Austin Housing Price Prediction: Model Summary and Justification

Prepared by: Andrew Chen, Grant Treadway, Humphrey Hui, Keenan J Hui, Tejal Meda

To predict housing prices in Austin, we first examined and cleaned the dataset, transforming variables and engineering features that reflect domain knowledge. We converted 'zipcode' to a factor to treat it as a categorical location indicator, and binary fields like 'hasGarage', 'hasSpa', and 'hasView' were also treated as factors. Log transformations were applied to 'livingAreaSqFt', 'latestPrice' and 'lotSizeSqFt + 1' to reduce skewness and avoid division errors.

Feature Engineering

Several meaningful features were engineered:
- age = latest_saleyear - yearBuilt: captures home age.
- sqft_ratio = livingAreaSqFt / (lotSizeSqFt + 1): distinguishes urban vs. suburban homes.
- bath_bed_weighted = numOfBathrooms + 0.5 * numOfBedrooms: gives more weight to bathrooms.
- lot_category: buckets lot size into small, medium, large, and very large.
- Other features included bath_per_bed, age_sqft_interaction, and bath_sqft_ratio.

To avoid data leakage, all transformations were based solely on predictors and not the outcome (latestPrice).

Data Splitting

Instead of a fixed train/test split, we used 5-fold cross-validation for a more robust and reliable estimate of each model's performance.
We tested a wide range of predictor combinations and, through trial and error guided by variable importance measures (%IncMSE, IncNodePurity), we consistently found that the same 14 predictors yielded the best balance of accuracy and interpretability across all models.

Model Evaluation

We trained and compared multiple models on the log-transformed price variable and evaluated their performance using RMSE (converted back to original price scale):

Model	Test RMSE
Bagging (CV)	174.33
Random Forest (CV)	175.05
BART (CV)	182.36
XGBoost (CV)	177.64
Regression Tree Pruned (CV)	218.65

Handling Holdout Zipcodes

During prediction on the holdout dataset, we encountered zipcodes that were not seen during training. To ensure consistency and avoid missing factor levels during inference, we replaced these unseen zipcodes with the most common zipcode from the training set. This prevented prediction errors and allowed the model to generalize better.

Conclusion

Our final model is Bagging, which achieved the strongest RMSE of 174.33. This could have been because bagging models resists overfitting through averaging over multiple training models. Bagging also decreases the variance while preserving low bias.