Project Report

INNOPOLIS UNIVERSITY

Solving Cold Start Problem For New Booking Aggregator

Students	Dmitry Beresnev Vsevolod Klyushev Nikita Yaneev
Course	Big Data Technologies and Analytics
Semester	Spring
Year	2025

Contents

1. Introduction

The online travel accommodation market is a dynamic and fiercely competitive landscape. Launching a new booking aggregator, particularly in a big touristic city like Rio de Janeiro, Brazil, presents formidable challenges. Foremost among these is the ‘cold start problem’: in the absence of historical user data — such as bookings, reviews, or behavioral patterns — the platform struggles to meaningfully rank listings or offer relevant recommendations. This initial lack of personalization can lead to a frustrating user experience, discouraging engagement and critically hampering the platform’s early growth and market penetration.

To effectively address this critical issue, we propose an innovative, data-driven solution. This project leverages publicly available Airbnb data for the Rio de Janeiro region to build a predictive model. Our core objective is to accurately estimate review scores for our new, unrated listings based purely on their intrinsic features — location, price, amenities, host attributes and others. These predicted scores will then serve as a foundational element for platform’s initial ranking algorithms, enabling business to surface high-quality accommodations, enhance discoverability, and immediately improve the user journey.

This initiative directly aligns our technical capabilities with key business objectives: effectively solve the cold start problem, increase booking conversion rates from launch, and accelerate revenue generation in the early stages. In the following sections of this report, we will delve into a specific business context, examine the data structure, and take a detailed look at our methodology for developing a reliable forecasting system designed to function successfully even under the constraints of launching a new platform.

2. Business Understanding

The primary business problem for our newly launched rental aggregator in Rio de Janeiro is the **cold start** challenge: with virtually no historical user interactions (reviews, bookings, or engagement metrics) on our platform, we cannot accurately rank or recommend listings, leading to poor user experience and low conversion rates. By leveraging publicly available data from established aggregators in the same market — specifically, Airbnb listings in Rio de Janeiro — we aim to train a predictive model that estimates guest review scores based on listing attributes, host characteristics, price, location, and amenities. Improving the quality of the ranking through predicted scores should drive higher click-through and booking rates, thereby increasing overall transactions and revenue. This approach aligns the technical data mining objective of **rating prediction** with the business objective of **cold start mitigation** and revenue growth [1, 2].

2.1 Current situation assessment

Today, our platform holds fewer than 100 listings with no guest reviews or booking history. In contrast, incumbent aggregators such as Airbnb and Booking.com maintain thousands of listings in Rio de Janeiro, complete with rich historical data and user feedback. Without any reliable signals, new users see an essentially random order of offers, which leads to dissatisfaction and drop-off. We have access to a CSV export of Airbnb listings (2018–2020) from Kaggle, and a small in-house team poised to ingest, process, and model this information.

2.1.1 Inventory of resources

- **Human Resources:**
  - Data scientists (1 ML engineers, 1 Data Analyst)
  - Data engineers (1 ETL specialist)
- **Data:**
  - Kaggle dataset ‘Airbnb Price Prediction in Rio de Janeiro’ [2]
- **Computing resources:**
  - University Cluster (16 vCPUs, 32 GB RAM)
- **Software:**
  - Python 3.11, PySpark, psycopg2, Beeline
  - Hadoop, Sqoop
  - Snappy, Parquet
  - PostgreSQL, Hive, Superset
  - JupyterLab for exploration

    - GitHub for version control

2.1.2 Assumptions and constraints

- **Assumptions:**
    - Publicly scraped Airbnb data reflects general guest preferences.
    - Host behaviour and listing attributes on Airbnb are analogous to our platform.
- **Constraints:**
    - Limited compute budget—no GPU clusters for now.
    - Dataset size should be at least 200 MB and 300,000 rows.
    - Dataset should include geolocational features (latitude, longitude)
    - Only CSV extracts; no live API access to external platforms.

2.1.3 Risks and contingencies

Risk: Data quality issues (missing values, formatting inconsistencies). *Contingency:* Implement robust ETL and imputation strategies; drop only if absolutely necessary.

Risk: Regulatory changes affecting data usage (GDPR). *Contingency:* Engage legal team early; anonymize personal identifiers.

2.1.4 Terminology

Cold start: The situation where no historical user-item interaction data exists [1].

Predictive model: An algorithm that estimates an outcome variable based on input features.

Amenities: Listing features (e.g., Wi-Fi, air conditioning).

RMSE / MAE: Root Mean Squared Error / Mean Absolute Error, metrics for regression quality.

2.1.5 Business Objectives

1. **Primary objective:** Mitigate cold start by accurately predicting review scores for new listings, thereby improving ranking relevance and increasing bookings.
2. **Related questions:**
    - How much do specific amenities influence predicted scores?
    - Does geographic location drive the majority of score variance?

2.2 Data mining objectives

- **Business goal:** Develop model for listing's review scores predictions.

- **Data mining goal:** Develop a regression model to predict a listing's review score (1–100), using host, listing, price, location, and amenities features.

2.2.1 Business success criteria

- Develop a model that predicts the rating score well enough.
- Product team reports 'useful insights' on feature importances in quarterly review.

2.2.2 Data mining success criteria

- Model achieves RMSE ≤ 10 on a hold-out set of Airbnb data.
- Feature importance analysis is reproducible and interpretable by non-technical stakeholders.

3. Data Understanding

This stage of the CRISP-DM process focuses on initial data analysis to understand the structure, content, and quality of the data provided for the project. This involves collecting, describing, exploring, and assessing the quality of the data in order to ensure it is suitable for modeling.

3.1 Initial data collection

For our analysis, we utilized historical data from an existing rental and hotel aggregator platform. The dataset spans several years and covers multiple seasonal periods. It includes both information about rental listings and their respective hosts.

The main data sources was CSV file with rental offer information exported from the aggregator (covering three years (2018–2020)). The dataset is available on Kaggle [2], which originally compiled publicly available information from the Airbnb website for Rio de Janeiro.

3.2 Data Description

Our data contains about 784K records and 108 different attributes. The entire dataset weighs about 2.5 GB. Among the features we have:

- **geodata** — housing position in coordinates
- **rental housing attributes** — attributes that housing has (e.g., Wi-Fi, TV, number of beds, etc.)
- **host characteristics** — host-related information
- **price** — the price of housing
- **review score rating** — historical rating on the aggregator platform

It is the review score rating that we want to learn how to predict in order to solve the cold start problem.

Overall, the structure and coverage of the data arc aligned with the project's goals, though some preprocessing and cleaning steps are necessary.

3.3 Data Exploration

During this stage, we explored key aspects of the dataset through HQL queries and visual analysis. The goal was to identify trends, anomalies, and relationships that could guide future modeling efforts and transformations. Below we present the insights gained from our initial data mining queries.

1. Price and Rating Distribution in Popular Neighborhoods

To understand pricing and quality dynamics across the most active regions, we selected the top 10 most frequently occurring neighborhoods with non-null entries. For these, we extracted corresponding price and review score ratings (Figures 1 and 2).

Key insights

- Weak Correlation inside neighborhoods: High prices in neighborhoods do not guarantee top reviews in it. Leblon, as most expensive, ranks 7th in mean scores, while Flamengo which is cheaper, scores 4th highest in ratings.
- High-Value Opportunities: Flamengo and Laranjeiras combine low prices with top-tier scores, so can be considered as best neighborhoods to budget-conscious travelers.
- Price-Score Variability: large gaps between mean and median in prices suggest existence of small number of 'luxury' offers, while gaps in scores — that there are small number of offers with small ratings

2. Relationship Between Night Limits, Pricing Buckets, and Ratings

This section explores how booking restrictions (minimum/maximum nights), pricing tiers, and cancellation policies relate to review scores. Prices were bucketed into five meaningful intervals for better interpretability (Figures 3 and 4).

