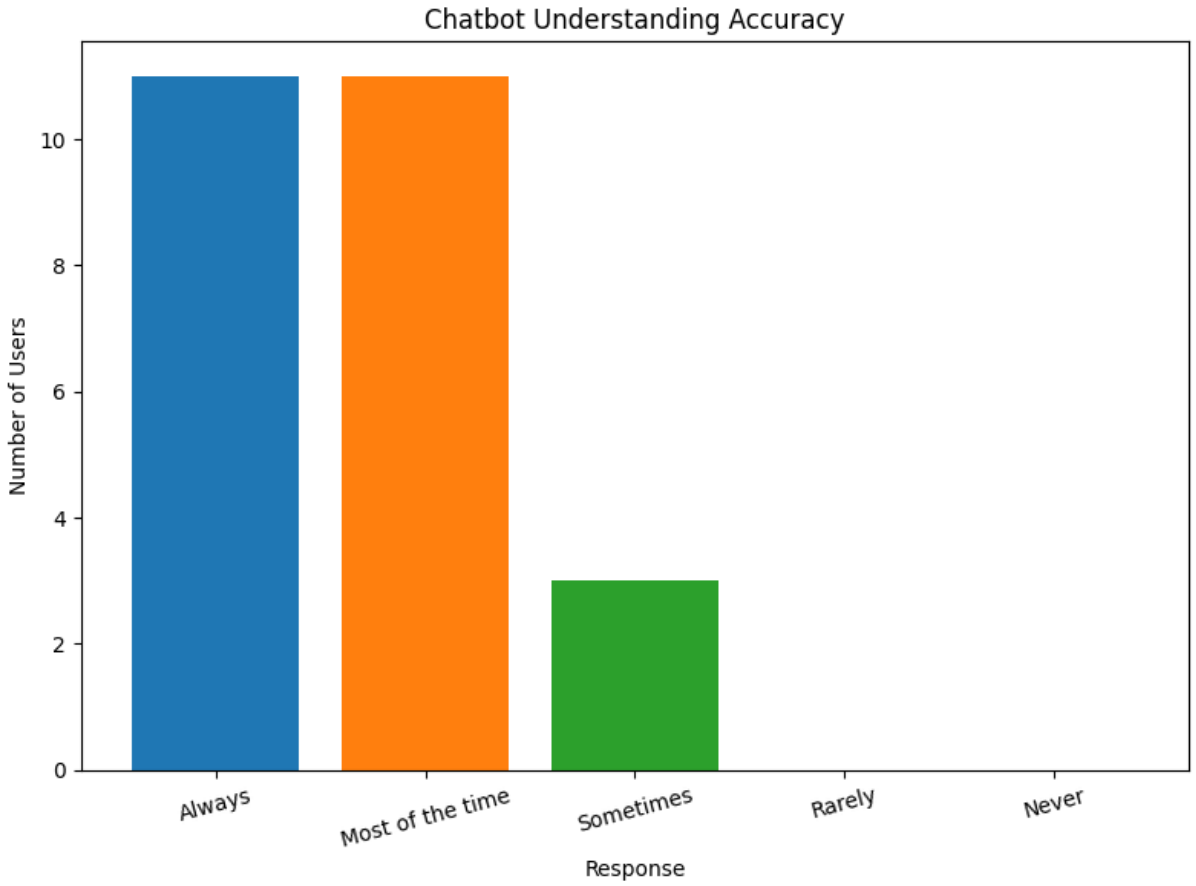

Figure 20 - Chatbot Understanding Accuracy

It is one thing for a chatbot to be able to understand what a user's "intent" is, but it is another to see if the chatbot gave an appropriate answer. From the feedback gathered, most users were "Very Satisfied" and "Satisfied" with the chatbot's answers, suggesting that replies were often consistent and align with the user expectations, with only a minimal percentage of users dissatisfied or neutral, pointing to isolated cases where either the information provided was insufficient in the user's eyes, or the quality or handling of the information did not satisfy the

user. These results point to the Proof of Concept currently attaining its goal of demonstrating a functioning product but also shows some room for improvement.

