Airbnb Price Prediction Analytics

Table of Contents

1. Airbnb

1.1. Problem understanding and approaches

Problem Understanding:
Pricing Airbnb listings can be a complex task, as it involves several factors such as location, property type, amenities, and guest reviews. Currently, many hosts rely on personal judgment or outdated pricing models, leading to inaccurate pricing that may either deter potential guests or result in lost revenue. Mispricing can negatively impact booking rates, host earnings, and overall market competitiveness.

One of the main challenges in pricing Airbnb listings is the skewed distribution of prices, with a few high-priced listings influencing the average, making it difficult to establish a fair pricing strategy. Additionally, there are complex interactions between various features such as the listing's location, its amenities, guest ratings, and seasonal demand, which further complicates the pricing decision. Without a data-driven approach, hosts are left with subjective price setting that does not align with market trends.

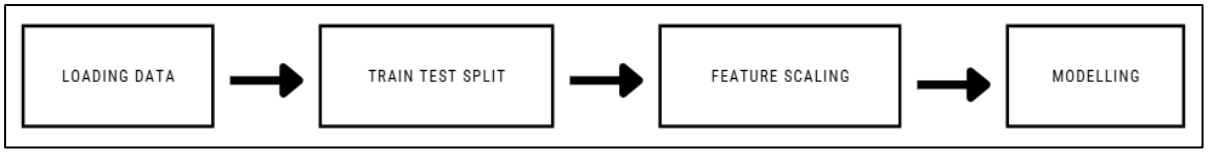
To solve these challenges, we propose developing a machine learning model to predict Airbnb listing prices more accurately. By analysing key features like location, property type, amenities, and review ratings, the model will provide more precise price predictions, helping hosts adjust their pricing to better reflect market demand and competitive pricing.

From a business perspective, this model will allow Airbnb hosts to optimise their pricing strategies, maximise occupancy rates, and improve profitability. By providing more accurate price recommendations, hosts can ensure their listings are priced competitively, attracting more guests while avoiding underpricing or overpricing. This will also lead to higher guest satisfaction and repeat bookings.

From a technical standpoint, the problem requires handling skewed data distributions and incorporating features with complex interactions. The model will utilise algorithms like XGBoost, LightGBM, or CatBoost to account for non-linear relationships between variables and handle outliers. These models are well-suited for this task due to their robustness in predicting continuous values and their ability to manage large, varied datasets. Additionally, feature engineering techniques will be employed to transform raw data into actionable insights, ensuring that the model provides accurate predictions under different market conditions.

By leveraging machine learning to predict Airbnb prices, this solution offers a data-driven approach that enhances pricing decisions, improving both business outcomes and guest experiences.

Approaches:



Overview of Steps Taken Throughout the Project

1.2. Summary of data cleaning and transformation process

My rationale for this approach is that, in Assignment 1, I completed the full data cleaning and transformation process, including handling missing values, encoding categorical variables, creating new features, and applying numerical transformations. Once these steps were completed, I exported the processed dataset for use in Assignment 2, meaning there's no need to repeat these steps. However, scaling (StandardScaler) was not applied before exporting the data, and this step is crucial for models like logistic regression, SVM, and neural networks that require numerical consistency. Therefore, in Assignment 2, I'll apply scaling using the same parameters from Assignment 1 to ensure consistency across both training and testing datasets. This approach is similar to the HR Analytics use case above, which prevents unnecessary repetition of preprocessing, ensures models are trained on properly processed data, and maintains an efficient workflow by only applying the necessary transformation, scaling, in Assignment 2.

1.3. Build the model(s)

Similar to the HR Analytics, I also spent a lot of time researching and evaluating various machine learning models to ensure I selected the most appropriate ones for my specific needs. Given that the target variable in my dataset, price of the Airbnb listings which was extremely left-skewed, I focused on models that could handle skewed distributions effectively. After conducting thorough research, I decided to implement XGBoost, LightGBM, Random Forest Regressor, MLPRegressor, Gradient Boosting Regressor (GBR), Support Vector Machines (SVM) with Regression (SVR), CatBoost, and Bayesian Ridge Regression. Each of these models provides distinct features and benefits that align perfectly with the goals of my project. Below, I explained why I chose these models and how their features contribute to the success of my project.

XGBoost (Extreme Gradient Boosting)

XGBoost is a gradient boosting model that excels at handling non-linear relationships and effectively manages skewed distributions. I chose XGBoost because it can model complex relationships in the data, which is essential when predicting prices based on features like location, amenities, and reviews. The built-in regularisation techniques help prevent overfitting, which is particularly important when working with skewed data. Additionally, XGBoost can weight instances during training, allowing it to handle imbalanced and skewed datasets more effectively.