Key insights

- **Higher Prices, Generally Higher Scores (Across Policies):** For most cancellation policies, the most expensive listings (2000+ dollars) tend to have the highest review scores. This suggests that price often correlates with quality/amenities that drive satisfaction, somewhat independently of the cancellation policy itself.
- **Super Strict Policies loose:** `super_strict_30` and `super_strict_60` tend to have scores lower than `moderate` or `flexible`, particularly for budget listings. This is a key finding — a very strict policy on a mid-priced item might be perceived negatively.
- **Moderate Policies Perform Well:** `moderate` and `strict_14_with_grace_period` policies, particularly for listings priced 1000–1999$ and 2000$+, show very high average scores (often above 95.5%–96%). `Flexible` also performs strongly across price bands.
- **Price Correlates with a Stay Length:** there is a clear trend: as the price of the listing increases, both the typical minimum and maximum number of booking nights also tend to increase.
- **Review Score with a Stay Length:** another clear trend is that as the stay length of the listing increases, mean review score also increases. Also, there is a noticeable 'phase transition' between 100–499$ and 500–999$ clusters: mean review score has here drastic increase.

3. Host Verification Features and Their Impact

Here we examine how different host verification combinations affect guest ratings and pricing. Hosts were grouped into categories based on features like profile picture, identity verification, and superhost status (Figure 5).

Key insights

- Superhosts alone achieve the highest review scores, even outperforming fully-featured hosts, suggesting that hosting experience across multiple listings correlates strongly with guest ratings.
- Adding verification on top of a profile picture (picture_verified) slightly reduces the average score, which may indicate diminishing returns or that verification alone is not a strong trust signal.
- Listings with no host features have significantly lower scores, highlighting the risk of promoting anonymous or minimally detailed hosts.
- Interestingly, fully_featured hosts (with all three attributes) do not significantly outperform hosts with just a picture or superhost status

4. Popular Property and Room Type Analysis Over Time (Query 4)

This query focuses on the top 10 property types and their associated room types. It tracks the price and review score ratings across months for combinations that are either:

- Among the most common property types
- Among the most common property-room type pairs

Additionally, we construct a descriptive label combining room type and property type for intuitive grouping (Figures 6 to 9).

Key insights

- **Property Type Matters for Scores:** for example,‘Guest suite’,‘Loft’ and‘Townhouse’ consistently achieve the highest review scores.
- Room Type is a Major Driver of Scores: while‘Private room’ consistently receives the highest scores, the‘Shared room’ and‘Hotel room’ receive significantly lower review scores.
- **Winning Combinations for High Scores:** combinations‘Private room in Condominium’ and‘Private room in Apartment’ consistently yields exceptionally higher scores.
- **Price and Score Relationship:** the most expensive popular options are typically‘Entire home/apt’ in House and in Apartment. While they score well, they don’t consistently outperform the‘Private room in Condominium/Apartment’ in terms of guest scores. This suggests that while guests are willing to pay more for entire spaces, the quality in well-managed private rooms within desirable property types can lead to even higher scores.

- **Seasonality Impact on Prices:** prices for popular combinations show slight seasonal fluctuations. Prices during local fall and winter are slightly higher.

5. Host Responsiveness and its Impact on Guest Experience

This analysis investigates how host responsiveness (response time and response rate) relates to guest satisfaction and pricing. Buckets for both price and response rate allow easier pattern recognition (Figures 10 and 11).

Key insights

- **Faster is Generally Better:** Across most price categories (except premium 2000$+ segment), faster host response times correlate with higher review scores. Scores tend to decrease as response time lengthens to 'within a day' and further to 'a few days or more'.
- **Unusual Premium Pattern for Response Time:** The 2000$ category shows a slightly unusual pattern: the 'a few days or more' category surprisingly has the highest average score. This could be due to a smaller sample size in this specific high-end segment for slow responders, or perhaps the hosts of this segment just do not communicate with the most of customers, preferring small number of elites.
- **Higher Response Rate Mean Higher Scores:** This shows a quite strong and consistent positive correlation. The more consistently a host responds, the higher their review scores, almost regardless of the listing's price.

6. Geospatial Distribution of High-Value and Highly-Rated Neighborhoods

In this query, we geolocate listings and mark those belonging to:

- **Concentration of High Scores and Prices:** Both the 'Mean score map' and 'Mean price map' show clear concentrations of higher values in specific areas, predominantly along the southern coastal strip of Rio de Janeiro (likely encompassing famous touristic areas). There's a strong visual similarity in the distribution patterns of high scores and high prices.
- **Overlap of Top Price and Top Score Neighborhoods:** the overlap exists but is not strong. Most overlapping occurs in central regions of Rio de Janeiro, where both review scores and prices are high. The south-west part of the city is high-priced and (relatively) low-scoring, whilenorth past is, vice versa, low-priced but very high-scoring. So the key finding is that central neighborhoods 'hidden-gems', so should have higher rating.

  This is useful for mapping 'premium' vs 'well-rated' areas (Figures 12 to 15).

Key insights

- Spatial clustering of high-value vs high-rating areas can inform geographic segmentation.
- Opportunity to identify luxury vs customer-favorite locations.

7. Amenity Combinations and Their Influence on Price and Ratings

This final analysis categorizes listings based on presence of key amenities: Wi-Fi, air conditioning, hot water, and refrigerator. We assess how different combinations affect price and satisfaction (Figure 16).

Key insights

- **WiFi is Critical:** Listings with WiFi, especially in combination with other key amenities like AC and a fridge, dominate the high scores. Even wifi_only generally outperforms listings with only AC, only a fridge, or only hot water.
- **Air Conditioning (AC) is Highly Valued:** Particularly when paired with WiFi. Given climate of Rio de Janeiro, this is unsurprising.
- **Hot Water is an Expectation:** Its presence in 'all_amenities' contributes to high scores, but hotwater_only scores poorly. This suggests it's a baseline necessity rather than a feature that elevates a listing if other desirable amenities are absent.
- **Seasonal Impact:** There are minor monthly fluctuations for most combinations, but the overall ranking of amenity packages by score remains largely the same throughout the year. For example, the need for AC might be perceived as slightly less critical in cooler months, but listings with wifi_ac still outperform those with just wifi_only.

3.4 Summary of Data Exploration

Further transformations and detailed exploration will build on these findings to enhance data quality and modeling performance.

3.5 Data quality

An assessment of data quality revealed several issues:

- All rows in our dataset have one or more null value, which is why we limit ourselves to certain features subset, that would be reviled in the next section.
- Several columns have problematic format. For example, column 'amenities' is presented in format {attr1, attr2, ...}, which requires additional efforts for formatting.
- Almost half of our dataset misses our target feature 'review_scores_rating' (385K out of 784K).

These issues will be addressed through data cleaning steps during the Data Preparation phase. Despite these inconsistencies, the dataset is overall of acceptable quality for modeling purposes.

4. Data Preparation

This phase of the CRISP-DM process is focused on refining and shaping the data to ensure it is suitable for analysis. At this point, we identify which parts of the dataset will be utilized, taking into account factors such as relevance to the analytical goals, data quality, and potential technical limitations including data size and format. Data preparation plays a critical role in the success of the overall modeling process, and decisions made here directly influence the accuracy and robustness of the final results.

4.1 Data selection

The original dataset consists of 108 columns, which contain a wide range of information. However, not all features are equally relevant or useful for our specific task — predicting the rating of rental listings. Therefore, we conducted a feature selection process and chose a subset of variables that are most informative for the prediction task, based on both domain knowledge and preliminary exploration.

The selected features are as follows:

1. `host_since` — the date when the host started renting properties on the platform
2. `host_response_time` — average response speed of the host to inquiries
3. `host_response_rate` — the percentage of inquiries the host responded to
4. `host_is_superhost` — indicates whether the host has superhost status
5. `host_has_profile_pic` — whether the host has uploaded a profile picture
6. `host_identity_verified` — whether the host has verified their identity
7. `neighbourhood` — the district in which the property is located
8. `latitude` — latitude for rent apartment
9. `longitude` — longitude for rent apartment
10. `property_type` — the general type of the rental property
11. `room_type` — classification of the room offered for rent
12. `accommodates` — how many guests the listing can accommodate
13. `bathrooms` — number of bathrooms in rent apartment
14. `bedrooms` — number of bedrooms in rent apartment
15. `beds` — number of beds in rent apartment
16. `bed_type` — type of beds provided
17. `amenities` — list of additional features and services offered
18. `price` — nightly rental price
19. `security_deposit` — the deposit required by the host

20. cleaning_fee — cleaning fee charged by the host
21. guests_included — number of guests included in the base price
22. extra_people — additional cost per extra guest
23. minimum_nights — maximum rental duration
24. maximum_nights — minimum rental duration
25. review_scores_rating — aggregated review rating (our target variable)
26. instant_bookable — whether the property can be instantly booked
27. cancellation_policy — cancellation terms specified by the host
28. require_guest_profile_picture — whether the host requires guests to have a profile picture
29. month — the month in which the data was recorded (used for seasonal adjustment)

These features were selected based on their potential impact on the guest experience and their assumed correlation with the listing's rating. The selected subset balances granularity and generality while reducing noise from redundant or irrelevant columns.