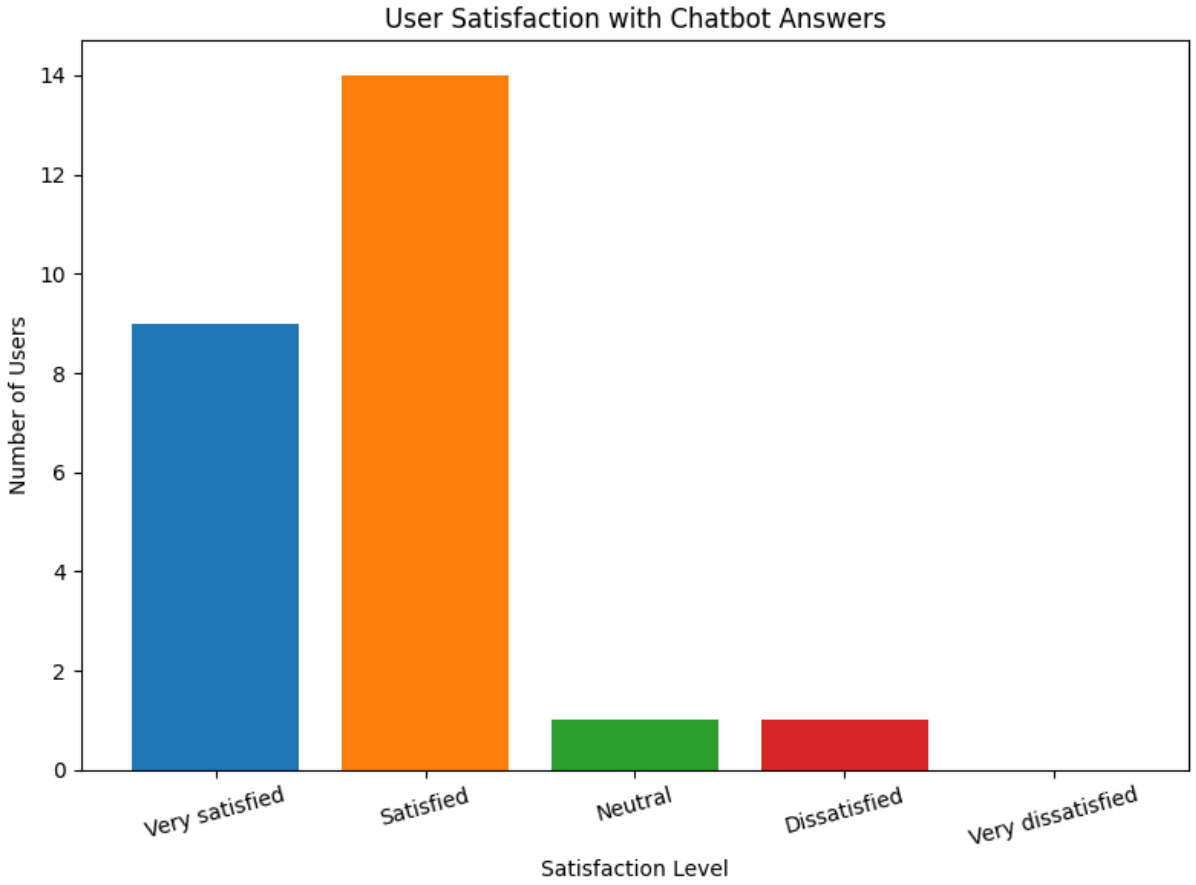

Figure 21 - User Satisfaction with Chatbot Answers

Still, some users provided important written feedback when asked about what failed. While most did not observe issues, the ones that did had a few common themes:

Some listings included more guests than requested, a possible cause could be on the prompt given to the LLM's configuration, or because the chatbot may treat the request properties has a minimum. For some of the less satisfied users, price filters were not always respected, likely caused either by the dataset's limited number of Airbnb's or by the previously referenced problem of the chatbot only having access to 400 entries at a time, chosen randomly by the chatbot, which could only include Airbnb's that could not meet the specified criteria. In multi-turn queries, the bot occasionally compared the wrong set of listings, calling back to its issue with conversation memory. Some users requested information about specific places in Lisbon, but that is information the chatbot is not currently trained with, so it either hallucinated or shrugged of the question.

These comments highlight the limitations of the current proof-of-concept setup, particularly around, Session memory (for ongoing conversations), Coverage gaps (from the dataset itself) and the Output formatting sometimes being inconsistent, starting to wonder off its template

after a few questions (long decimal values in price, when the chatbot was instructed to only have 2 decimals, or no longer giving the attributes of an Airbnb in its specified format).

5.3 INTERPRETABILITY

One of the objetives of this chatbot was to make it friendly and easy to talk to. Our data was treated and structured in a way that would help facilitate this approach from the start, and it was then fed to the chosen llm, who already had its own way of responding to prompts. After prompt engineering to modify some aspects of its responses, like limiting its replies to themes on topic, its tone and its clarity, we asked our users how they felt about the chatbot's dialogue, and the anwsers where overwhelmingly positive, with only a very small percentage of users saying somethings didnt feel easy to understand.

This suggests most users found the bot's tone friendly and the information clearly communicated, which validates our decision to use a templated but conversational prompt for the output generation.

Users responded positively regardless of whether they interacted with the bot in English or Portuguese, and there was no written feedback highlighting tone or clarity as a weakness.

5.4 CONTEXT AWARENESS

With these questions, we wanted to understand if the chatbot was able to access the necessary information to respond effectively to user questions and whether the provided information was helpful enough to support users in making decisions.