LightGBM (Light Gradient Boosting Machine)

LightGBM is another gradient boosting model, similar to XGBoost, but optimised for faster training and memory efficiency. This makes it an ideal choice when working with large datasets, such as the Airbnb dataset. Like XGBoost, LightGBM can manage skewed data by assigning more importance to underrepresented areas, ensuring better predictions on imbalanced datasets. The speed and efficiency of LightGBM further make it a valuable tool in my project, especially when computational resources are a concern.

Random Forest Regressor

Random Forest is an ensemble method that builds multiple decision trees and averages their predictions, which enhances its robustness and accuracy. I selected Random Forest for its ability to handle complex, non-linear relationships between features, such as interactions between property features (e.g., size, location). Random Forest also doesn't require assumptions about the distribution of the target variable, which makes it well-suited for handling left-skewed distributions like Airbnb prices. Additionally, the method's ability to average over multiple trees helps reduce overfitting, making it reliable for large datasets.

MLPRegressor (Multi-Layer Perceptron Regressor)

MLPRegressor is a neural network-based model that can capture non-linear relationships between input features and the target variable. This is especially useful when the relationships in the data are

complex, as is the case with Airbnb pricing, where factors like location, amenities, and seasonality may influence the price in non-linear ways. The flexibility of MLPRegressor allows it to model intricate patterns and higher-order interactions between predictors, making it a powerful tool for capturing the complexity of the data.

Gradient Boosting Regressor (GBR)
Gradient Boosting Regressor is another powerful ensemble method that focuses on iteratively refining predictions by correcting the errors of previous models. I chose GBR for its ability to handle skewed data, which is a key feature when working with the highly skewed price distribution in my dataset. GBR can model complex, non-linear relationships between features and target variables, which is crucial when predicting prices influenced by various property characteristics. Furthermore, GBR is robust to outliers, an important feature when predicting prices that may have extreme values.

Support Vector Machines (SVM) with Regression (SVR)
Support Vector Regression (SVR) is a powerful method for regression tasks that is particularly useful in high-dimensional spaces and non-linear scenarios. I chose SVR because it is effective at modeling non-linear relationships between features and the target variable, which is common in pricing tasks like predicting Airbnb prices. SVR is also robust to outliers due to its epsilon-insensitive loss function, making it a reliable choice for datasets with extreme values or noise, such as those involving pricing data. Its ability to perform well in high-dimensional spaces is also beneficial, given the numerous features in the dataset.

CatBoost
CatBoost is a gradient boosting algorithm specifically designed to handle categorical features efficiently. I selected CatBoost because it can handle datasets with both numerical and categorical features without extensive preprocessing. Like other gradient boosting models, CatBoost is effective for managing skewed distributions and can improve predictions by reducing bias and variance. It also requires fewer hyperparameter tuning steps compared to other models, making it both efficient and effective for my project.

Bayesian Ridge Regression
Bayesian Ridge Regression is a linear regression model that incorporates Bayesian inference to regularise coefficients and estimate model parameters probabilistically. I chose this model because it is particularly useful when dealing with skewed data and offers regularisation, which is important when working with a target variable like price that may exhibit significant skew. Bayesian Ridge Regression also provides probabilistic predictions, which is beneficial in quantifying uncertainty and making more informed decisions when predicting prices.

1.4. Evaluate and Improve the model(s)

Table 1: Hyperparameters of the Top 3 Models Selected for the Voting Models or Included in the Voting Ensemble

Model	Best Hyperparameters
LightGBM	{'subsample': 0.6, 'num_leaves': 31, 'n_estimators': 500, 'min_child_samples': 10, 'max_depth': 30, 'learning_rate': 0.01, 'colsample_bytree': 1.0}
CatBoost	{'random_strength': 10, 'learning_rate': 0.1, 'l2_leaf_reg': 5, 'iterations': 250, 'early_stopping_rounds': 30, 'depth': 6, 'bagging_temperature': 2}
XGBoost	{'subsample': 0.8, 'reg_lambda': 1, 'reg_alpha': 1, 'n_estimators': 200, 'min_child_weight': 3, 'max_depth': 7, 'learning_rate': 0.05, 'gamma': 0.2, 'colsample_bytree': 1.0}

Voting (tuned)	Best weights found: [2, 0.5, 2]