4.2 Data cleaning

To preserve as much valuable data as possible, we applied targeted imputation strategies for handling missing values. Our approach included the following treatments:

- host_response_time — replaced with 'unknown'
- host_response_rate — set to 0
- security_deposit, cleaning_fee — imputed with 0
- amenities — missing values replaced with an empty string

For columns where missing values were less frequent and could significantly affect data integrity (e.g., bedrooms, price), we chose to drop those rows entirely. After cleaning, the dataset was reduced to approximately 379,600 records, which still offers a robust sample for modeling.

In addition to imputation, several formatting adjustments were made:

- Monetary columns such as price, security_deposit, cleaning_fee, and extra_people were stripped of currency symbols to enable numerical operations.
- The amenities column was cleaned of extraneous curly braces and quotation marks for easier parsing and analysis.

These steps ensure consistency across records and prepare the data for efficient feature extraction and model input.

4.3 Data construction

This phase focuses on the generation of new attributes derived from existing ones, with the aim of enriching the dataset and improving the performance of downstream models.

Derived attributes The `amenities` field originally contained over 180 unique entries in a single textual column, representing various features of each listing. To extract more actionable information, we analyzed the frequency of occurrence of these amenities and selected the 20 most common ones. Each selected amenity was transformed into a separate binary feature indicating its presence or absence in a given listing.

This transformation enabled the model to better capture the impact of amenities on listing ratings and allowed for more interpretable feature importance in later stages of the project.

4.4 Data compression

As a part of the project, we were required to test different data compression tools and select the most suitable one. The results are presented in Table 1.

Table 1: Comparison of models

Tool	Memory, Mb	Build Time, seconds
parquet	47.2	27.6
avro + snappy	122.4	**23.8**
avro + bzip2	**21.4**	56.7

You can notice that there is only one row for `parquet`, while for `avro` two approaches were tested. The thing is that `parquet` is not compatible with `bzip2`, so only combination `parquet+snappy` was tested.

As a result, the `parquet` was selected as the most suitable option, because it offer the better time/resulting size tradeoff.

5. Modeling

This stage of the project focuses on applying machine learning techniques to the structured dataset in order to develop a predictive system for estimating listing ratings. The aim is to select a model that not only performs well statistically, but is also interpretable and practical for deployment in a real-world recommendation pipeline, particularly under cold-start conditions where historical data is absent.

5.1 Modeling techniques

To explore the modeling space, we trained and evaluated two distinct approaches:

Linear Regression — A classic and widely used statistical model that assumes a linear relationship between the input features and the target variable. It serves as a strong baseline due to its simplicity, speed, and interpretability. Linear Regression attempts to fit a straight line that best describes the relationship between the predictors and the rating score..

Random Forest Regressor — A powerful ensemble method that aggregates multiple decision trees, capturing non-linear relationships and interactions between variables. It is well-suited for structured data and has strong generalization ability.

Preprocessing assumptions

- Categorical variables were transformed using One-Hot Encoding.
- The string `amenities` field was vectorized using a pre-trained `word2vec` model to preserve semantic relationships.
- Geolocation data was converted from latitude/longitude to ECEF coordinates to better encode spatial proximity.

These transformations ensured the models could effectively leverage all available information and handle heterogeneous feature types.

5.2 Generate test design

We split the dataset into an 80% training set and a 20% test set, ensuring a clean separation for unbiased evaluation. This split strategy allowed us to simulate real-world model behavior on unseen listings.

We evaluated models using two standard regression metrics:

- **Root Mean Squared Error (RMSE)** — Emphasizes large errors, useful for detecting problematic predictions.
- **Mean Absolute Error (MAE)** — Measures average prediction error in the original rating scale, making it more interpretable.

5.3 Build model

Model training was optimized using cross-validation and grid search to explore hyperparameters and reduce overfitting. This ensured that the final models were not just fitted to training data but generalized well to new inputs.

Model behavior: Random Forest effectively captured non-linear relationships, especially between location, property characteristics, and host quality. Linear Regression, though limited in expressive power, served as a lightweight and interpretable baseline.

5.4 Assess model

Based on the results of the cross validation, we decided to choose the RF model as the main one. Below is a table of models and their results.

Table 2: Grid Search Results For Random Forest Regressor (RMSE)

	Max Depth 5	**Max Depth 10**
5 Trees	9.12	8.43
10 Trees	9.16	**8.4**

Table 3: Grid Search Results For Random Forest Regressor (MAE)

	Max Depth 5	**Max Depth 10**
5 Trees	5.7	5.35
10 Trees	5.74	5.32

Table 4: Grid Search Results For Linear Regressor (RMSE) Where λ-Elastic Net Parameter and β-Reg Parameter

	λ= **0.0**	λ= **0.5**	λ= **1.0**
β= **0.01**	9.13	**9.0**	9.12
β= **0.1**	9.07	9.05	9.15
β= **1.0**	9.02	9.28	9.32

Table 5: Grid Search Results For Linear Regressor (MAE) Where λ-Elastic Net Parameter and β-Reg Parameter

	λ= **0.0**	λ= **0.5**	λ= **1.0**
β= **0.01**	5.64	5.64	5.66
β= **0.1**	5.62	5.69	5.7
β= **1.0**	5.65	5.85	5.95

6. Evaluation

Evaluation on the test set revealed clear differences in model performance:

- **Random Forest** achieved a lower MAE of approximately 8 points, showing it could reliably estimate listing ratings without prior review data.
- **Linear Regression** underperformed on listings with extreme or high scores, highlighting its limitations in modeling complex patterns.

Interpretability: Despite being more complex, the Random Forest model still allowed us to extract feature importance, which provided actionable insights.

Model ranking: Based on performance and utility, the Random Forest Regressor was chosen as the preferred model. It delivers accurate predictions while maintaining enough interpretability for business analysis and decision-making.

Future directions: Although current resource constraints prevented us from testing more computationally intensive models (like Polynomial Regression or Deep Neural Networks), this foundation offers several promising paths:

- Exploring time-based features (e.g., seasonal patterns)
- Using embedding techniques for high-cardinality categorical variables
- Scaling model training with GPU acceleration

In its current state, the model is both effective and production-ready, offering a scalable solution to the cold start problem while leaving room for future refinement.

Table 6: Comparison of models

Model	RMSE	MAE
Linear Regression	9.06	5.68
Random Forest Regressor	8.28	5.34

7. Deployment

At this stage, we focus on how the developed models can actually be used in a real-world scenario. The goal is to smoothly integrate the best-performing model — Random Forest Regressor — into the ranking system of a rental aggregator platform. The idea is simple: use what we have built to help users see better offers first, especially when no reviews or previous data are available — the so-called cold start situation.

Deployment plan The deployment is planned as follows:

Export the trained model into a serialized format (e.g., using `joblib` or `pickle`) for reuse in a production environment.

Create a lightweight API service (e.g., Flask or FastAPI) that receives a listing's input data and returns a predicted rating.

Integrate this prediction service into the platform's backend, so it can run in real-time or batch mode, depending on the traffic and system load.

Periodically retrain the model when more listings and real user feedback become available, to improve its accuracy over time.

This setup is simple enough to maintain, yet flexible if we decide to scale or swap out the model later.

7.1 Limitations and Challenges

There were a few things we bumped into along the way:

Cold start still means guessing: Even with a good model, predicting without real user reviews will always have some uncertainty.

Limited compute resources: We couldn't try more advanced models (like neural networks) due to hardware constraints.

Sparse and messy data: Some fields (like amenities) were messy, and it took time to clean and transform them into usable features.