Across all questions, the responses were consistently positive. Most participants indicated that the chatbot had the relevant data “most of the time” or “always”, which correlates with its ability to access its database to answer different queries, said database that was organized in a structured pre-formatted template to help the LLM craft an answer to the user. Nearly all users reported that the information was useful when choosing an Airbnb, with responses ranging from "somewhat helpful" to "very helpful". None of the respondents said the information was not useful.

These results suggest that even with the known limitations of this proof of concept, the chatbot was effective in performing its current mission, by giving the relevant information.

We can confirm that these findings are consistent with feedback analysed so far, where the main criticism comes from limitations in the chatbot’s long-term memory and output formatting after a few messages, rather than the chatbot failing its mission. Within the limitations of our proof of concept, users perceived the system’s coverage and response content as both reliable and useful.

5.5 IMPROVEMENTS

In this section, our goal was to gather as much feedback as possible about the main limitations users noticed and what they felt like was the most urgent improvements that needed addressing and additional features to be added.

Although the text generated by our model was often clear, some users reported other types of struggles with the bot. Phone users struggled with the chat input box, which on smaller phones could be covered up by the phone's keyboard showing up, a situation that the UI does not automatically adapt to. Other users noted, like previously mentioned, that the chatbot seemed to lose its response formatting after a few queries were submitted, such as suddenly dumping all information on one large paragraph instead of the structured format it was taught, and sometimes hallucinating multiple decimal values of an entry's price, even though the provided data only had 2 decimal values, and the chatbot was told to not exceed these 2 decimal values. These issues come partially from langflow's limited UI design and from further prompt engineering being required to improve our chatbot, as well as perhaps investigate other LLM's to base our model in or even attempt to utilize finetuning. Other technologies besides langflow might also need to be explored, as langflow is the perfect way to be introduce oneself to the RAG architecture, but not the best for elaborate projects.

Despite all this, in terms of conversational performance, most users did not report mistakes. Some even noted that the chatbot often returned relevant information they hadn't explicitly requested, suggesting a helpful degree of proactivity. However, several users mentioned that the chatbot occasionally recommended listings that did not exactly match the previously specified filters, but technically complied with their requests, such as giving rooms that could house more guests than the requested number or slightly adjusting price limits to provide better search results. Other already mentioned aspects were reiterated here, like the lake of knowledge about the whereabouts of some places, and issues related to its longterm memory

When asked about what they would improve first, most users talked about structural improvements to the answers from the chatbot, with a little additional content that they felt could help. Ideas ranged from making formatting of the responses more consistent (particularly ensuring bullet points are kept and limiting decimals in prices), shortening time to answer, and expanding available listings beyond the city they are limited to now (Lisbon), to offering a map to see the locations visually. A few users also suggested improvements to the memory of the chatbot, to permit a larger number of conversations rather than receiving just responses.

This all matches with the criticised aspects of the chatbots, showing the users took the survey seriously and were able to provide meaningful feedback. Some users also suggested that the chatbot should include photographs of the suggested airbnbs and links to the Airbnb listing itself, as well as using their names instead of a numeric ID. Some other users also suggested expanding the knowledge base of the chatbot with information like number of bathrooms, pet friendliness, breakfast inclusion and accessibility. These are all suggestions that will help

us develop our chatbot with more relevant information for future versions of it, but some of the complains are also tied to the limited database we used. One user also felt important to mention that scales of evaluation used by the chatbot should be mentioned at least once, to better explain the information it is giving, which is a valid addition and would no doubt be beneficial.

All these suggestions of improvements mostly reflect already mentioned limitations, most of which we anticipated in development, but the feedback provided by users on aspects that could be improved and added was invaluable. With these, it's possible to form a road map of improvements to our chatbot, with it being clear that in a future developed iteration of the chatbot, it would benefit most from better memory, visual elements, and more filtering options. Revising our prompt in some respects is also a step we would take.

5.6 Overall User Perception and Satisfaction

To wrap up our chatbot's evaluation, we tried to explore what impression it had left on our users, addressing their satisfaction with its service, willingness to use it in future versions, and even if they would eventually recommend it to friends.

Regarding the users' interest in using the chatbot, they were first asked whether they would use this chatbot instead of manually searching for an Airbnb, assuming it had updated information, with most responses revealing a strong interest in doing so. When asked about the possibility of using an improved version of the system in a future trip to Lisbon, users expressed the same willingness to rely on the chatbot as a helpful tool in that context. These answers suggest that even in its current prototype stage, the chatbot already has the potential to be adopted by users in real scenarios, by simply adding a reliable database to it.

