Data-Driven Pricing Strategy for Airbnb in Australia

Introduction

In the rapidly evolving sharing economy, Airbnb has recognizes the necessity for a intricate pricing model that accurately reflects the numerous factors that affect rental prices. Our data analysis team has been commissioned to develop a predictive pricing service aimed at calculating the recommended nightly prices for Airbnb listings. This service is particularly focused on properties along Australia's dynamic East Coast, where unique urban and natural attractions play a significant role in shaping rental prices.

Our objective is to harness advanced statistical learning techniques to forge a robust model capable of nightly price prediction for a range of properties. It is committed to providing landlords with data-driven decision-making capabilities and enhancing their competitive advantage in the market while maximizing profitability.

Our approach began with meticulous data cleaning and pre-processing to verify the accuracy and integrity of the information in our training dataset. We then proceeded to examine the relationship between property characteristics and their respective prices, considering factors such as location, size, bedroom count, amenities quality, and customer reviews.

We adopted multiple models and conducted cross validation to determine the most suitable prediction method. These models include Ridge regression, regression tree, and gradient boosting. We found that the Gradient Boosting Machine performs best in handling complex nonlinear relationships with its powerful performance. This model not only helps new landlords determine prices, but also allows Airbnb to monitor and optimize pricing strategies across the entire platform, thereby improving revenue and competitiveness.

1.Data Processing

1.1 Data description

The dataset we are processing is primarily concerned with real estate leasing information. It comprises a training set of 12,500 entries and a test set of 2,500 entries. The variable 'price' is considered the predictive target, as it is present in the test set. The data quality is notably high, with only two missing values in the 'Bed' field within the training set. Additionally, there are some outliers which we plan to address appropriately in our subsequent research and analysis. Our objective is to establish a price prediction model that relies on thorough exploratory analysis, which will guide the pre-processing and the development of the model.

1.2 Data pre-processing

In the process of data analysis and model construction, understanding the distribution characteristics of target variables is a crucial step. We have conducted in-depth exploration and research on the data to obtain a clear overview of its distribution, central trend, and variability.

We observed that the average rent of the 'price' in the training set was approximately AUD 285.37, with a standard deviation of AUD 201.28, indicating significant volatility in the data. There is a significant difference between the maximum and minimum prices, ranging from AUD 19 to AUD

1249, indicating a high likelihood of outliers. When processing these data, we will consider the effects of skewness and kurtosis, and implement some form of data transformation, such as logarithmic transformation, to stabilize variance and reduce the impact of outliers.

We also noted that the maximum number of bedrooms is 35, an outlier. However, given that it could represent a legitimately large property, such as a villa or castle, we will retain this data point.

Firstly, in the data processing stage, we conducted missing value detection and found that there were missing values in the 'beds' column in the training set. To maintain the original distribution of the data and not introduce new biases, we chose to fill the missing values in this column with mode.

At the same time, we observed that the 'bedrooms' and' beds' columns have floating-point types. Theoretically, there are no half beds or half bedrooms, so we changed the floating-point type to an integer here to make the data more reasonable. To ensure consistent performance of the model during training and testing, the 'bedrooms' and' beds' columns were also processed similarly in the 'test' dataset.

In 'property', here are many different categories in the category feature 'type'. To reduce model complexity, we set a threshold of 100 and classify categories with frequencies below the threshold as' Other '. This not only reduces the total number of categories, but also helps the model to better generalize and avoid overfitting problems caused by too many categories.

2. Exploratory Data Analysis

To visually display the distribution of data and enable us to quickly understand the main characteristics of the data, a histogram visualization of "price" was performed to describe the distribution of housing quantity in various price ranges.

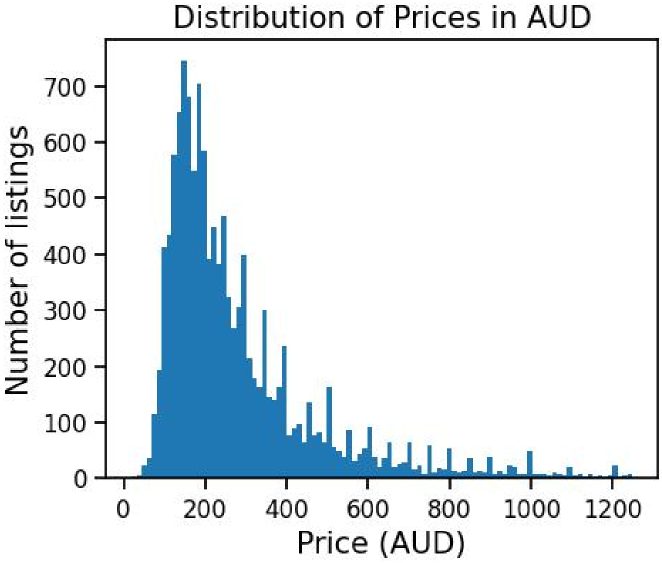

Figure 1 Distribution of Prices in AUD

We observe that most property prices range from AUD 0 to AUD 400, peaking between AUD 100 and AUD 200, indicating a significant right-skewed distribution. This suggests that while most housing prices are relatively low, a few are exceptionally high. To address the right skewness and better satisfy the assumptions of linear regression and other models, we will apply a logarithmic transformation to the data.
As we can see from Figure 2 below, the ‘Price’ data distribution after logarithmic transformation is closer to a normal distribution.

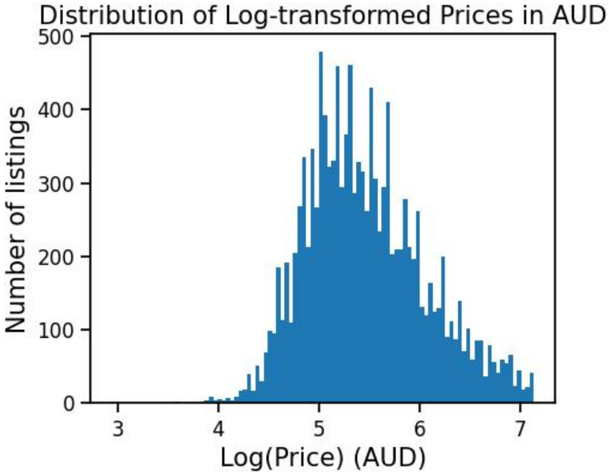

Figure 2 Distribution of Log-transformed Prices in AUD

To help users better understand the trends and correlations in the data, we visualize the relationship between each numerical feature in the training dataset and the 'price'. Based on this result, we can obtain some information.