Geographical bias: Since we only worked with listings from Rio, the model might not generalize well to other locations without fine-tuning.

8. Contributions and Reflections on own work

The authors of this report and the contributors of the project are presented in Table 7. (**Fill in the contributions table.**)

Stages	Student 1 Dmitry Beresnev	Student 2 Vsevolod Kliushev	Student 3 Nikita Yaneev
Role	ML engineer	Data engineer	Data Analyst
Introduction	40%	30%	30%
Business understanding	30%	40%	30%
Data understanding	20%	60%	20%
Data preparation	20%	50%	30%
Modeling	45%	10%	45%
Evaluation	45%	10%	45%
Deployment	0%	0%	0%
Project Stage I	33%	34%	33%
Project Stage II	34%	33%	33%
Project Stage III	33%	33%	34%
Project Stage IV	33.3%	33.3%	33.3%

Table 7: Contributions table

8.1 Report summary

Problem Formulation and Scoping

We had not reformulated the problem itself during the work. Initially we decided to use only subset of the original dataset. However, we faced with problem with big number of undefined values: after filtering them the dataset becomes too small to satisfy project requirements. So we have decided to use the whole initial dataset, so after filtering undefined values, it size remains acceptable.

Knowledge Search for Business Problem Scoping

The primary approach to scoping the business problem involved identifying a suitable proxy for real-world data. The selection of the Airbnb Rio de Janeiro dataset was crucial, as it provided a rich source of listings with features (price, location, amenities, host details) and historical review scores analogous to what our new aggregator would eventually handle. The CRISP-DM methodology, referenced in the document, guided the structured approach to understanding the business needs and translating them into data mining objectives.

Implementation, Testing, and Validation of Results

Implementation The project uses a data pipeline involving HQL for initial exploration and Python for initial data preparation (cleaning, feature selection, construction of new features like binary amenity indicators from a complex text field). Modeling was performed via PySpark using Linear Regression and Random Forest Regressors, with preprocessing steps including One-Hot Encoding for categorical variables, word2vec for amenities, and ECEF coordinate transformation for geolocation data.

Testing Models were evaluated on an 80/20 train-test split using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as key performance metrics

Validation Cross-validation and grid search were employed during model building to optimize hyperparameters (e.g., tree depth for Random Forest, regularization parameters for Linear Regression) and ensure generalization to unseen data, mitigating overfitting. The Random Forest model ultimately achieved an RMSE of 8.28 and MAE of 5.34.

Helpful Sources

The Kaggle dataset 'Airbnb Price Prediction in Rio de Janeiro' [2] was the foundational resource, providing all the data for analysis and modeling. The CRISP-DM methodology [1] provided a structured framework for the entire data mining process, from business understanding to deployment considerations. The specified software stack (Python, PySpark, Hadoop, Hive, PostgreSQL, etc.) enabled the practical execution of data processing and machine learning tasks.

What We Would Do Differently

Earlier, More Rigorous Data Quality Assessment We have faced with data quality issues, such as every row having nulls and nearly half the dataset missing the target variable (review_scores_rating). A more intensive initial data audit could have led to faster decisions on imputation strategies, necessary data filtering, or even exploring supplementary data sources earlier.

Deeper Model Engeenering Given more resources, we would have explored Deep Neural Networks or more complex ensemble methods, like MLP and SVM

Deeper Feature Engineering for Complex Fields The amenities column, with over 180 unique entries, was complex. While binary indicators for the top 20 were created, exploring more sophisticated NLP techniques or embedding methods for this feature from the outset might have yielded better insights and model performance more quickly.

More Diverse Dataset Maybe it would be good idea to incorporate other datasets to original one (for example, for cities near to Rio de Janeiro).

Other Changes to the Project

- We would propose to add some GPU to cluster

References

[1] Peter Chapman. Crisp-dm 1.0: Step-by-step data mining guide. 2000. URL https://api.semanticscholar.org/CorpusID:59777418.

[2] Hazujaf. Airbnb price prediction in rio de janeiro - python. Kaggle Dataset, 2023. URL https://www.kaggle.com/datasets/hazujaf/airbnb-price-prediction-in-rio-de-janeiro-python. Accessed: 20.05.2025.