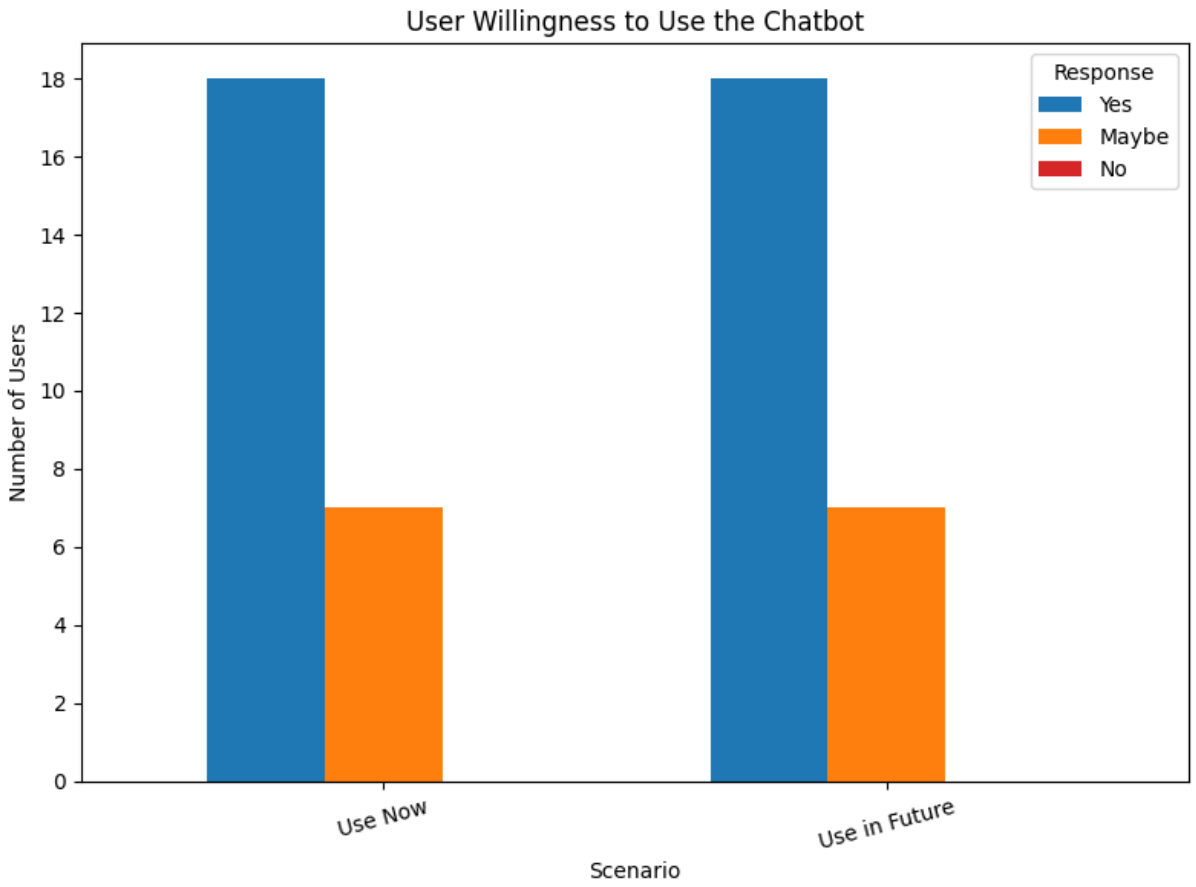

Figure 22 - User's willingness to use the chatbot again

When asked how overall satisfied they were with the experience with the chatbot, users marked their experience from a range of 1 to 10. There were the majority scores concentrated near the top of the range, which suggests that the experience not only was a positive one but also met the users' expectations. Even though there were a small number of users offering rather more neutral scores, the responses were in accordance with the notion that the system met its primary objectives.

