Predicting Property Price Outcomes in Australia Using Machine Learning

Word Count: 1,635

1.1 Introduction

This report presents machine learning models to predict whether 6,957 Australian homes (Feb 2022–Feb 2023) will sell above or below their listing price, helping sellers set more accurate prices in a market where small errors can incur large financial costs.

1.2 Numerical Data Exploration and Preparation

Section 1.1 of the Jupyter Notebook, starts by exploring the data, learning its size, datatypes, duplicate rows, `NaN` values and statistics. The target class is mapped to numerical values 1 and 0, duplicate rows are removed and features with >25% missingness are removed, while remaining gaps under the threshold are imputed with medians (`property_size`, `number_of_baths` and `number_of_parks`). Unrealistic outliers are then handled based on the statistics.

After excluding *Equal* outcomes to make it a binary decision problem and performing data cleaning, the final dataset contains 4,937 properties with 63.6% *Higher* and 36.4% *Lower* price classifications as seen in figure 1.

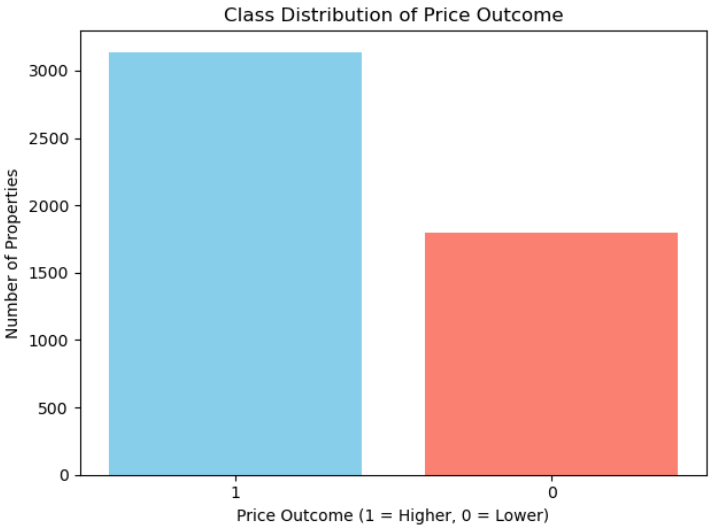

Figure 1: Class distribution of target variable

To complement the data preparation process, a feature correlation matrix was generated to identify variables most strongly associated with the target outcome. This provided early insight into potentially important predictors. However, as seen in figure 2, only `listed_price` seems to be correlated with `price_outcome`.

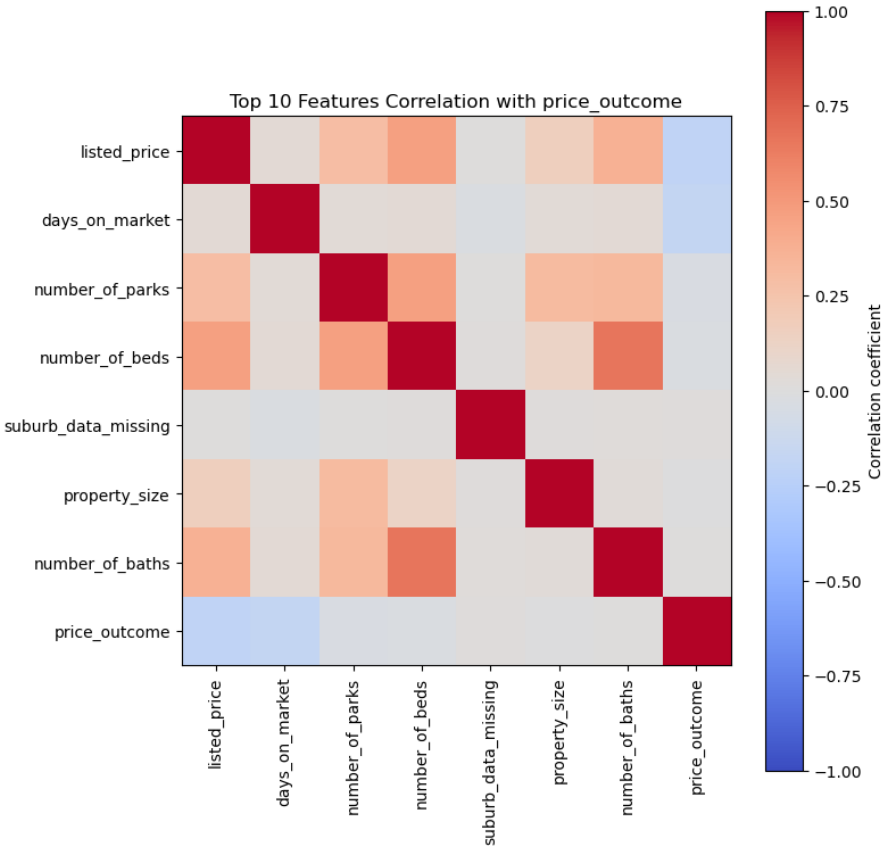

Figure 2: Top 10 feature correlations with price outcome

1.3 Text Preparation and Feature Generation

Part 1.3 of the Jupyter Notebook shows text preparation and feature generation.`property_state` and `property_address` were removed due to redundancy and uniqueness, as suburb information was retained. High-cardinality categorical variables (`property_suburb`) utilized frequency encoding to preserve informational value without dimensional explosion, while columns of lower cardinality were handled using one-hot encoding.

Textual analysis of `listing_description` combined foundational techniques from lectures with advanced methods from external research. The pipeline included TF-IDF vectorization with singular value decomposition (SVD) for dimensionality reduction and sentiment analysis using VADER.

Features generated from external research aimed to be domain-specific features relevant to real estate pricing. These include date handling to find season and month listed along with days since the start of the dataset. These date-based features aim to capture seasonality and shifting market trends. Other features generated from `listing_description` are description word count,

nine domain-specific keyword flags identified through real estate market analysis, emotional intensity and readability score. These features aim to capture how the description of the properties can affect price outcome by capturing physiological and emotional effects along with common words.

1.4 Construction of Model Inputs and Outputs

For modeling, the target `price_outcome` was stored in y, while X included all other variables except `listing_description`, which had been transformed into derived features. A secondary matrix, X_1, excluded features engineered from external research to evaluate their contribution to model performance. A fixed random seed of 30 was set to ensure reproducibility for any procedures involving randomness.

2. Model Building & Evaluation

2.1 Model Development and Performance

Three distinct machine learning architectures were implemented to address the binary classification challenge, selected for their complementary strengths in handling many columns and providing business interpretability. Table 1 presents performance results from 5-fold cross-validation using optimized hyperparameters (see part 2.2) for each model.

Table 1: Model Performance Characteristics

Metric	Baseline Model	Decision Tree	Random Forest	SVM
Accuracy	0.669	0.737	0.787	0.702
Recall	0.869	0.797	0.931	0.887
Precision	0.693	0.792	0.778	0.714
F1 Score	0.769	0.794	0.847	0.791
ROC-AUC	0.677	0.725	0.847	0.742

Decision Tree served as both an interpretable baseline model with a max depth of four, and a model with optimal parameters, chosen for its transparent decision-making process that reveals key pricing patterns. Compared to the baseline model, the optimal model improves F1 score from 76.9% to 79.4%, while maintaining high interpretability. This makes it well-suited for real estate pricing outcomes that require transparent reasoning behind pricing predictions. Figure 3 shows the baseline model's decision path, offering clarity on how features like `days_on_market` and `listed_price` influence predictions. However, Decision Trees with large depths tend to overfit and may generalize poorly when faced with rare property types.

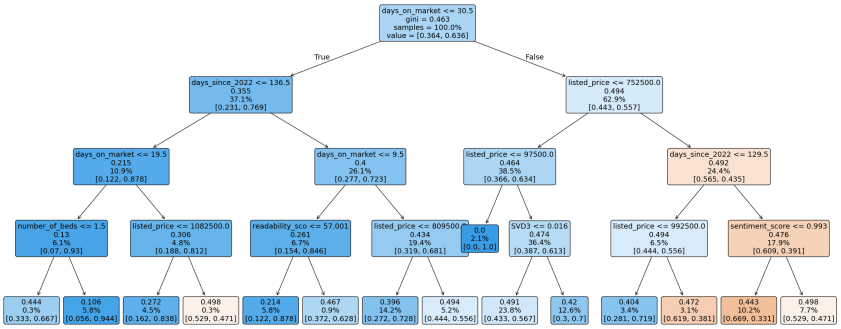

Figure 3: Baseline Decision Tree visualization

Random Forest was selected for its ensemble learning capabilities and robustness to overfitting, particularly valuable when combining numerical and text features. It demonstrates the best overall performance, achieving an F1 score of 84.7% and ROC-AUC of 84.7%, making it the most reliable model. This ensemble method benefits from averaging multiple trees, reducing variance while preserving important patterns. While training time increased severely compared to Decision Trees (1 min vs 8 seconds), this is still considered fast. In a business setting, Random Forest is suitable for automating pricing insights across large property portfolios, with feature importance analysis aiding managerial interpretation.

Support Vector Machines (SVM) with radial basis function kernel were implemented for their effectiveness in high-dimensional spaces, despite longer training times. The model shows an F1 score of 79.1%, similar to Decision Tree, but falls short of Random Forest in both performance and scalability. While it captures complex, non-linear interactions, especially between textual and structured features, its high training time (approx. 2.5 minutes) and low interpretability make it less suitable for business contexts requiring quick, explainable results. Thus, the increased complexity is not justified in this use case.

2.2 Hyperparameter Optimization

To improve model performance while preserving operational efficiency, a grid search was conducted for each model using 5-fold cross-validation with F1 as the scoring metric to provide a balanced view of precision and recall. The optimal hyperparameters are seen in table 2 below. By including more hyperparameters, the scores could have increased at the risk of becoming overfit. To keep runtime low, few but reasonable parameters were tested. Beyond hyperparameter tuning, performance was also improved by refining the feature set (see part 2.4).

Table 2: Optimal Hyperparameters from Grid Search

Model	**Optimal Parameters**
Decision Tree	max_depth=16, min_samples_split=2
Random Forest	n_estimators=200, max_depth=12, min_samples_split=5
SVM	C=1, kernel=rbf

Decision Tree Parameters and Improvement over Baseline Model

The Decision Tree model was tuned to enhance performance beyond the baseline configuration, which used default parameters. While the baseline model achieved a reasonable F1 score of 76.9% and ROC-AUC of 67.7%, its shallow configuration (`max_depth=4`) limited the model's ability to learn deeper relationships in the data, likely reducing its generalization performance across more complex or less frequent property types.

To address this, values of [4, 8, 12, 16] for `max_depth`, were tested to control tree complexity and reduce overfitting while still allowing the model to capture non-linear price patterns. Simultaneously, `min_samples_split` values of [2, 5, 10] ensured decision nodes were formed only when supported by sufficient data, critical for smaller or less frequent property groupings.

The optimal parameters allowed the model to detect meaningful hierarchies, such as listed price thresholds, without overfitting to rare or overly specific location patterns. The tuned model subsequently achieved an improved F1 score of 79.4% and a ROC-AUC of 72.5%, demonstrating both improved predictive capability while remaining computationally efficient and highly interpretable.

Random Forest Parameters
The Random Forest was optimized by testing `n_estimators` values of [50, 100, 200], `max_depth` of [4, 8, 12], and `min_samples_split` of [2, 5]. The final model parameters resulted in the highest F1 score (84.7%) and ROC-AUC (84.7%) among all models.

SVM Parameters
For the SVM, the `C` parameter was tested at [0.1, 1, 10] to adjust the regularization boundary, and only the `rbf` kernel was retained after initial testing showed worse results for linear and polynomial alternatives. The final configuration performed well with an F1 score of 79.1% but incurred the highest fit time. This suggests limited operational suitability.

2.3 Text Feature Impact Analysis

Part 2.2.1-2.2.2 in the Jupyter Notebook shows two instances of a Random Forest model with the same parameters, but different feature inputs, X and X1, as discussed in part 1.4. The scores are seen in the table below:

Table 3: Feature Set Performance Comparison

Metric	All Features	Lecture Features	Improvement
Accuracy	0.787	0.759	+3.7%
Recall	0.931	0.925	+0.6%
Precision	0.778	0.752	+3.4%
F1	0.847	0.830	+2.1%
ROC-AUC	0.847	0.818	+3.5%

Table 3 illustrates that the additional features of X compared to X1 gave substantial performance

gains across all metrics (+2.1% for F1). The Random Forest's feature importance revealed critical predictors of price outcomes:

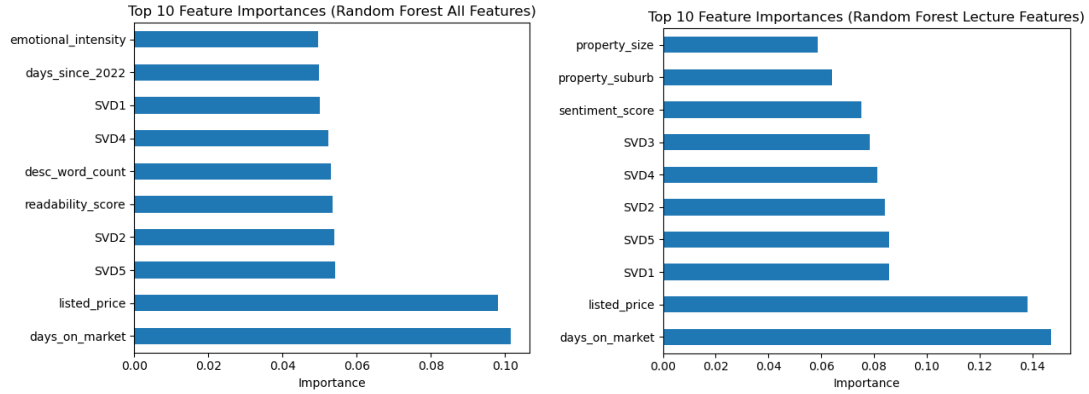

Figure 4: Feature importance for Random Forest models: All features (left) vs features learned to generate from lectures (right)

As seen in figure 4, some of the most important features that contributed to the increased scores, were indeed the features generated from external research such as `emotional_intensity` (5% importance), `days_since_2022` (5% importance), `desc_word_count` (5.5% importance) and `readability_score` (5.5% importance). These domain-specific features, along with date encoding and keywords like "renovated" or "luxury", helped the model capture nuances beyond structured numerical fields. This shows that advanced text features provide the model with nuanced insights into buyer psychology, clarity of communication, and market timing, all of which are critical in real estate transactions. Emotional intensity and readability score, for example, capture the persuasive power and accessibility of the listing description, while date-based features account for shifting market conditions.

3. Findings & Conclusion

3.1 Model Performance

The analysis demonstrates that Random Forest emerges as the superior model for price outcome prediction, achieving 84.7% F1 accuracy through its ability to handle complex interactions between numerical and text features, while still being interpretable through feature importance. While the Decision Tree provides valuable interpretability (79.4% F1), its susceptibility to overfitting and worse understanding of complex interactions makes it less suitable for strategic pricing decisions. The SVM's moderate performance (79.1% F1) and high computational cost further confirm Random Forest's operational superiority in real estate applications.

3.2 Business Recommendations

Based on these findings, sellers and real estate agents should be aware of possible strategies to raise the selling price. This includes highlighting keywords, maintaining a high emotional intensity score, and ensuring the description is easily readable and not too long to achieve high readability scores.

On the other hand, buyers should be aware of these tools and realize how much the description can affect the price outcome. If they can avoid this bias, they may avoid overpaying, and can even find good deals below the listed price.

Real estate owners should deploy Random Forest models for portfolio reviews, prioritizing properties with predicted *Lower* outcomes for price adjustments. If they want to explain key price drivers to customers, they should use Decision Tree visualizations.

3.3 Limitations

It is important to acknowledge the limitations of these findings. This dataset is based only on Queensland, and does not contain factors such as photo analysis and macroeconomic indicators like interest rates.

3.4 Final Recommendation

The Random Forest model with full feature integration is recommended for Australian real estate pricing. Its high F1-score (84.7%) and ROC-AUC (84.7%) captures complex patterns, while still allowing to extract feature importance for interpretability and having a reasonable runtime. This balance of accuracy and operational feasibility positions it as the optimal tool for navigating Australia's dynamic property market.