Table 2: Best Performing Models

Model	Train R^2	Train MAE	Train MSE	Test R^2	Test MAE	Test MSE	Explained Variance	Final Model
LightGBM	0.601	0.228	0.1000	0.502	0.257	0.124	-	-
CatBoost	0.582	0.248	0.105	0.485	0.128	0.129	-	-
XGBoost	0.614	0.228	0.097	0.505	0.260	0.124	-	✅
Stacking (3 Models)	0.59	0.22	0.10	0.50	0.25	0.13	Train: 0.59 Test: 0.50	-
Voting (no weights specified)	0.60	0.23	0.10	0.50	0.26	0.12	Train: 0.60 Test: 0.50	-
Voting (tuned)	0.60	0.23	0.10	0.50	0.27	0.12	Train: 0.60 Test: 0.50	-

When evaluating the performance of XGBoost against other models like LightGBM, CatBoost, and the ensemble methods (Stacking and Voting), I found XGBoost to be the best fit for predicting Airbnb prices.

XGBoost vs LightGBM:
In terms of performance, XGBoost clearly outperforms LightGBM. XGBoost has a higher Train R^2 of 0.614, compared to LightGBM's 0.601. This indicates that XGBoost explains more of the variance in the training data, which is crucial when predicting Airbnb prices, where a variety of factors (location, amenities, etc.) affect the price. Furthermore, XGBoost achieves a lower Train MSE (0.097) than LightGBM (0.1000), meaning it produces fewer large errors. While the Test R^2 values are quite similar (0.502 for XGBoost and 0.505 for LightGBM), XGBoost slightly outperforms LightGBM in Test MSE (0.124 vs 0.128), which indicates better predictive accuracy on unseen data. Given these results, I believe XGBoost is more reliable for generalising predictions, making it a better choice overall for Airbnb price prediction.

XGBoost vs CatBoost:
CatBoost has a lower Test MAE (0.128), suggesting it might have an edge in minimising absolute errors. However, when it comes to predicting Airbnb prices, capturing the overall variance (R^2) and minimising squared errors (MSE) are more important than just focusing on absolute errors. XGBoost excels in these areas with a higher Train R^2 (0.614) and a lower Train MSE (0.097), indicating it better captures the complexity of the price prediction task. Although CatBoost might be a better choice if minimising MAE is the top priority, I chose XGBoost because it offers a better balance of variance explanation and error minimisation, which are more critical for accurate price predictions in the Airbnb context.

XGBoost vs Stacking and Voting Models:
I also considered the Stacking and Voting ensemble models, but neither showed a significant improvement over XGBoost. The Stacking model has a Train R^2 of 0.59, slightly lower than XGBoost's 0.614, and a similar Test R^2 of 0.50. The Voting models had comparable results with a Train R^2 of 0.60. While ensemble methods can combine the strengths of multiple models, in this case, the added complexity of Stacking and Voting didn't provide meaningful gains in performance. Given that XGBoost already offers strong performance without the extra complexity, I felt that the simplicity

and efficiency of XGBoost made it a better choice.

Final Choice of XGBoost:
Ultimately, I chose XGBoost because it delivers the best performance across the most important metrics for predicting Airbnb prices. It not only has the highest Train R^2 but also the lowest MSE, indicating it fits the training data well and generalises effectively to new data. Although CatBoost's lower Test MAE is worth noting, I felt that XGBoost's superior R^2 and MSE made it the more robust model overall. Additionally, the lack of substantial improvement from the ensemble models made the decision clearer. XGBoost strikes the perfect balance between accuracy, interpretability, and computational efficiency, making it the best choice for this task.

1.5. Summary

In this case study, the problem statement was to predict Airbnb listing prices using a machine learning model. The problem stems from the complexities involved in pricing, such as factors like location, property type, amenities, and guest reviews. Current pricing models used by hosts can be outdated or overly reliant on personal judgment, leading to either underpricing or overpricing. These mispricing issues negatively affect booking rates, earnings, and competitiveness in the market. Moreover, the distribution of listing prices is heavily skewed, complicating the ability to establish fair pricing strategies.

To address these challenges, the goal was to build a data-driven machine learning model that could accurately predict prices by considering features such as location, property attributes, and guest reviews. By doing so, the model would enable Airbnb hosts to optimise their pricing strategies, improve occupancy rates, and maximise profitability. From a technical perspective, handling skewed distributions and complex feature interactions was a key consideration in selecting the appropriate model. The chosen machine learning models, such as XGBoost, LightGBM, CatBoost, and others, were selected based on their ability to deal with skewed data, capture non-linear relationships, and manage complex feature interactions, all of which are crucial for predicting Airbnb prices.