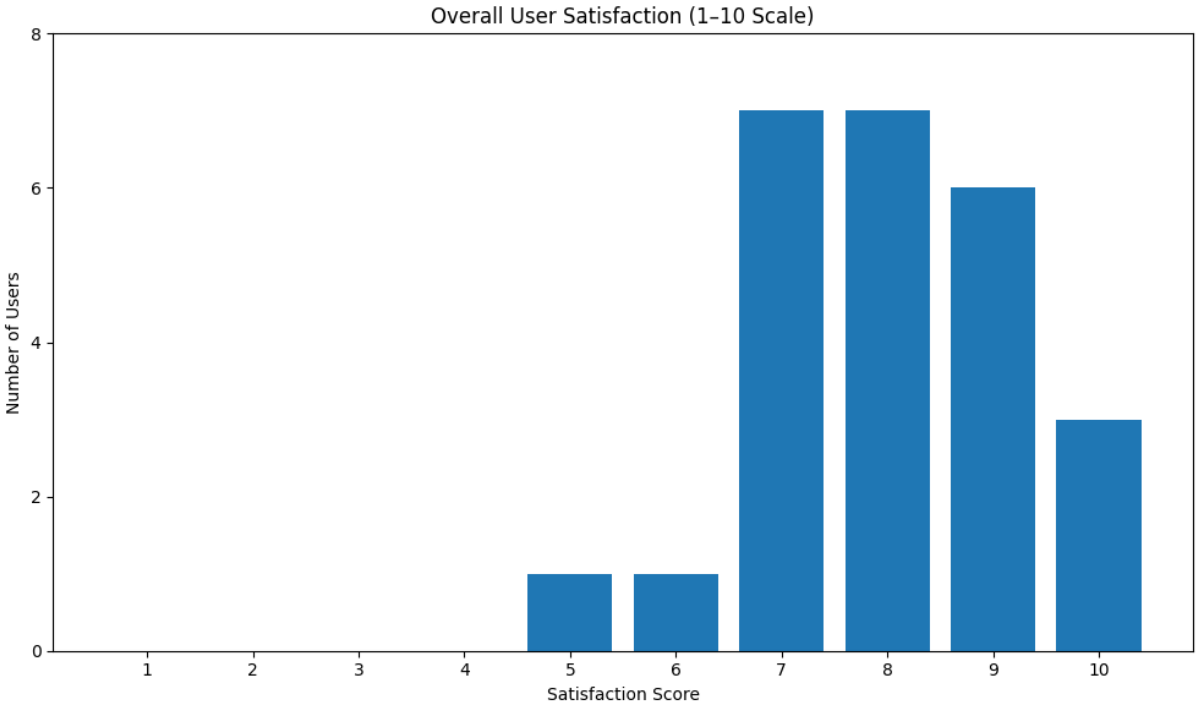

Figure 23 - User Satisfaction Score

Along with satisfaction, users were also asked if they would suggest this chatbot to a tourist coming to Lisbon. Once again, the scores speak for themselves, with them mostly skewing towards the higher end, but not as much as their overall satisfaction. This is good because for something like tourism, trips are usually inspired by either friends or influencers, so it being worthy of being shared can have a significant impact.

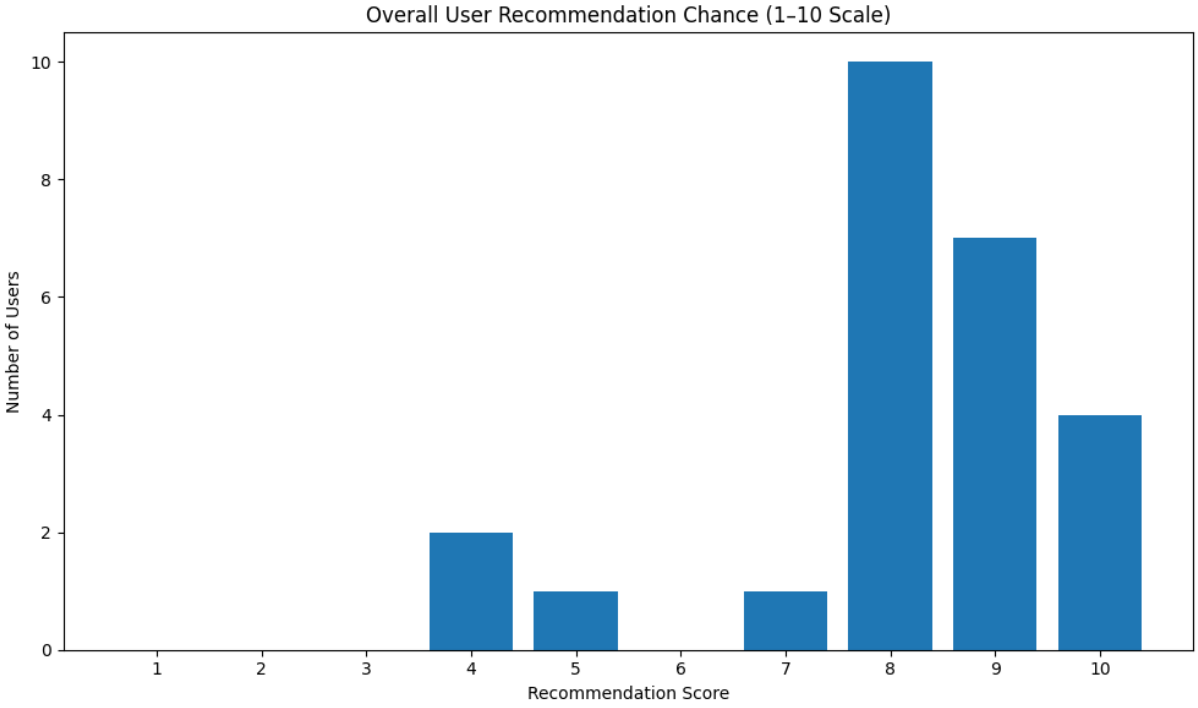

Figure 24 - User Recommendation Chance

Together, these three indicators point to a successful proof of concept. While areas for improvement remain, particularly in responsiveness, interface limitations, and memory handling, the system left a strong impression. The feedback confirms that a well-targeted, domain-specific chatbot can offer a satisfying and valuable experience to users.