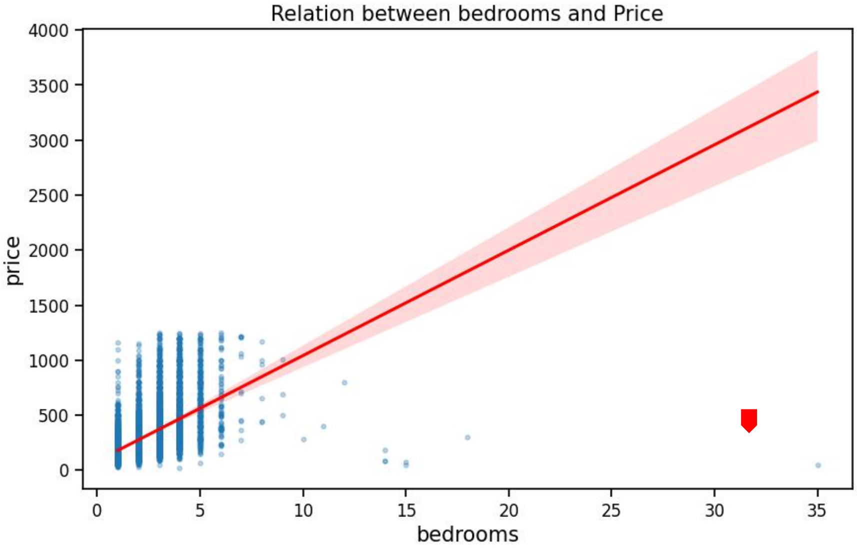

Figure 3 Relation between bedrooms and Price

Based a Figure 3, there is a positive correlation between the number of "bedrooms" and the "price". As the number of "bedrooms" increases, the price also tends to rise, which is an important feature for the predictive ability and explanatory power of the model.

For landlords, understanding how the number of bedrooms affects prices can help them better price their listings. Landlords may set higher prices for listings with multiple bedrooms and can also help predict market demand for different types of listings. For example, if large family or group tourists are more common in a certain region, listings with more bedrooms may be more popular, resulting in higher prices.

To examine the relationship between categorical variables and price, we will utilize Boxplots for visualization. These plots will specifically compare the ‘price’ variable with different categorical variables, focusing on the top 20 most common categories to maintain graph clarity and prevent confusion from an excess of categories.

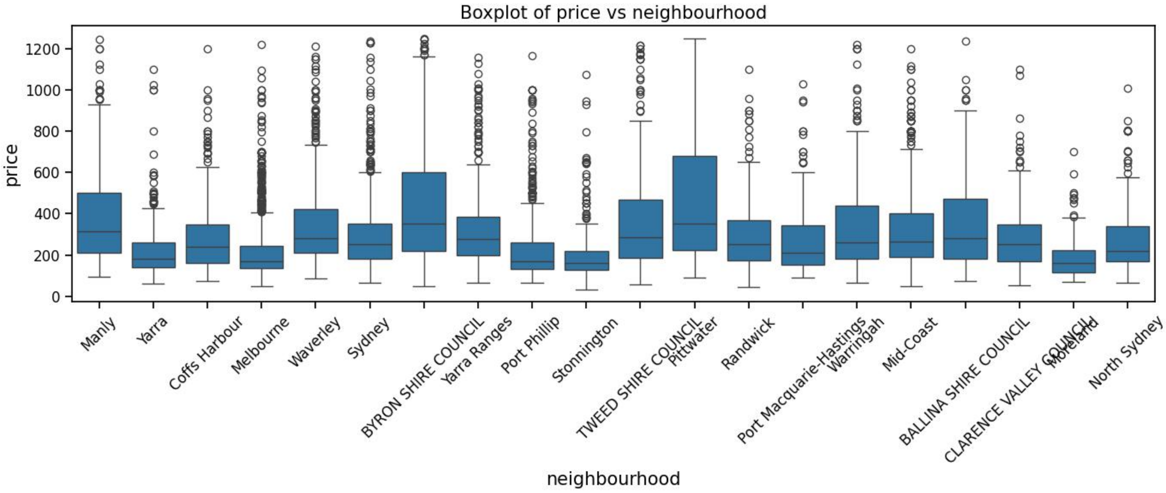

Figure 4 Boxplot of price vs neighbourhood

We observe the relationship between the price and the neighborhood, which are the more important variables. According to Figure 4, this is to ensure the clarity of the graph and avoid confusion caused by too many categorical. We can see that the median housing prices in each neighbourhood, some places have significantly higher prices than others, such as "BYRON SHIRE COUNTIL," "Sydney," and "TWEED SHIRE COUNTIL," with relatively high median prices. We found that almost all regions showed outliers in prices, which may indicate that the prices of some properties are much higher than the typical prices in the region. In certain regions such as Sydney and North Sydney, even with many outliers, the median is still in a lower price range, which means that although there are high priced listings, the prices of most listings are still relatively moderate.

The high values in the dataset may not necessarily indicate errors or anomalies; instead, they likely reflect the real market price variations among different types of accommodations. We consider their presence to be justified.

Next, we extract words from each description, filter out common English stop words and punctuation, and ultimately output the twenty most common words found in the dataset after this processing.

To better provide intuitive information, visualization will be done through bar chart format. This can quickly understand the main content and keywords described in the dataset, thus facilitating better data analysis and decision-making.