I focused on pre-processing data from a previous assignment, ensuring consistency by applying scaling methods in this second phase of the project. I also experimented with several models, carefully choosing the ones that handled skewed distributions well, such as Voting, LightGBM, and CatBoost. After evaluating the models, I found XGBoost to be the most effective in predicting prices. Its performance was superior in terms of variance explanation and error minimisation when compared to other models like CatBoost and ensemble approaches such as Stacking and Voting.

Ultimately, XGBoost was selected because it not only performed well on the training data but also generalised effectively to unseen data. It provided the best balance of minimising errors and capturing the complexity of the pricing task. This allowed me to confidently conclude that XGBoost was the ideal model for predicting Airbnb prices and could help Airbnb hosts make better pricing decisions, ultimately leading to improved revenue and guest satisfaction.

This project was not only about applying machine learning models but also about understanding the real-world business impact, helping Airbnb hosts make more informed, data-driven decisions. The process of navigating through data preparation, model selection, and evaluation provided me with a deeper understanding of how machine learning can be used to solve complex business problems.

4. Conclusion

The Airbnb Pricing Prediction use case demonstrates the transformative potential of machine learning in addressing complex, real-world business challenges. Below, I summarise the key takeaways and reflect on the broader implications of this project.

I think that this project highlighted the importance of aligning technical solutions with business objectives. Machine learning models are not just about accuracy but also about driving tangible business outcomes, such as reducing turnover or increasing revenue. And for model selection and evaluation, a variety of models were explored in both projects, including ensemble methods like Random Forest, Gradient Boosting (XGBoost, LightGBM, CatBoost), and Support Vector Machines. Each model was chosen for its ability to handle specific challenges, such as imbalanced data or non-linear relationships. In Airbnb Pricing Prediction, XGBoost was selected for its superior performance in explaining variance and minimising errors. It emphasised the importance of hyperparameter tuning and model evaluation metrics (e.g., R^2, MAE, MSE) to ensure robust and generalisable models. Ensemble methods like Voting and Stacking were explored in this project to combine the strengths of multiple models. While these methods added complexity, they did not always outperform individual models like XGBoost or LightGBM. This highlights the importance of balancing model complexity with performance gains. Additionally, I think that this whole project underscored the need for machine learning solutions to align with real-world business needs. In Airbnb Pricing Prediction, the goal was to optimise pricing for profitability and guest satisfaction. Where the success of these models depends not only on technical performance but also on their ability to address specific business pain points.

Lastly, for me, this project was not just about building models but also about understanding the broader impact of machine learning on business outcomes. They reinforced the importance of balancing technical rigor with practical considerations, such as interpretability, scalability, and real-world applicability.

5. Reflection

5.1. Suggest possible further improvement(s) to the current ML solution.

To further enhance the data, expanding the collection to include both structured and unstructured data can improve model robustness. Structured data, such as seasonal pricing trends and guest demographics, can provide more comprehensive insights, while unstructured data like property images and guest reviews can offer valuable context to support better predictions.

To improve the model architecture, parallel modeling can be implemented by combining structured CSV data, such as booking history, with unstructured image data using multimodal architectures. For example, employing Convolutional Neural Networks (CNN) for image data and XGBoost for tabular data can leverage the strengths of both data types. Additionally, implementing Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM) models can capture sequential patterns, such as guest stays or employee career trajectories. To further boost prediction accuracy, ensemble methods like stacked models or gradient-boosted trees can be deployed, allowing for improved performance while managing computational constraints effectively.

To optimise computation, upgrading infrastructure by scaling RAM and GPU resources is essential to efficiently handle complex models like deep neural networks, ensuring faster processing and better performance. Additionally, migrating to the cloud can further enhance computational power by utilising platforms like AWS SageMaker or Google Colab for distributed training of resource-intensive models. This approach enables more scalable and efficient model training, reducing bottlenecks and leveraging cloud-based resources for improved performance and flexibility.

Deployment & Integration

Idea 2: Airbnb host toolkit

For Airbnb Price Prediction Model, I think that we could embed price optimisation models into

Airbnb's host portal via API partnerships. And also create a chrome extension to analyse competitor listings using existing ML models.

Business Value of these ideas

We could pilot a premium "Smart Pricing" feature for Airbnb hosts with revenue-sharing models. And for HR Analysis Prediction Model, we could license HR promotion prediction models to mid-sized enterprises through SaaS platforms. These improvements balance technical feasibility with business impact, focusing on scalability, usability, and measurable outcomes. And during that project if we do pursue further improvements, we should prioritise integration pipelines first so that we can demonstrate immediate value before expanding model complexity.