After gathering and analyzing the feedback our users shared with us, we can note a few reoccurrences in their responses, both on strong points and weakpoints of our chatbot, and we can use these to improve our chatbot should we ever develop any future versions of it. There was a lot of positive feedback related to the chatbot being easy to access and interact with, with it being only slightly less reliable on mobile devices. This means that our objective for this PoC's user experience and general architecture we decided to use to deploy were effective and attained our goals. The chatbot was also praised for its ability to understand and properly respond to questions made by users with fluent and easily understandable grammar, with users commenting that the information was delivered successfully and effectively. Even thought our dataset contained some limitations due to its nature, most users still found that the information that we gave to the chatbot was useful and meaningful for decisions that they could be trying to make, which shows the potential of our RAG approach even with our primitive deployment.

There was also, of course, some constructive feedback that pointed to a few recurrent limitations, some we already knew we were going to have, others we didn't initially consider. The chatbot has short or non-existent memory of previous messages on a conversation with a user, so multi prompt conversations will feal weird since referring previous messages tends to fail, especially when mentioning previous listings the chatbot provided. Some users also noted that the format of the answers would start to disassemble once a session started getting long, with the bot losing some properties of the customized prompt we made for it, like changing from a list format to simply dumping all information in one paragraph. On mobile devices, particularly Safari on iPhone, some users experienced UI related issues, where once the keyboard showed up, it would cover the text box, so the users were unable to see what they were typing until they were done writing and could close the keyboard. These difficulties, although expected for a primitive project like ours, show room for improvement and clear paths to take for refinement.

From user suggestions we can also gather some useful information about improvements we can make to our system. Enhancing the consistency of our message output structure, increase the number of listings available per session and gathering more information for our dataset are all preliminary and obvious objectives we could accomplish on the next version of this chatbot. Furthermore, improving the UI of our system should also be a priority, mainly by developing an alternative mobile UI for users to use on their phones. Users also suggest some improvements to our database that would no doubt be implement in future versions, for example, adding the name of our Airbnb to the dataset instead of the number ID, providing links, images and more information about the airbnbs, the view of the airbnbs and other aspects.

6 CONCLUSIONS AND FUTURE RESEARCH

This project set out to investigate the potential of LLM chatbots in the tourism domain, by developing a proof of concept, in the form of a chatbot assistant trained with data to help visitors choose an Airbnb in Lisbon. While LLMs like ChatGPT and Bard have shown they can excel at most general tasks and provide lots of flexibility, they struggle with tasks that might make them rely on their sometimes outdated information or simply unavailable. Our goal when creating this chatbot was to create a chatbot that would address those flaws when handling the specific topic, we were going to train it for.

To achieve this, we designed and deployed our chatbot with a RAG system that transformed a dataset pre-processed by us about Airbnb data into natural language entries. These were then indexed and retrieved as context using a vector database, allowing the chosen LLM, ChatGPT 4o. to answer questions based on the data it was given, which was meant to simulate real updated listing information. A engineered prompt guided the bot's responses, enforcing a specific tone, structure, formatting, and filtering behaviour, and the system was deployed publicly via a LangFlow based web interface and tested by real users through a structured feedback form.

With the feedback gathered, we were confident to say that the proof of concept achieved its main objective of showing that an AI assistant could provide useful assistance in tourism environments. Users found the bot simple to operate and believed the bot correctly understood their questions and provided answers that were clear and helpful and well organized. The system delivered benefits to users despite its inability to process the full dataset simultaneously and conversation history. Most participants also mentioned that they would use a fully developed public version of this chatbot in their next trips should they be looking for assistance with choosing a room.

7 LIMITATIONS AND FUTURE PROJECTS

It was not expected for our project to face no constrains, and we did have a few. The langflow web interface, particularly on mobile devices, created some UI challenges, the bot was not always consistent with the given engineered prompt, and it often struggled with multi message questions due to its current inability of storing and consulting chat history with the user. Nonetheless, these issues were expected and highlighted given the available resources and did not prevent us from attaining our Proof of Concept's main goal: demonstrating a capable chatbot with information specific to tourism and showing how useful it could be for users struggling with some decisions.

Looking ahead, this proof of concept could serve as the foundation for a more robust travel assistant. With improved infrastructure, the bot could access real listings, incorporate user requested filters more effectively, remember conversation history, and even connect to booking platforms. Its architecture could also be repurposed to other public-facing services, such as city guides, museum information bots, or public transit assistants, wherever reliable, contextual information is more useful than general answers.

Looking ahead, our proof of concept sets a foundation for what could be a complete travel assistant system. Adding in access to live data through improvements in its infrastructure and incorporating some of the feedback users shared, it could become a top-of-the-line system to build an itinerary for users, without them having to struggle so much to schedule their vacations or trips. This "blueprint" could also easily be adapted to other scenarios that might benefit from a 24/7 online source of information.