A. Appendices

A.1 Plots and Figures

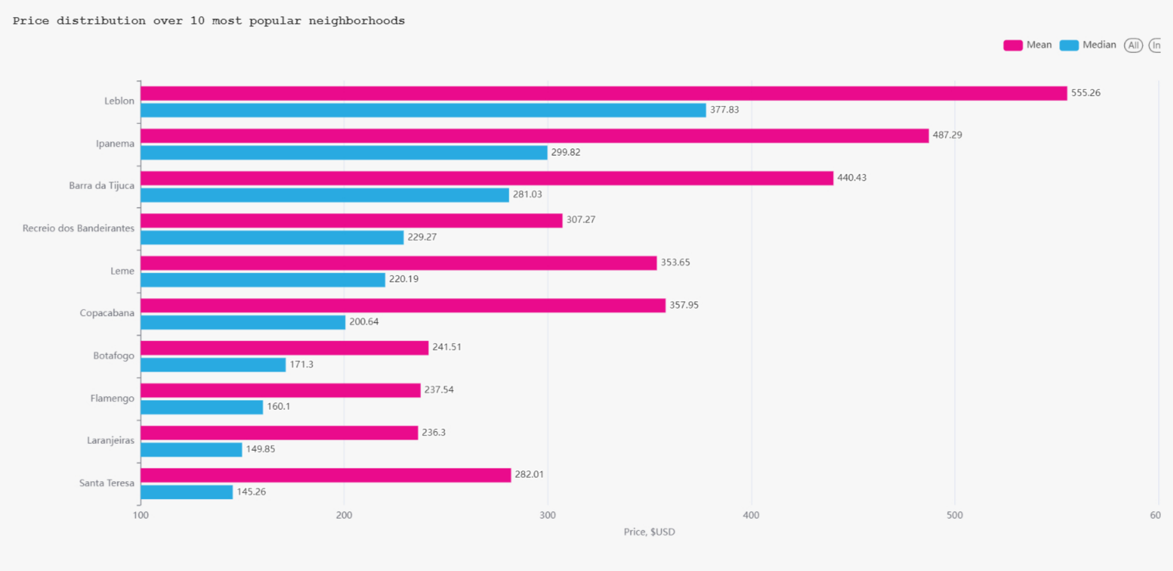


Figure 1: Price distribution over 10 most popular neighborhoods

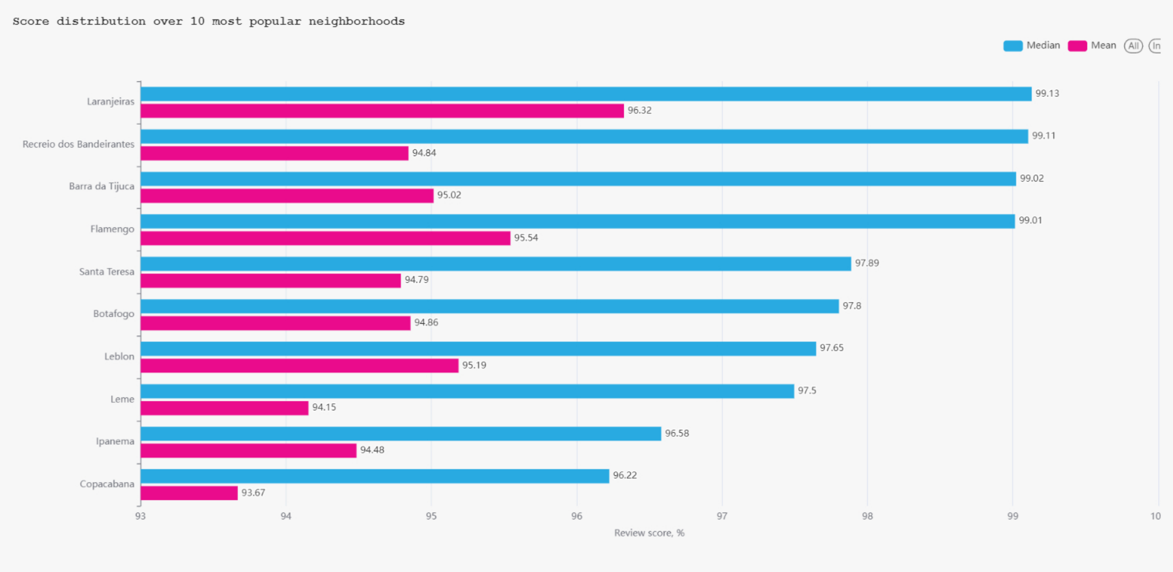


Figure 2: Score distribution over 10 most popular neighborhoods

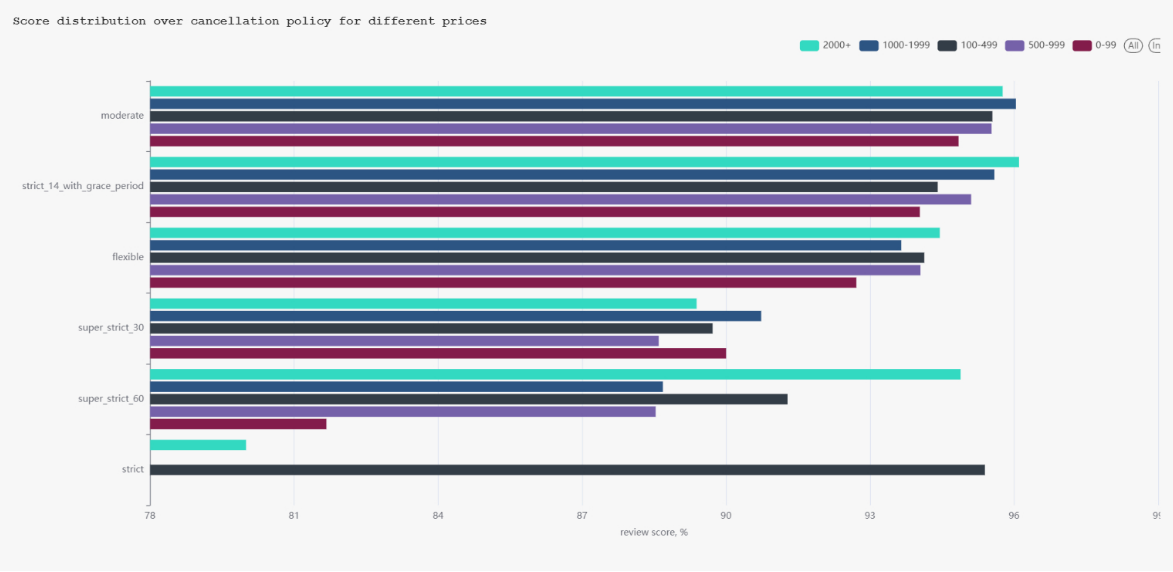


Figure 3: Score distribution over cancellation policy for different prices

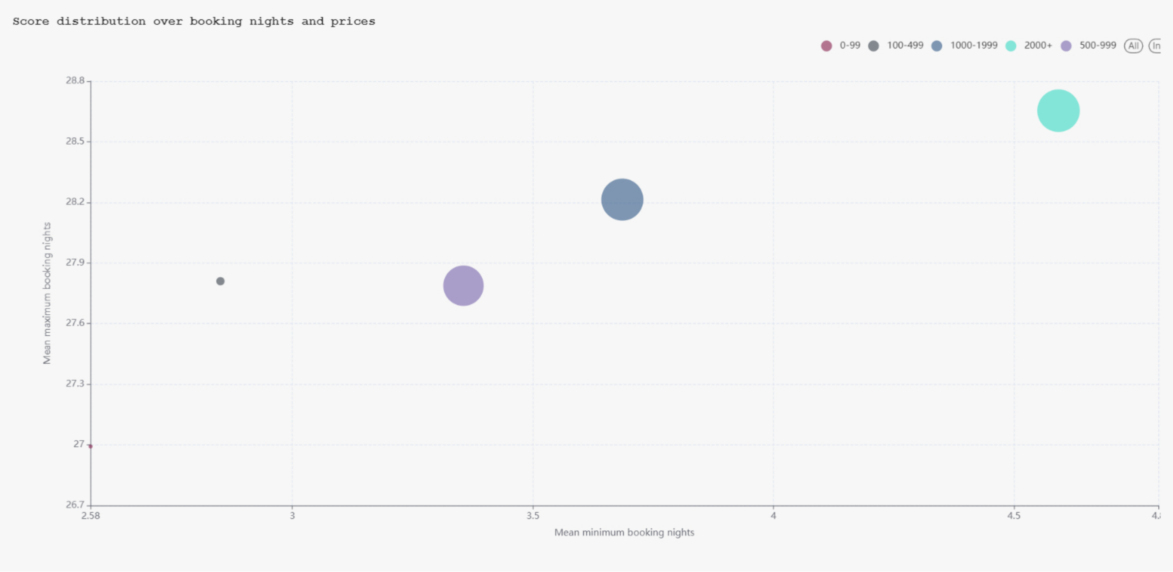


Figure 4: Score distribution over booking nights and prices

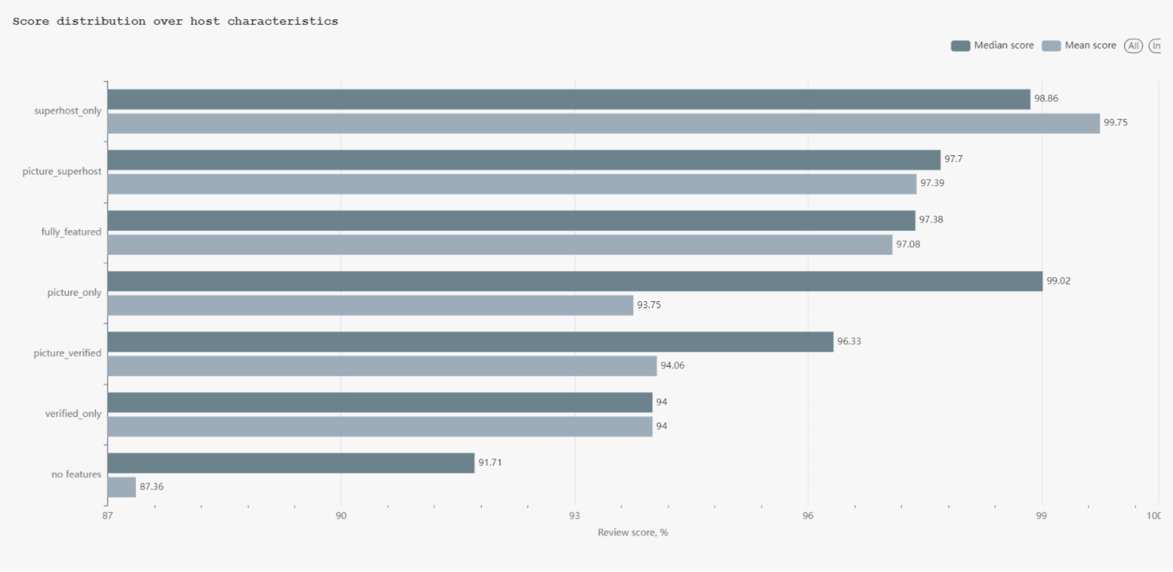
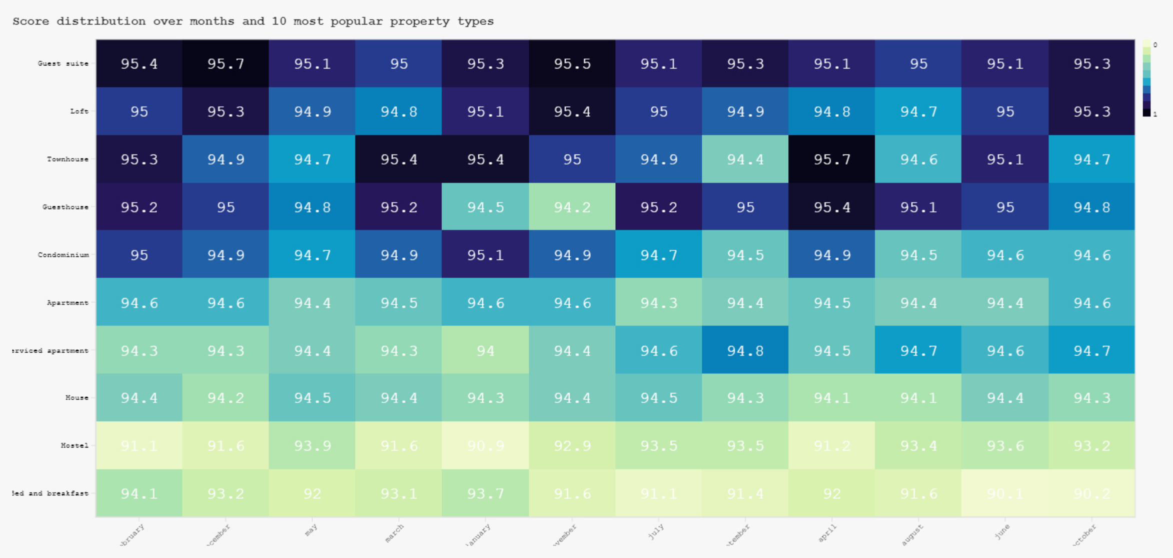


