Comparative Analysis: Machine Learning vs. AI

1. Supervised Learning using Linear Regression

1.1 Part 1: Machine Learning

In this part, we trained a linear regression model on structured, numeric data. The dataset included features like: Number of bedrooms, distance to city center, metro distance, room type etc.

We aimed to train the model so that it could predict the price for new apartments added to the airbnb listing. Therefore we chose to not include factors like cleanliness rating and guest satisfaction. The model learned a mathematical relationship between these features and the actual nightly price (realSum). Once trained, it could predict relatively accurate prices based on new data. This was the ideal method to use, since high accuracy and repeatability are important factors for us.

1.1.1 Model 1

Initially we thought it would be a good idea to have a single model that took data from all cities across Europe at once. For all tests we used the same values to keep consistency and a known target price.

First we wanted to test the predicted price against an actual price in the dataset. The values used to predict a price comes from the first entry in the dataset. The actual price of the first entry in the dataset is $194.03. Model 1 suggested a price of $151.88 leaving us $42.15 off the mark.

Another expected result would be the price being different from city to city. London is a very expensive city but the price is nearly identical regardless of the chosen city.

This led us to do some more testing and we decided on another approach to the model. Instead of having a single model we would have 1 model for each city and compare the results to our first model.

1.1.2 Model 2

To address the problem with similar pricings independently of the city we decided to make a model for each city.

The results from this model were a big improvement which makes sense, as a model is now only trained on data that belongs to its own city. Running the same values used in model 1 in our new model 2 yielded a better result both in terms of accuracy on the first dataset entry (only $19 off the actual price) and gave a noticeably different price depending on the city.

1.2 Part 2: AI

In the second part of the project, we approached the same task of estimating Airbnb prices. This time, instead of training a traditional supervised model, we leveraged LLaMA 3, a large language model running locally through Ollama, combined with Retrieval-Augmented Generation (RAG).

Since LLaMA 3 cannot process raw '.csv' files directly via the API, we first transformed each Airbnb listing into a descriptive, human-readable sentence. For example:

> *"A 2-bedroom apartment in Paris with room for 4 people, 3.5 km from the city center and 1.2 km from the metro. Nightly price: $150."*

The RAG + LLaMA 3 approach differs significantly from traditional machine learning because it does not involve training a predictive model. Instead, it combines retrieval and language model inference to generate intelligent responses based on previously seen data. One of its major strengths is its flexibility and intuitiveness. Users can simply interact with the system through natural language prompts, making it accessible even for those without a technical background.

1.2.1 The results using llama3

Before implementing RAG, we wanted to see how well the model could handle the task without feeding it any context, such as the data set. The query is sent to the llama API and we got some varying responses. The suggested price was usually in the range 120-170$ with some big outliers every so often.

After implementing RAG, the results were less accurate than expected. We tested with different queries with and without the use of 4 T's.
The 4 T's refer to:

- **Task** - What the system is trying to accomplish.
- **Target** - Who is the audience for this text.
- **Tone** - What style of language does the LLM use.
- **Trait** - The personality or style you want the LLM to use.

Our expectation was that by adding the 4 T's in our queries, it would improve the price suggestions. However, what we learned using the llama3 model was that the results were really inconsistent. The result of the exact same query could vary by more than 100% when we ran it. The changing results most likely stems from the non deterministic nature of language models and lack of grounding in structured numeric patterns.

Also on average we got the best results closest to the expected realSum using a query without focus on the 4 T's. On average with that model we got 122$ which was still about 70$ off the target price.

1.3 Conclusion: Comparing the Two Approaches

In summary, the traditional linear regression model provided more accurate, consistent, and reproducible price predictions, particularly when trained on city specific data. This made it ideal for our goal of estimating Airbnb prices with a high degree of precision and reliability. It also allowed us to clearly understand the relationship between input features and the resulting predictions, making the model explainable and trustworthy for structured decision-making.

In contrast, the RAG + LLaMA 3 approach offered a more flexible, interactive, and user-friendly experience, but it struggled to deliver consistent and accurate numeric outputs. While it excelled at interpreting natural language and generating contextual answers, its lack of grounding in structured numerical logic made it unsuitable for tasks that require high precision.
Additionally, its non-deterministic behavior, producing different outputs for the same query, further reduced its reliability for this specific use case.

Overall, the linear regression approach was the clear winner in price prediction. The local AI method was less precise but offered a more flexible and human centric interface that could complement but not fully replace traditional machine learning models.

2. Unsupervised Learning with KMeans Clustering

In this part of the project, we trained a KMeans clustering model on the Airbnb dataset to uncover natural groupings among the listings without relying on labeled outcomes. The goal was to identify clusters that reflect meaningful patterns in the data, such as groups of listings that are budget-friendly, centrally located, or suited for larger groups.

By analyzing key features like price, distance to the city center, number of bedrooms, and proximity to public transportation, the clustering process allowed us to explore the hidden structure of the dataset. This helped us understand how different listings relate to each other and provided a foundation for recommending similar options to users based on their preferences.

2.1 Model Training

The KMeans algorithm grouped listings by minimizing variance within clusters, effectively placing similar apartments together. Once trained, we could assign new listings to a cluster and use the collective characteristics of that cluster to inform recommendations or insights.

The main steps in training the KMeans clustering model were:

- Preparing the dataset by selecting key features such as price, number of bedrooms, distance to city center, distance to metro, room type, and guest capacity
- Standardizing feature values to ensure variables on different scales contributed fairly
- Choosing the number of clusters (k) carefully, using the elbow method to balance simplicity and explanatory power
- Running KMeans to minimize within-cluster variance and generate group assignments
- Evaluating the resulting clusters by examining feature distributions and interpreting distinct patterns

This training process uncovered meaningful groupings, laying the groundwork for both ML-driven recommendations and AI-supported characterizations.

2.2 Cluster Characterization Approaches

2.2.1 Traditional Data Analysis

Once the clusters were established, we analyzed them using descriptive statistics, summary tables, scatter plots, and bar charts. These tools helped identify defining traits for each cluster, such as average price, distance to the city center, and typical apartment sizes.

Z-score normalized cluster means by feature group (scaled to [0,1] for visualization)

	Distance	Accommodation	Quality	Price	Superhost
Cluster 0	0.390000	0.970000	0.570000	0.970000	0.400000
Cluster 1	0.430000	0.400000	0.000000	0.480000	0.330000
Cluster 2	0.360000	0.530000	0.670000	0.420000	0.980000
Cluster 3	0.430000	0.280000	0.590000	0.320000	0.480000
Cluster 4	0.380000	0.390000	0.570000	0.510000	0.320000
Cluster 5	1.000000	0.410000	0.590000	0.300000	0.470000

These numeric and visual analyses provided transparent, reproducible insights that were grounded directly in the data and could be precisely traced back to the underlying features.

2.2.2 AI-based Characterization using LLaMA 3

In parallel, we used the LLaMA 3 model integrated through the Ollama API to generate natural language characterizations of the clusters. We provided the AI with the clustered_airbnb.csv dataset and prompted it to describe the characteristics of each group.

The AI-generated summaries offered human-readable descriptions that highlighted patterns, relationships, and contextual details within each cluster. These natural language outputs sometimes revealed nuances or interpretive connections less obvious in the numerical summaries, adding an extra layer of insight.

2.3 Search Engine Approaches

In addition to cluster characterization, we also developed a Search Engine to help users find the best Airbnb listing match based on their preferences.

We implemented two parallel approaches:

- A machine learning–driven approach that applied structured filtering and selection logic to return the listing that best fit the user's provided parameters (such as budget, number of bedrooms, and location)

- An AI-enhanced approach using LLaMA 3, where the system retrieved relevant listings and provided not only a match but also a natural language explanation of why each option was recommended.

To improve recommendation quality, the machine learning–driven search calculated a **value score** for each listing. This score combined multiple factors, including price (favoring lower cost), distance to the city center and metro (favoring closer proximity), number of bedrooms (favoring larger accommodations), and guest satisfaction ratings (favoring higher-scoring listings). By normalizing and averaging these features, the system produced a standardized score scaled between 0 and 100, allowing for easy comparison across listings.

2.4 Reflection/Conclusion

This project explored two key components within the unsupervised learning part: a search engine for identifying the best Airbnb matches and a system for characterizing KMeans-generated clusters. In both areas, we implemented two complementary approaches: a machine learning–driven method focused on precise, quantitative outputs, and an AI-supported method using LLaMA 3 that emphasized flexibility and natural language explanations.

In the search engine, the machine learning approach consistently retrieved the top-ranked listing based on a calculated value score, offering optimized, data-driven recommendations. The AI-enhanced approach, while sometimes selecting listings with slightly lower scores, still found

high-quality options and added value by providing human-readable explanations that made recommendations easier to understand and more engaging.

For cluster characterization, the traditional analysis produced clear, reproducible summaries using descriptive statistics and visual tools, giving us precise profiles of each cluster's features. The AI-based characterization, though aiming to provide adaptive, context-aware summaries, often repeated generic terms such as "urban," "cozy," and "city," which added little value given the dataset's focus on capital cities. The term "cozy" was frequently applied even to listings with only above-average satisfaction scores, limiting the AI's descriptive depth. However, as with the AI-enhanced search engine, its strength lay in producing human-readable explanations, making cluster outputs more accessible despite limitations in descriptive precision.

Overall, combining these approaches showed how structured machine learning and modern AI can complement each other within the unsupervised learning context. While the machine learning methods provided precise, data-driven outputs and reliable optimization, the AI-supported methods added interpretability and user-friendly communication through natural language.