Ultimately, this project illustrates that with the right design choices, even a limited RAG implementation can achieve strong results. LLM technology keeps evolving every day, and it's hard to keep up with it. As new technologies emerge, new and perhaps more optimized ways of developing tools like ours will likely appear. This constant evolution opens the door for future iterations of our chatbot to be more powerful, reliable, and accessible.

8 BIBLIOGRAPHICAL REFERENCES

Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. In I. Maglogiannis, L. Iliadis, & E. Pimenidis (Eds.), *Artificial Intelligence Applications and Innovations* (Vol. 584, pp. 373–383). Springer International Publishing. https://doi.org/10.1007/978-3-030-49186-4_31

Akma, N., Hafiz, M., Zainal, A., Fairuz, M., & Adnan, Z. (2018). Review of Chatbots Design Techniques. *International Journal of Computer Applications*, *181*(8), 7–10. https://doi.org/10.5120/ijca2018917606

Augello, A., Gentile, M., & Dignum, F. (2018). An Overview of Open-Source Chatbots Social Skills. In S. Diplaris, A. Satsiou, A. Følstad, M. Vafopoulos, & T. Vilarinho (Eds.), *Internet Science* (Vol. 10750, pp. 236–248). Springer International Publishing. https://doi.org/10.1007/978-3-319-77547-0_18

Benaddi, L., Ouaddi, C., Souha, A., Jakimi, A., & Ouchao, B. (2024). Model-Driven Engineering to develop chatbots for smart tourism. *2024 International Conference on Global Aeronautical Engineering and Satellite Technology (GAST)*, 1–5. https://doi.org/10.1109/GAST60528.2024.10520796

Brandtzaeg, P. B., & Følstad, A. (2017). Why People Use Chatbots. In I. Kompatsiaris, J. Cave, A. Satsiou, G. Carle, A. Passani, E. Kontopoulos, S. Diplaris, & D. McMillan (Eds.), *Internet Science* (Vol. 10673, pp. 377–392). Springer International Publishing. https://doi.org/10.1007/978-3-319-70284-1_30

Bruno Marietto, M. D. G., Aguiar, R. V., Barbosa, G. D. O., Botelho, W. T., Pimentel, E., Franca, R. D. S., & Da Silva, V. L. (2013). Artificial Intelligence Markup Language: A Brief Tutorial. *International Journal of Computer Science & Engineering Survey*, *4*(3), 1–20. https://doi.org/10.5121/ijcses.2013.4301

Bulchand-Gidumal, J. (2020). Impact of Artificial Intelligence in Travel, Tourism, and Hospitality. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-Tourism* (pp. 1–20). Springer International Publishing. https://doi.org/10.1007/978-3-030-05324-6_110-1

Calvaresi, D., Ibrahim, A., Calbimonte, J.-P., Schegg, R., Fragniere, E., & Schumacher, M. (2021). The Evolution of Chatbots in Tourism: A Systematic Literature Review. In W. Wörndl, C. Koo, & J. L. Stienmetz (Eds.), *Information and Communication Technologies in Tourism 2021* (pp. 3–16). Springer International Publishing. https://doi.org/10.1007/978-3-030-65785-7_1

Canonico, M., & Russis, L. D. (2018). A Comparison and Critique of Natural Language Understanding Tools. *CLOUD COMPUTING*.

Carvalho, I., & Ivanov, S. (2024). ChatGPT for tourism: Applications, benefits and risks. *Tourism Review*, *79*(2), 290–303. https://doi.org/10.1108/TR-02-2023-0088

Jeong, C. (2023). A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture. *Advances in Artificial Intelligence and Machine Learning*, *03*(04), 1588–1618. https://doi.org/10.54364/AAIML.2023.1191

Jiang, H., Cheng, Y., Yang, J., & Gao, S. (2022). AI-powered chatbot communication with customers: Dialogic interactions, satisfaction, engagement, and customer behavior. *Computers in Human Behavior*, *134*, 107329. https://doi.org/10.1016/j.chb.2022.107329

Jung, S. (2019). Semantic vector learning for natural language understanding. *Computer Speech & Language*, *56*, 130–145. https://doi.org/10.1016/j.csl.2018.12.008

Ketakee Nimavat, & Tushar Champaneria. (2017). Chatbots: An Overview Types, Architecture, Tools and Future Possibilities. *International Journal for Scientific Research and Development*, *5*(7), 1019–1024. https://ijsrd.com/Article.php?manuscript=IJSRDV5I70501

Kucherbaev, P., Bozzon, A., & Houben, G.-J. (2018). Human-Aided Bots. *IEEE Internet Computing*, *22*(6), 36–43. https://doi.org/10.1109/MIC.2018.252095348

Lai, W. Y. W., & Lee, J. S. (2024). A systematic review of conversational AI tools in ELT: Publication trends, tools, research methods, learning outcomes, and antecedents. *Computers and Education: Artificial Intelligence*, *7*, 100291. https://doi.org/10.1016/j.caeai.2024.100291