Figure 5: Score distribution over host characteristics



	bruary	cember	may	march	anuary	vember	july	tember	april	august	june	ctober
Guest suite	95.4	95.7	95.1	95	95.3	95.5	95.1	95.3	95.1	95	95.1	95.3
Loft	95	95.3	94.9	94.8	95.1	95.4	95	94.9	94.8	94.7	95	95.3
Townhouse	95.3	94.9	94.7	95.4	95.4	95	94.9	94.4	95.7	94.6	95.1	94.7
Guesthouse	95.2	95	94.8	95.2	94.5	94.2	95.2	95	95.4	95.1	95	94.8
Condominium	95	94.9	94.7	94.9	95.1	94.9	94.7	94.5	94.9	94.5	94.6	94.6
Apartment	94.6	94.6	94.4	94.5	94.6	94.6	94.3	94.4	94.5	94.4	94.4	94.6
erviced apartment	94.3	94.3	94.4	94.3	94	94.4	94.6	94.8	94.5	94.7	94.6	94.7
House	94.4	94.2	94.5	94.4	94.3	94.4	94.5	94.3	94.1	94.1	94.4	94.3
Hostel	91.1	91.6	93.9	91.6	90.9	92.9	93.5	93.5	91.2	93.4	93.6	93.2
ed and breakfast	94.1	93.2	92	93.1	93.7	91.6	91.1	91.4	92	91.6	90.1	90.2

Figure 6: Score distribution over months and 10 most popular property types

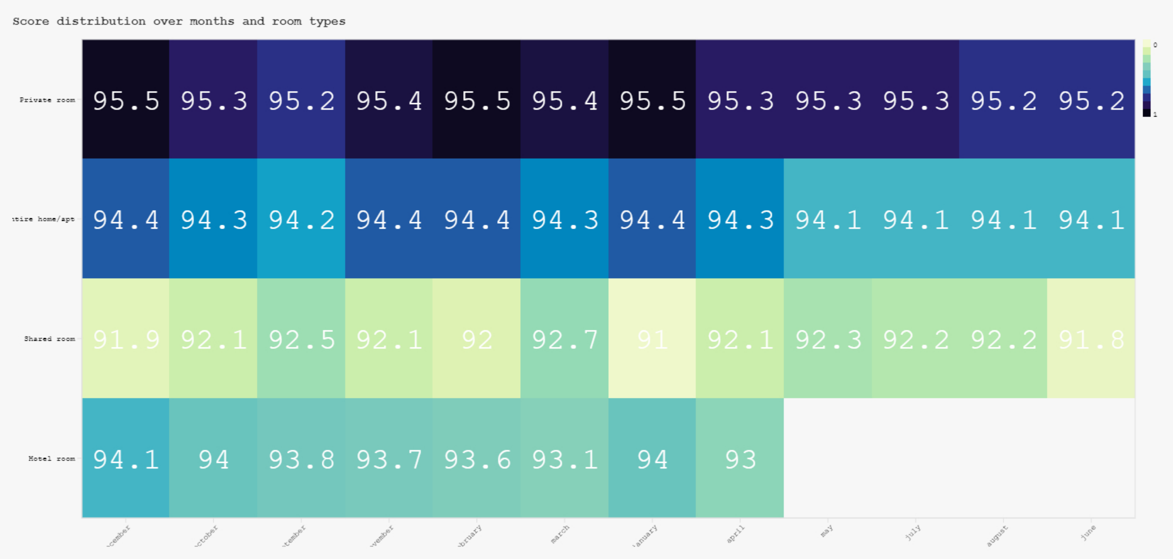


Figure 7: Score distribution over months and room types

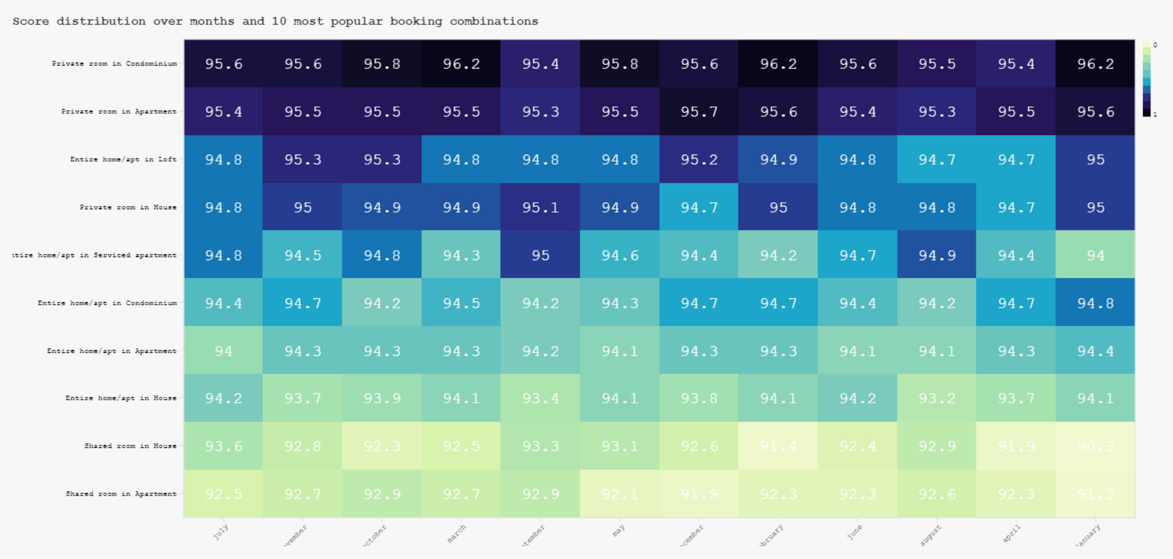
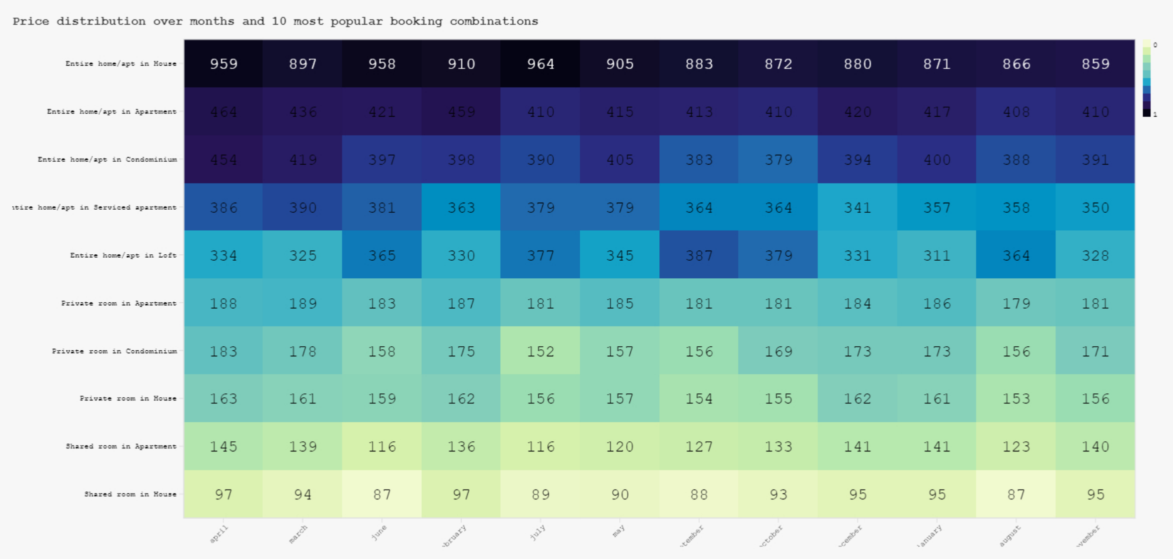


Figure 8: Score distribution over months and 10 most popular booking combinations



	april	march	june	bruary	july	may	tember	ctober	cember	anuary	august	vember
Entire home/apt in House	959	897	958	910	964	905	883	872	880	871	866	859
Entire home/apt in Apartment	464	436	421	459	410	415	413	410	420	417	408	410
Entire home/apt in Condominium	454	419	397	398	390	405	383	379	394	400	388	391
ıtire home/apt in Serviced apartment	386	390	381	363	379	379	364	364	341	357	358	350
Entire home/apt in Loft	334	325	365	330	377	345	387	379	331	311	364	328
Private room in Apartment	188	189	183	187	181	185	181	181	184	186	179	181
Private room in Condominium	183	178	158	175	152	157	156	169	173	173	156	171
Private room in House	163	161	159	162	156	157	154	155	162	161	153	156
Shared room in Apartment	145	139	116	136	116	120	127	133	141	141	123	140
Shared room in House	97	94	87	97	89	90	88	93	95	95	87	95

Figure 9: Price distribution over months and 10 most popular booking combinations

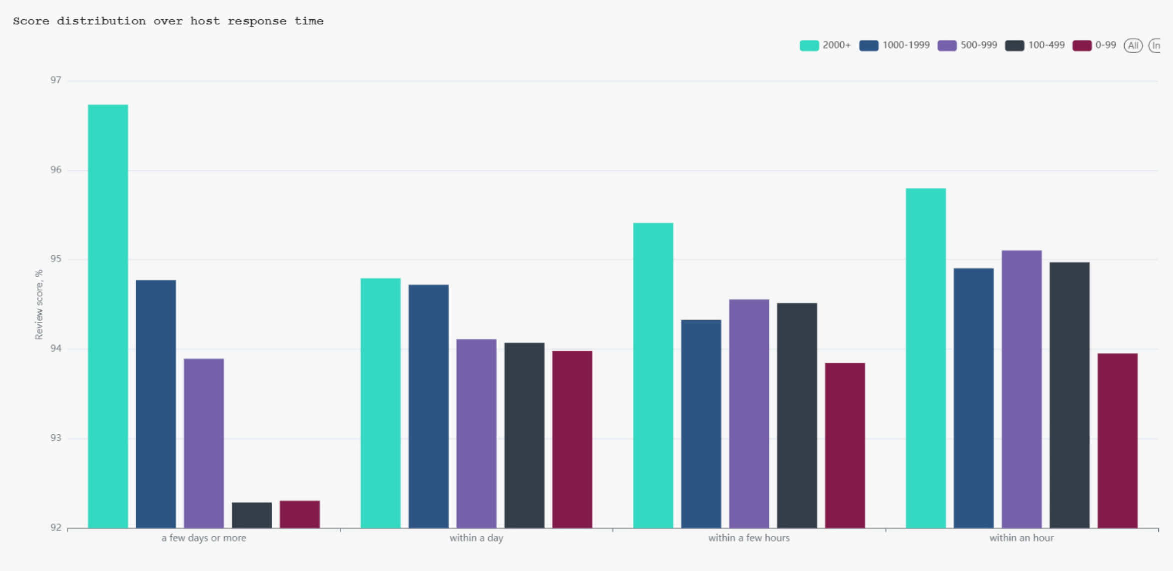


Figure 10: Score distribution over host response time

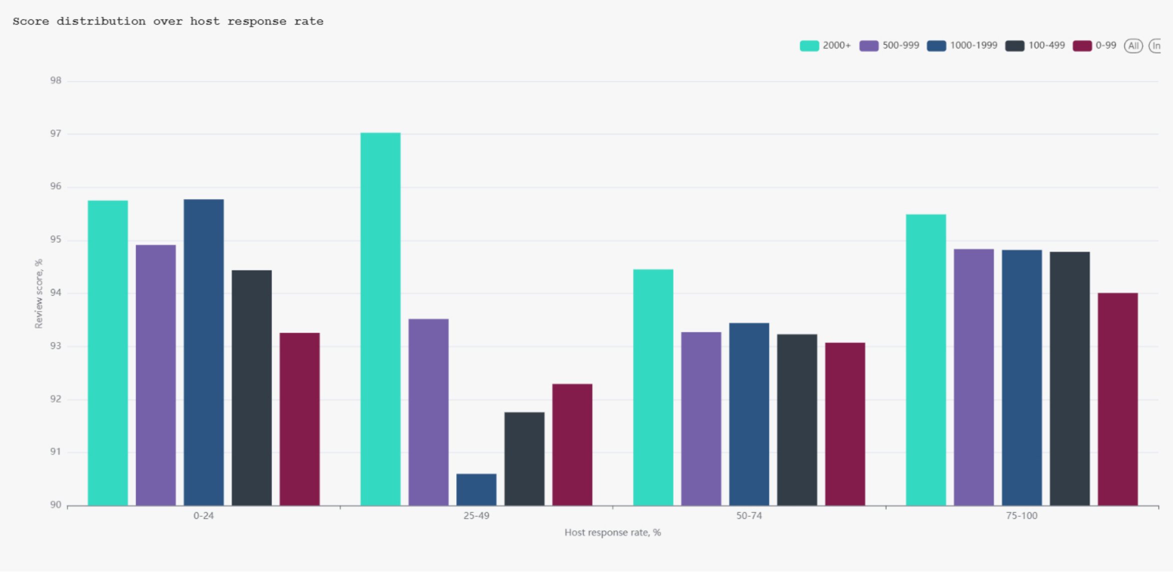


Figure 11: Score distribution over host response rate

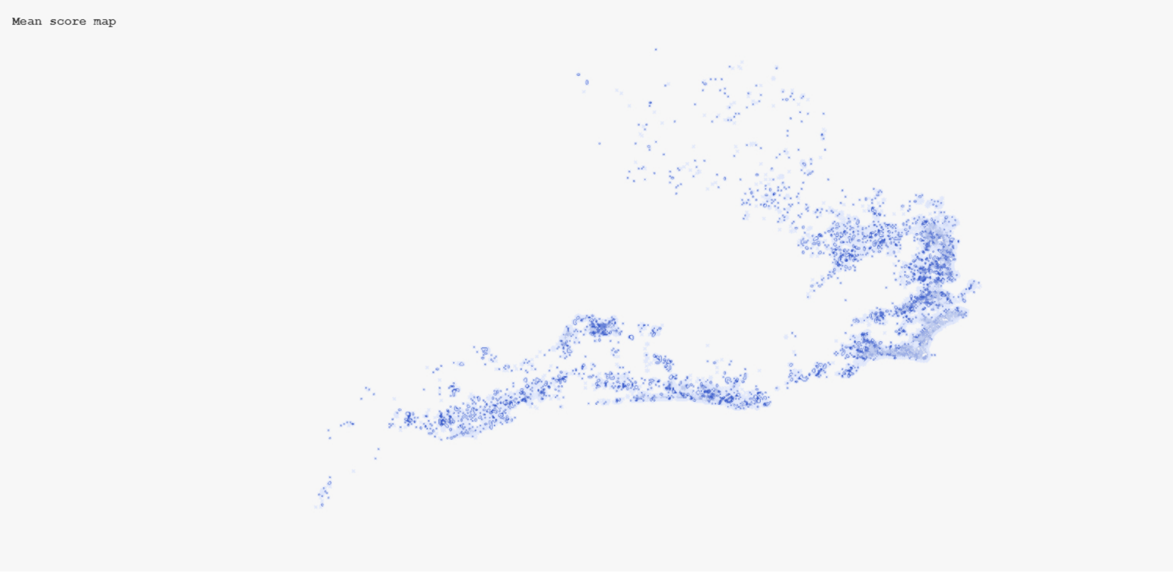


Figure 12: Mean score map

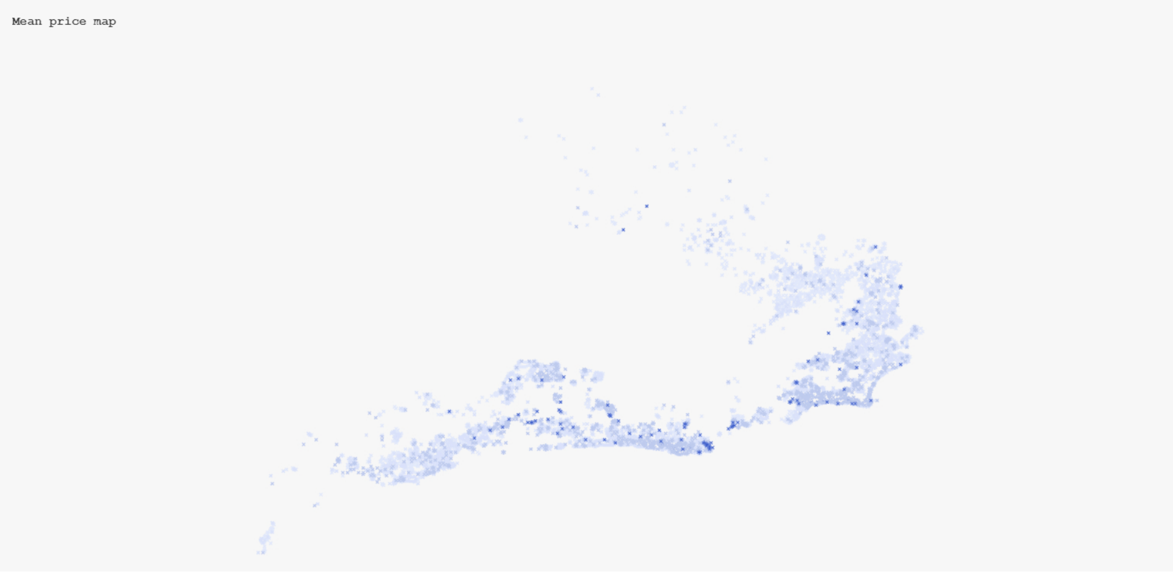


Figure 13: Mean price map

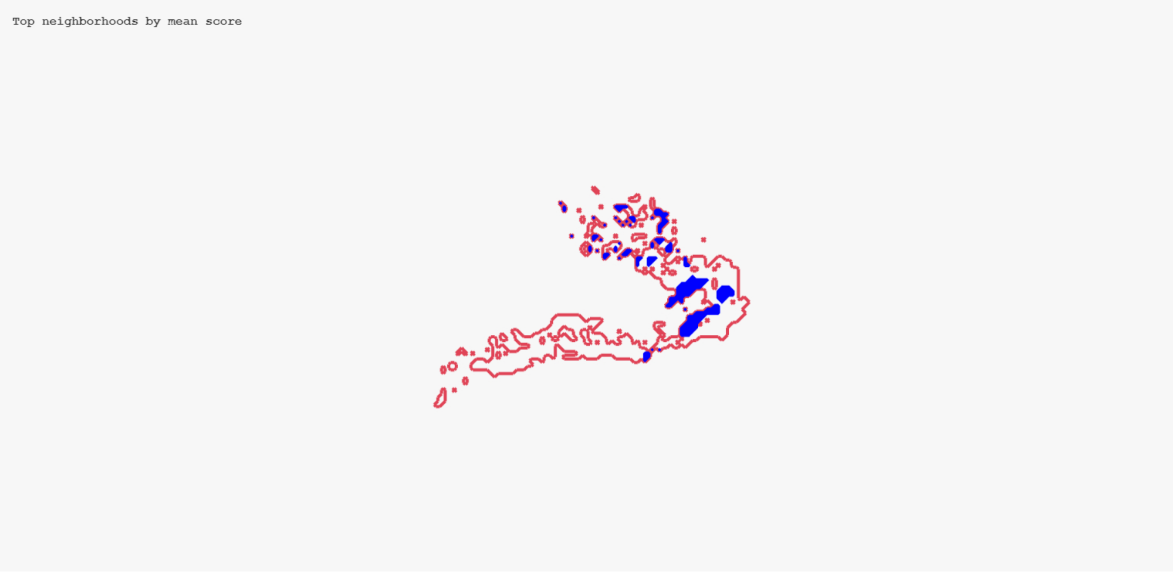


Figure 14: Top neighborhoods by mean price

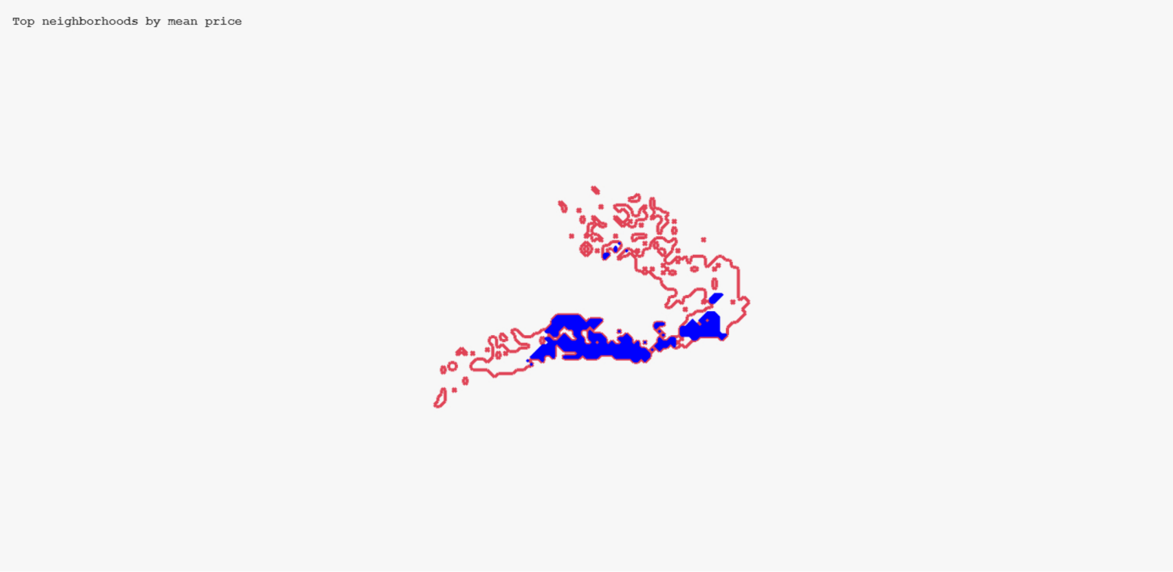


Figure 15: Top neighborhoods by mean score

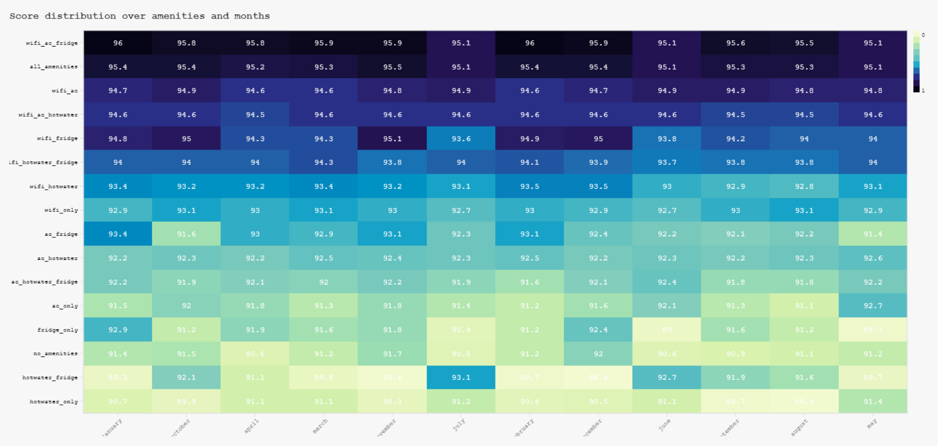


Figure 16: Score distribution over amenities and months