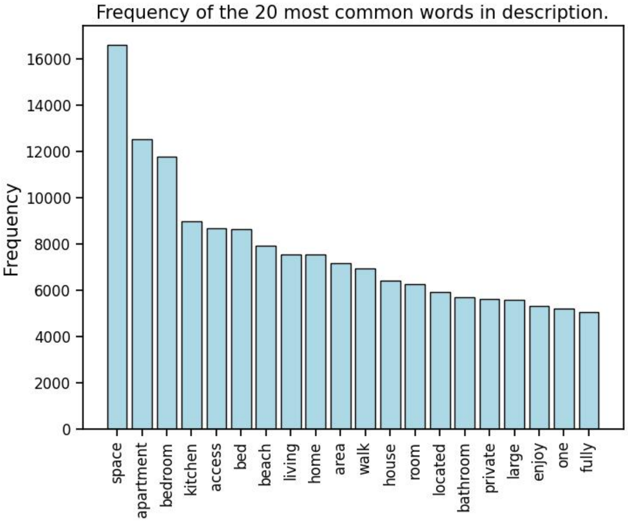

Figure 5 Frequency of the 20 most common words

From Figure 5, it can be concluded that the terms "space", "apartment", "bedroom", and "kitchen" have been mentioned multiple times. Therefore, preliminary analysis suggests that these variables may be key factors or attributes that potential tenants are more willing to consider when viewing properties. The frequency of words such as' beach 'and' access' may also indicate the importance of location attributes and accessibility in descriptions. This may help us understand the content that people prioritize or consider these attributes to be the most attractive and can help us understand the focus and features described in the dataset.

Based on Figure 6, using bar charts to visualize the top 10 amenities from the convenience features of the dataset can be used to predict models to evaluate how the presence or absence of these amenities affects prices.

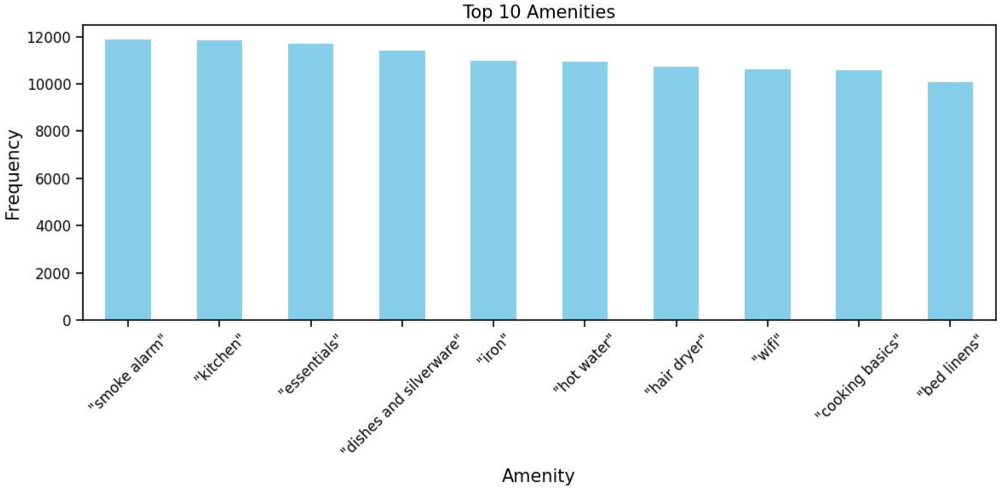

Figure 6 Top 10 Amenities

To explore the relationship between price and geographical location, a latitude and longitude scatter plot was used to display the geographical distribution of Airbnb rental prices in different cities. Several major cities in Australia, such as Sydney, Melbourne, and the Gold Coast, were also annotated.

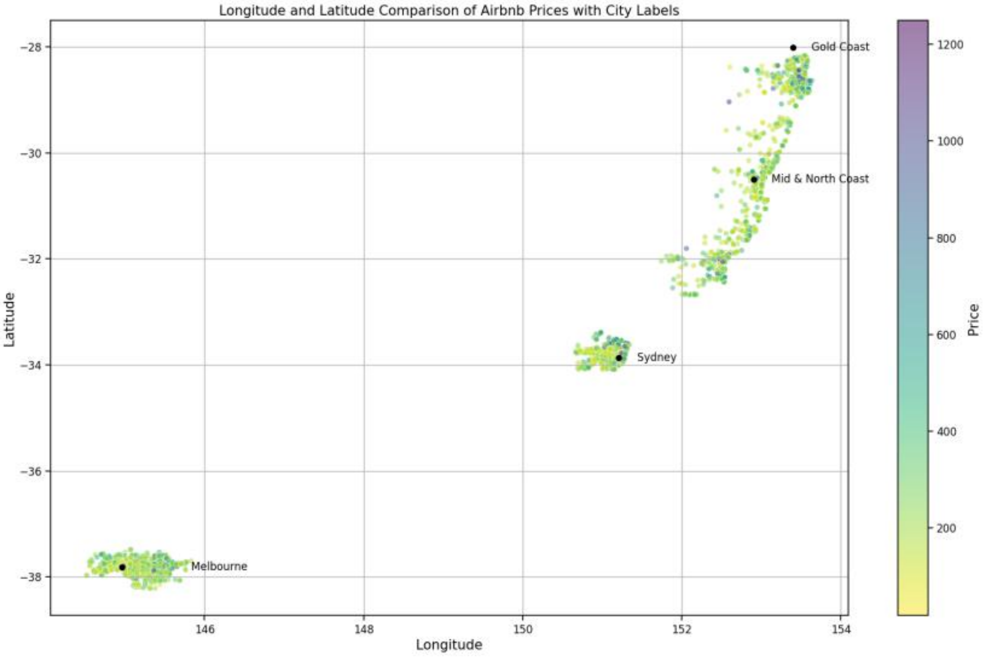

Figure 7 Longitude and latitude comparison of Airbnb prices

According to Figure 7 above, we found that prices exhibit significant geographical stratification, especially in Sydney. We can see that the prices in the northeast corner of Sydney are higher, and this area may contain high-value accommodation locations, such as areas close to beaches, city centres, or other tourist attractions.

The distribution of Melbourne shows a relatively concentrated price range, which may reflect the homogeneity of accommodation prices within the city. Compared to Melbourne, the price distribution in the Gold Coast region is wider, with colours extending from light green to purple, indicating a significant variation in accommodation prices. Considering that the Gold Coast is a famous tourist destination with many beaches and vacation facilities, this may reflect the existence of some high-end or characteristic accommodations.

In short, prices vary with the distance from the city centre and coastline. Areas close to the city centre or with excellent natural landscapes such as beaches will have higher tourist attractions, thereby driving up rental prices.

This is valuable information for potential tenants and landlords, as it can help understand the impact of different geographical locations on rent pricing. At the same time, it can also guide price setting, market positioning, and strategic planning.

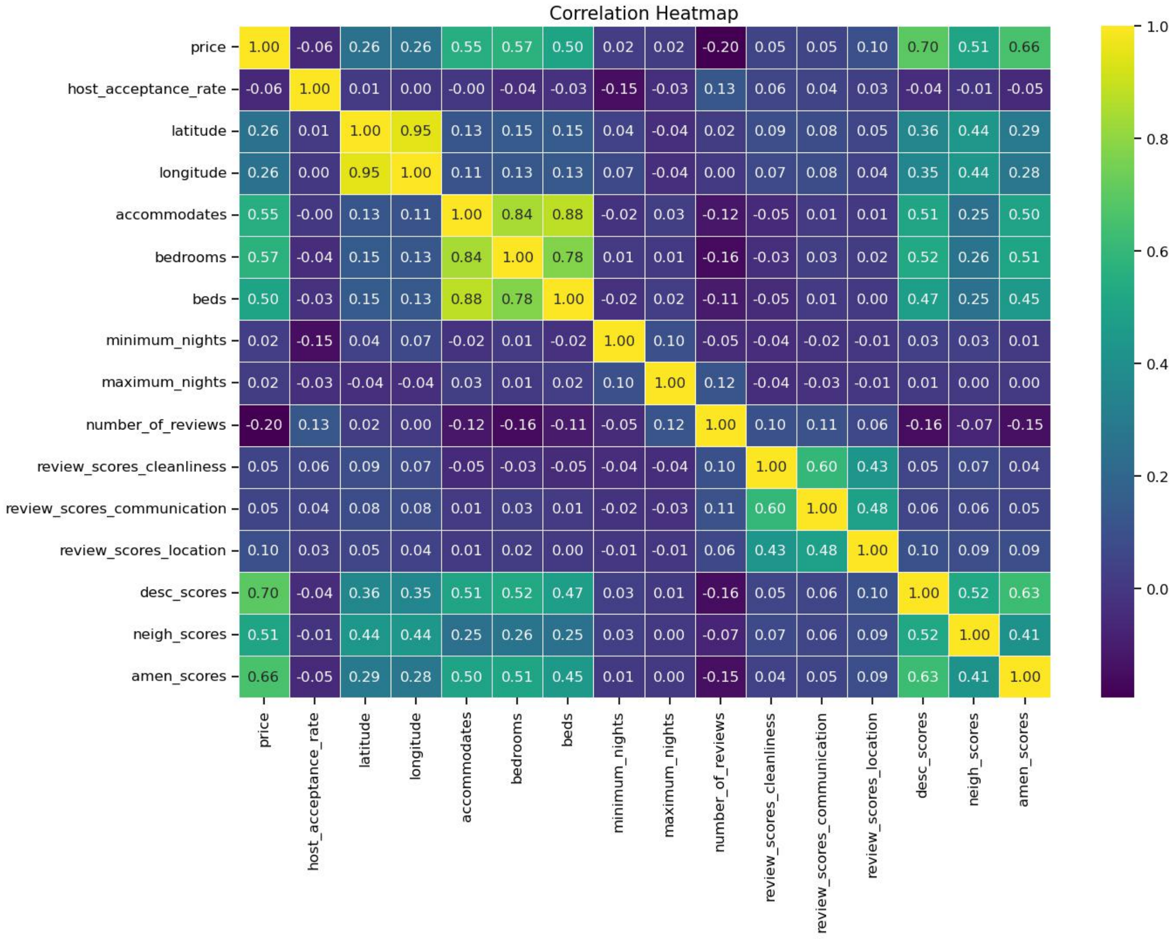

Figure 8 Correlation Heatmap

According to Figure 8, we can see that ‘price’ has a strong positive correlation with ‘accommodates’, ‘beds’, and ‘bedrooms’, which means that houses with more bedrooms or beds are relatively more expensive, this information is very helpful for us to conduct subsequent model training in the future.

3. Feature Engineering Process

3.1Text Data Transformation

Text Data Processing

Our approach to feature engineering began with meticulous pre-processing of text data. Recognizing that unstructured text data holds a wealth of information that could potentially be leveraged for our predictive models, we employed a series of text normalization techniques:

- **HTML Tag Removal**: We used ‘Beautiful Soup’ to cleanse our text data of any HTML tags, ensuring that only content text was subjected to further analysis.

- **Text Lowercasing**: To maintain consistency and avoid duplications caused by case differences, we converted all text to lowercase.

- **Punctuation and Numeric Removal**: Regular expressions were used to replace punctuation and numbers with spaces, eliminating irrelevant characters from our text.

- **Stop word Exclusion**: Employing a predefined list of English stop words, we filtered out high-frequency words that carry minimal informative weight.

- **Stemming**: We applied ‘Porter Stemming’ to reduce words to their base or root form, streamlining our feature space and focusing on the essence of the content.

These pre-processing steps transformed our textual data into a cleaner, more analysable form, ripe for feature extraction.

Feature Extraction with TF-IDF

For transforming pre-processed text into numerical features, we chose the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. This method was preferred over others for its ability to reflect the importance of words in relation to both the document and the entire corpus. By setting a maximum feature threshold of 1000, we balanced between a rich representation of our textual data and computational efficiency.

We vectorized three key text columns—'description', 'neighborhood_overview', and 'amenities'—each providing a unique perspective on the listing's potential influence on pricing.

Linear Regression on Text Features

Our feature engineering extended into the realm of predictive modelling, where we trained separate Linear Regression models for each TF-IDF vectorized text feature. This step was pivotal as it allowed us to extract predictive scores from the models, which we hypothesized would be indicative of the textual data's influence on listing prices.

We diligently ensured that each step of our feature engineering process was guided by the principle of relevance and justification. The decision to employ TF-IDF, for instance, was supported by its proven effectiveness in capturing the essence of text data for linear models. Similarly, the choice of stemming and stop words was made after careful consideration of the trade-offs between feature space dimensionality and informational integrity.

3.2 Correlation Analysis with Price

We performed a correlation analysis to understand the relationship between various factors and the listing price.

Through meticulous pre-processing and TF-IDF vectorization of text data, we engineered predictive features reflecting the informational value of listings' descriptions, neighbourhood overviews, and amenities. Our linear models, trained on these features, have demonstrated significant positive correlations with the price, affirming their predictive power.

Our correlation analysis has identified key features that influence listing prices. The engineered text scores (**desc_scores**, **neigh_scores**, **amen_scores**) are notably the most correlated with price, followed by the number of bedrooms and accommodation capacity. The geographical coordinates showed a moderate correlation, suggesting locational impact on pricing. In contrast, the host acceptance rate and number of reviews displayed a negative correlation with price.

3.3 Feature Scaling Based on Correlation

To refine our predictive model, we implemented a feature scaling strategy based on the correlation strengths of individual features with the price. By scaling each feature by its absolute correlation value, we enhanced their relative importance in line with their predictive relevance. This consistent scaling across both training and test sets ensures model robustness and maintains data integrity, ultimately improving the accuracy of our pricing predictions.

3.4 Multicollinearity Assessment

In ensuring the robustness of our regression model, we performed a multicollinearity check using the Variance Inflation Factor (VIF). Our VIF analysis revealed that most features are well within acceptable limits, with latitude, longitude, accommodates, and ‘price_log’ marginally exceeding the threshold, suggesting a moderate interdependence.

3.5 Location-Based Feature Engineering

We augmented our dataset with a new feature, 'neighbourhood_rank', which classifies neighbourhoods into 'High', 'Medium', or 'Low' price categories. This categorization was derived from the average listing price per neighbourhood, with thresholds set at the 33rd and 67th percentiles of these averages. This strategic grouping reflects the varying desirability and expected pricing within different locations, providing our model with a nuanced understanding of location value.

The introduction of 'neighbourhood_rank' aligns with our objective to capture significant location effects on rental prices, thereby improving the model's accuracy and interpretability.

4. Methodology

In this section, three models are selected: ridge regression, regression tree, and gradient boost. The standardized numerical features applied include host acceptance rate, latitude, longitude, accommodates, bedrooms, beds, minimum nights, maximum nights, number of reviews, review scores cleanliness, review scores communication, review score’s location, and three score variables created by feature engineering (description scores, neighbourhood overview scores, and amenities scores). Categorical features (neighbourhood, property type, and room type) are converted into a set of binary dummy variables by one hot encoder. And the response is the price.

4.0 Data splitting

Before training the model, we split the in-sample data into the training and validation sets. The training set contains 80% in-sample data and is used to train the models. And the other 20% is the validation set, which is used to compare the generalization performance of the models. After model selection, we will train the final model with all the in-sample data.

4.1 Ridge regression

Firstly, we built a ridge regression model as the interpretable linear model. The reason that we chose this model is that ridge regression solves the overfitting problem in simple linear regression models by adding a penalty term that measures the model complexity:

$$\text{minimize}\ \{\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij})^2+\lambda\ \sum_{j=1}^{p}\beta_j^2\},$$

where λ is a tuning parameter controlling the shrinking level. Ridge regression assumes there is a linear relationship between the response and the predictors, as does linear regression.

Regularization strength depends on the tuning parameter λ. The larger λ, the smaller coefficients βs. We specified the candidate λ as 151 evenly spaced numbers on the log scale between 10^{-5} and 10^{4}. The optimal λ is determined by 5-fold cross validation.

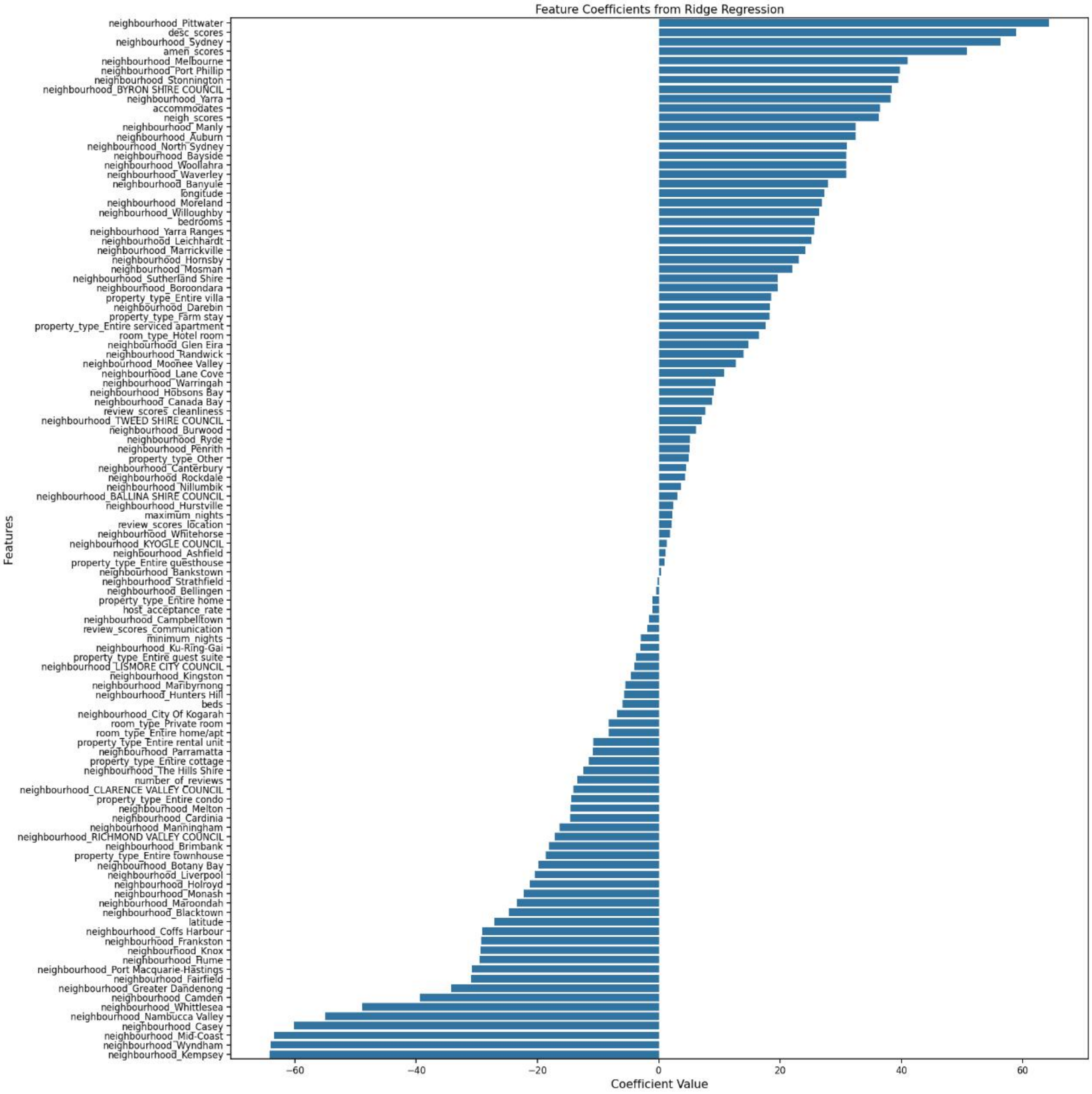

Figure 9 Coefficients in Ridge regression

The model is trained on the training set. The selected λ value is 13.18, and the detailed coefficients are shown below. The characteristics that most increased the price were a neighbourhood in Pittwater or Sydney or having a higher description or amenities score. Neighbourhoods in Kyogle Council, Ashfield, Bankstown, Strathfield, or Bellingen have little effect on prices. And neighbourhoods in Kempsey, Wyndham, Mid-coast, or Casey have a significant negative effect on the price. Therefore, in this model, neighbourhood is a crucial feature influencing the price. Unlike lasso regression, ridge retains features with small coefficients, so that each feature influences the response.

4.2 Regression tree

The regression tree method is a supervised machine learning method that accommodates nonlinear relationships and can ignore multicollinearity, which is suitable for the multivariate study of this case. The regression tree model divides the data into multiple non-overlapping regions where observations are close in all dimensions and assumes the prediction is the average response of each region,

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

where R_m is the region and c_m is the average response of the corresponding region R_m.

The regression tree is grown by the transformed training set with parameters including the minimum sample number of each leaf (15) and a random seed. We used cost-complexity pruning to shrink the tree to its optimum size. The cost-complexity criterion is:

$$R_\alpha(T) = R(T) + \alpha \cdot |T|$$

, where T is the target sub-tree and α is the tuning parameter that measures the degree of pruning. Fitting the model with the training set, the optimal α is 40.51, found by a grid search with 5-fold cross validation.

The following visualization shows the relative importance of each feature. The rental market is a perfect competitive market where the prices set by the supply side are also determined by the willingness of the demand side. The description score is the most important feature, with at least triple as much importance as other features. So, an attractive description can easily increase the price of a property. The amenities score, the number of bedrooms, and the neighbourhood overview score also have significant importance. It follows that users prefer more comprehensive amenities and better locations. And more bedrooms in a listing can accommodate more renters and cost more. The importance of room type, property type, and neighbourhood is almost zero, which reflects users focus on living experiences rather than listing type and location.

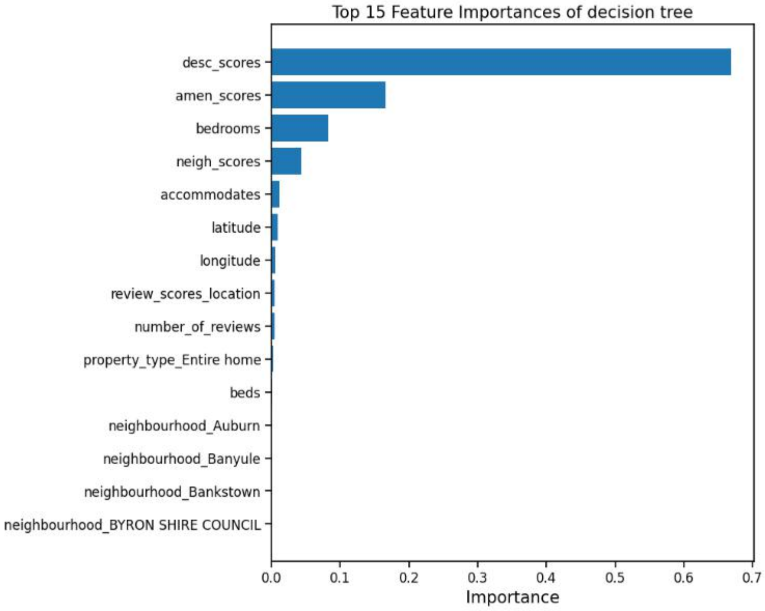

Figure 10 Feature importance in regression tree

4.3 Gradient-boosting

The reason we chose gradient boosting is that it has high prediction accuracy with relatively few parameters. The gradient-boosting method consists of multiple trees. An important assumption is that the model is in an additive form, as determined by the additive loss correction iteration. Each tree fits the loss of the previous tree:

$$Tree_m \quad (x) = Tree_m - 1 \quad (x) - \alpha \cdot \nabla L\,(y, Tree_m - 1 \quad (x))$$

L is the loss function, y is the true value, $Tree_m(x)$ is the m-th tree, $Tree_{m-1}(x)$ is the previous tree, and α is the learning rate, and is updated to add to the previous tree:

$$Model_m \quad (x) = Model_{m-1}(x) + \alpha \cdot Tree_m(x)$$

where $Model_m(x)$ is the updated model after adding the m-th tree and $Model_{m-1}(x)$ is the cumulative model up to the previous tree, α is the learning rate which controls the contribution of each tree. Each fit is a weak learner that is related to the previous loss rather than the response, and the model assumes the $f_M(x)$ the final model after M iterations of the boosting process a strong learner.

After the train-test split, the training set contains 10,000 observations and is a small dataset. We specified a learning rate of 0.01 or 0.05, the number of trees of 1000 or 2000, the maximum depth of 3 or 5, and the fraction of samples of 0.5 or 0.6. The model is fitted on the training model with grid search using 10-folds cross-validation for these tuning parameters.

The tuning parameters selected by the grid search and corresponding analysis are: The learning rate is 0.01. Lower leaning rate allows each tree to contribute less to the final prediction, so that prevent overfitting. The maximum depth is 5. Higher depth allows for more detailed binary splitting to capture the pattern. The number of trees is 1000. The model has a satisfying prediction performance after 1000 iterations. The fraction of samples is 0.6. Higher fraction improves the randomness and avoid overfitting.

The relative importance of each feature is shown below. The features with higher importance are consistent with the regression tree model. But in the model, the gap between the number of bedrooms and the neighbourhood overview score is smaller. This may be because the regression tree model focuses only on the splitting of one tree, while the gradient-boosting model corrects the loss by iterating over multiple trees.

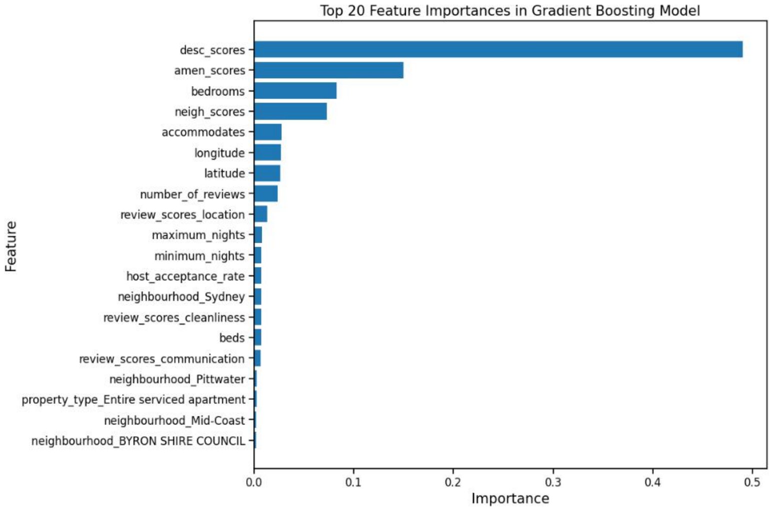

Figure 11 Top 20 Feature Importance in Gradient Boosting Model

5. Evaluation and Comparison

Model Validation

Our evaluation focused on the Root Mean Squared Error (RMSE) for three models: Ridge CV, Decision Tree, and Gradient Boosting. The RMSE values, reflective of the validation data, were:

- **Ridge CV:** 115.09
- **Decision Tree:** 124.87
- **Gradient Boosting:** 105.74

These figures fall within the expected range of approximately $100-250 AUD for the training and validation RMSE, suggesting a reasonable model performance. Notably, the Gradient Boosting model exhibits the lowest RMSE, indicating its higher accuracy in this context.

Model Retraining and Grid Search CV

We combined the training and validation sets to retrain our Gradient Boosting model, enhancing its learning with a broader dataset. To optimize our model parameters, we employed ‘Grid Search CV’, which exhaustively searched through a specified parameter grid over a 10-fold cross-validation. The parameters included learning rate, max depth, number of estimators, and subsample rate.

The final model pipeline integrated a pre-processor with ‘Standard Scaler’ for numerical features and ‘One Hot Encoder’ for categorical variables, ensuring that all features contributed appropriately scaled and encoded information to the model.

Test Set Predictions and Kaggle Submission

Predictions on the test set were generated using the same feature set as the training model to maintain consistency. The mean price of our test predictions was $281.83, reflecting the general pricing trend anticipated by our model.

The submission file, 'model3_gbb_submission.csv', was created with these predictions and included the necessary identifiers for Kaggle submission. The model demonstrated an RMSE of 116.71 on the public leader board and 125.29 on the private leader board.

Model Comparison in Business Context

When comparing these models, it's essential to consider their complexity, interpretability, and suitability for our business goals: understanding the factors affecting listing prices and accurately predicting them.

- **Ridge CV**: Offers a balance between complexity and interpretability. However, its slightly higher RMSE compared to Gradient Boosting suggests a lower predictive accuracy.

- **Decision Tree**: Provides high interpretability but at the cost of increased RMSE, making it less suitable for precise price predictions.

- **Gradient Boosting**: Exhibits the highest complexity but also the best performance in terms of RMSE. Its ability to capture complex nonlinear relationships makes it highly suitable for predicting listing prices, albeit with reduced interpretability.

Limitations in Business Context

Regarding limitations, each model presents unique challenges:

- **Ridge CV**: While Ridge CV effectively manages multicollinearity and overfitting, its linear nature restricts its ability to model the complex and nonlinear interactions often present in real estate pricing data, potentially leading to underfitting and less precise predictions.

- **Decision Tree**: Decision Trees are highly interpretable but are often too sensitive to small data variations, resulting in overfitting. They may not capture the broader trends necessary for making accurate price predictions in a market with diverse and fluctuating characteristics.

- **Gradient Boosting**: This model's strength in handling complex datasets comes with the risk of overfitting, especially with noisy data. Its predictions can be less transparent, making it challenging to extract straightforward rules or insights that are valuable in a business context.

Conclusion

In summary, our comprehensive analysis and modelling efforts have provided Airbnb with a robust predictive pricing tool that considers the multifaceted nature of the Australian East Coast market. By meticulously pre-processing data, engaging in exploratory analysis, and applying advanced machine learning techniques, we have developed a service that offers landlords actionable insights and competitive pricing strategies.

Our findings underscore the significance of property attributes, location, and amenities in influencing rental prices. The Gradient Boosting model has emerged as the most accurate, adeptly managing complex nonlinear relationships within the dataset. Although its predictions are less interpretable compared to simpler models, the trade-off for precision and adaptability to diverse data characteristics is deemed valuable for Airbnb's dynamic pricing needs.

While each model has its limitations, such as the potential for overfitting or underfitting, our validation and testing processes have refined their predictive capabilities. The models offer a spectrum of utility, from the interpretability of Decision Trees to the nuanced complexity captured by Gradient Boosting, providing Airbnb a suite of options tailored to various business scenarios.

Ultimately, our endeavour equips Airbnb with a data-driven foundation to optimize their pricing strategies, thereby enhancing profitability and market presence. This service not only benefits landlords by maximizing their earnings but also ensures that Airbnb remains at the forefront of the sharing economy, empowered by statistical learning and advanced analytics.

References

No references

Statement of Contribution

Team Member Contributions:

1. **Mengran Cheng (520185557):**
    - Contributed to writing the report sections on the Introduction, Data Processing, and Exploratory Data Analysis (EDA).
    - Engaged in discussions with the team to suggest modifications to the data cleaning and EDA code.
    - Responsible for documenting meeting notes and collaborated with Xiaoxuan Peng in writing the meeting minutes section.
2. **Pengchen Ren (530295990):**
    - Involved in late-stage code adjustments, model modifications.

    - Authored sections of the report on Feature Engineering Process, Evaluation and Comparison, and Conclusion.
    - Managed the consolidation and formatting of the final report.
3. **Yishuo Chen (530116938):**
    - Fully responsible for writing and correcting all code sections.
    - Took charge of the entire modelling process, from conceptualizing and building the three different models to parameter tuning and making iterative improvements.
    - Executed all visualization tasks and underwent over 30 revisions post-model completion to achieve a satisfactory RMSE.
4. **Yixuan He (520602308):**
    - Played a key role in coding for Data Cleaning and EDA.
    - Actively participated in writing Feature Engineering and provided instructions for Modelling.
    - Added descriptive Markdown commentary below each code cell in the Jupyter notebook.
5. **Xiaoxuan Peng (510600949):**
    - Participated in writing the Methodology section of the report.
    - Supplied model parameters during the coding phase and visualized specific values for the model's parameters and coefficients.
    - Co-authored the meeting minutes section with Mengran Cheng.

Use of Generative AI (e.g., ChatGPT):

- Throughout the Jupyter Notebook coding process, ChatGPT was utilized to assist with debugging, providing creative ideas, and correcting grammatical errors and typos. Its generative capabilities were instrumental in refining code, enhancing the clarity of written content, and aiding in the visualization of complex data.

Appendix

Meeting Minutes 1

Date: 23 Oct 2023

Present: 510600949; 520185557; 520602308; 530116938; 530295990

Apologies: \

Agenda

1. Review the guidelines and requirements for the Kaggle competition.
2. Assign roles and responsibilities to team members.
3. Discuss strategies for model construction and validation.
4. Agree on communication plans and meeting arrangements and develop the waterfront for the next meeting.

Meeting notes

Firstly, an overview of the Kaggle competition will be provided. For convenience, members without a Kaggle account will use social login to create an account. The team leader will be responsible for creating the Kaggle team and coordinating meetings, conducting data analysis, and pre-processing, and preparing for the first model building sprint starting on October 29th.

Actions

Here you should list the actions each member should take before the next meeting. You should also list tentative due dates for each action. Here is an example of what action items may look like.

- 530295990, 530116938 (28/10/2023): Draft a schedule for model testing rounds to ensure sufficient time to validate and select the final model

- 520185557, 510600949 (28/10/2023): Establish shared documents for project reports and outline the structure according to competition guidelines, research, and document potential model evaluation techniques.

- 520602308(28/10/2023): Lead the initial data pre-processing stage, modify, and improve the pre-processing process, and lay the foundation for model establishment

Report progress at any time and discuss in group groups.

Meeting Minutes 2

Date: 30 Oct 2023
Present: 510600949; 520185557; 520602308; 530116938; 530295990
Apologies: \

Agenda

1. Complete the review and evaluation of the first round of model construction.
2. Discuss the preliminary sprint results of the Kaggle competition.
3. Filter out models that meet the requirements for further optimization.
4. Review whether the visualization charts of EDA (exploratory data analysis) meet the predetermined standards.

Meeting notes

All members completed the first round of model construction and testing on time. After team discussion, several well performing models were selected for further adjustment and optimization. The code format has been preliminarily unified and further review is needed to ensure consistency. The EDA visualization chart basically meets the requirements, but some charts need to be fine-tuned to improve the efficiency of information transmission.
Actions

Here you should list the actions each member should take before the next meeting. You should also list tentative due dates for each action. Here is an example of what action items may look like.

- 530295990, 530116938 (2/11/23): Based on the results of the first round of model sprint, adjust and optimize the model, prepare for the second round of model testing, including parameter adjustment and cross validation

- 520185557,510600949 (2/11/23): Conduct a final review of the code format to ensure cleanliness, consistency, and ease of understanding. Update the shared document of the project report to reflect the latest progress.
- 520602308 (1/11/23): Review and fine-tune EDA visualization charts to ensure compliance with requirements, Provide improved visual charts to the team for final review.

Given the approaching deadline for the Kaggle competition, team members need to ensure the timeliness and quality of their work. Ensure that all submitted work is jointly reviewed and agreed upon by the team.

Meeting Minutes 3

Date: 3 Nov 2023
Present: 510600949; 520185557; 520602308; 530116938; 530295990
Apologies: \

Agenda

1. Explain code logic.
2. Discuss report requirements.
3. Update Kaggle

Meeting notes

Explain the current models; Identify features engineering that need to be changed; Discuss hyperparameters; Arrange the division of the report; Publish the files for Kaggle.

Actions

- 520602308, 530116938 (3/11/2023): Finalize the code
- 530295990 (3/11/2023): Update Kaggle
- 520185557 (4/11/2023): Write the report - Data processing & EDA
- 510600949 (5/11/2023): Write the report – Methodology
- 530295990 (5/11/2023): Write the report – Feature engineering, Validation, Conclusion