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks* (No. arXiv:2005.11401). arXiv. https://doi.org/10.48550/arXiv.2005.11401

LMSYS. (2024, May 8). Llama 3: An open reproduction and analysis. https://lmsys.org/blog/2024-05-08-llama3

Lu, W., Luu, R. K., & Buehler, M. J. (2025). Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities. *Npj Computational Materials*, *11*(1), 84. https://doi.org/10.1038/s41524-025-01564-y

Maglogiannis, I., Iliadis, L., & Pimenidis, E. (Eds.). (2020). *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II* (Vol. 584). Springer International Publishing. https://doi.org/10.1007/978-3-030-49186-4

Neupane, N. (2023, December 21). *Retrieval-Augmented Generation (RAG)*. Medium. https://netraneupane.medium.com/retrieval-augmented-generation-rag-26c924ad8181

Nguyen, H. T., Tran, T. T., Nham, P. T., Nguyen, N. U. B., & Le, A. D. (2023). AI Chatbot for Tourist Recommendations: A Case Study in Vietnam. *Applied Computer Systems*, *28*(2), 232–244. https://doi.org/10.2478/acss-2023-0023

OpenAI. (April 14, 2025). Introducing GPT-4.1 in the API https://openai.com/index/gpt-4o

OpenAI. (2024). GPT-4o: Hello GPT-4o https://openai.com/index/hello-gpt-4o/

Orlov, V., Tynchenko, V., Volneykina, E., Shutkina, E., & Stupin, A. (2024). RETRACTED: Developing a chatbot-based information system for employee interaction. *E3S Web of Conferences*, *549*, 08018. https://doi.org/10.1051/e3sconf/202454908018

Palihapitiya, C. (2023, December 18). *A short history of OpenAI*. Chamath's Substack. https://chamath.substack.com/p/a-short-history-of-openai

Pillai, R., & Sivathanu, B. (2020). Adoption of AI-based chatbots for hospitality and tourism. *International Journal of Contemporary Hospitality Management*, *32*(10), 3199–3226. https://doi.org/10.1108/IJCHM-04-2020-0259

Ramesh, K., Ravishankaran, S., Joshi, A., & Chandrasekaran, K. (2017). A Survey of Design Techniques for Conversational Agents. In S. Kaushik, D. Gupta, L. Kharb, & D. Chahal (Eds.), *Information, Communication and Computing Technology* (Vol. 750, pp. 336–350). Springer Singapore. https://doi.org/10.1007/978-981-10-6544-6_31

Rather, R. A. (2025). AI-powered ChatGPT in the hospitality and tourism industry: Benefits, challenges, theoretical framework, propositions and future research directions. *Tourism Recreation Research*, *50*(3), 652–662. https://doi.org/10.1080/02508281.2023.2287799

Thomas, N. T. (2016). An e-business chatbot using AIML and LSA. *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2740–2742. https://doi.org/10.1109/ICACCI.2016.7732476

Wallace, R. S. (2009). The Anatomy of A.L.I.C.E. In R. Epstein, G. Roberts, & G. Beber (Eds.), *Parsing the Turing Test* (pp. 181–210). Springer Netherlands. https://doi.org/10.1007/978-1-4020-6710-5_13

Wang, L., Chen, S., Jiang, L., Pan, S., Cai, R., Yang, S., & Yang, F. (2025). Parameter-efficient fine-tuning in large language models: A survey of methodologies. *Artificial Intelligence Review*, *58*(8), 227. https://doi.org/10.1007/s10462-025-11236-4

Zlatanov, S., & Popesku, J. (2019). Current Applications of Artificial Intelligence in Tourism and Hospitality. *Proceedings of the International Scientific Conference - Sinteza 2019*, 84–90. https://doi.org/10.15308/Sinteza-2019-84-90

9 APPENDIX A

This is to certify that
Project No.: **INFSYS2025-7-99569**
Project Title: **Chatbot for Lisbon Tourism A Proof of Concept Using LLMs and Retrieval-Augmented Generation to Assist on choosing Airbnb**
Principal Researcher: **Vasco Lhansol Souto Massapina**

According to the regulations of the Ethics Committee of NOVA IMS and MagIC Research Center this project was considered to meet the requirements of the NOVA IMS Internal Review Board, being considered **APPROVED** on 7/9/2025.

It is the Principal Researcher's responsibility to ensure that all researchers and stakeholders associated with this project are aware of the conditions of approval and which documents have been approved.

The Principal Researcher is required to notify the Ethics Committee, via amendment or progress report, of
- Any significant change to the project and the reason for that change;
- Any unforeseen events or unexpected developments that merit notification;
- The inability of the Principal Researcher to continue in that role or any other change in research personnel involved in the project.

Lisbon, 7/9/2025

NOVA IMS Ethics Committee
ethicscommittee@novaims.unl.pt

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa