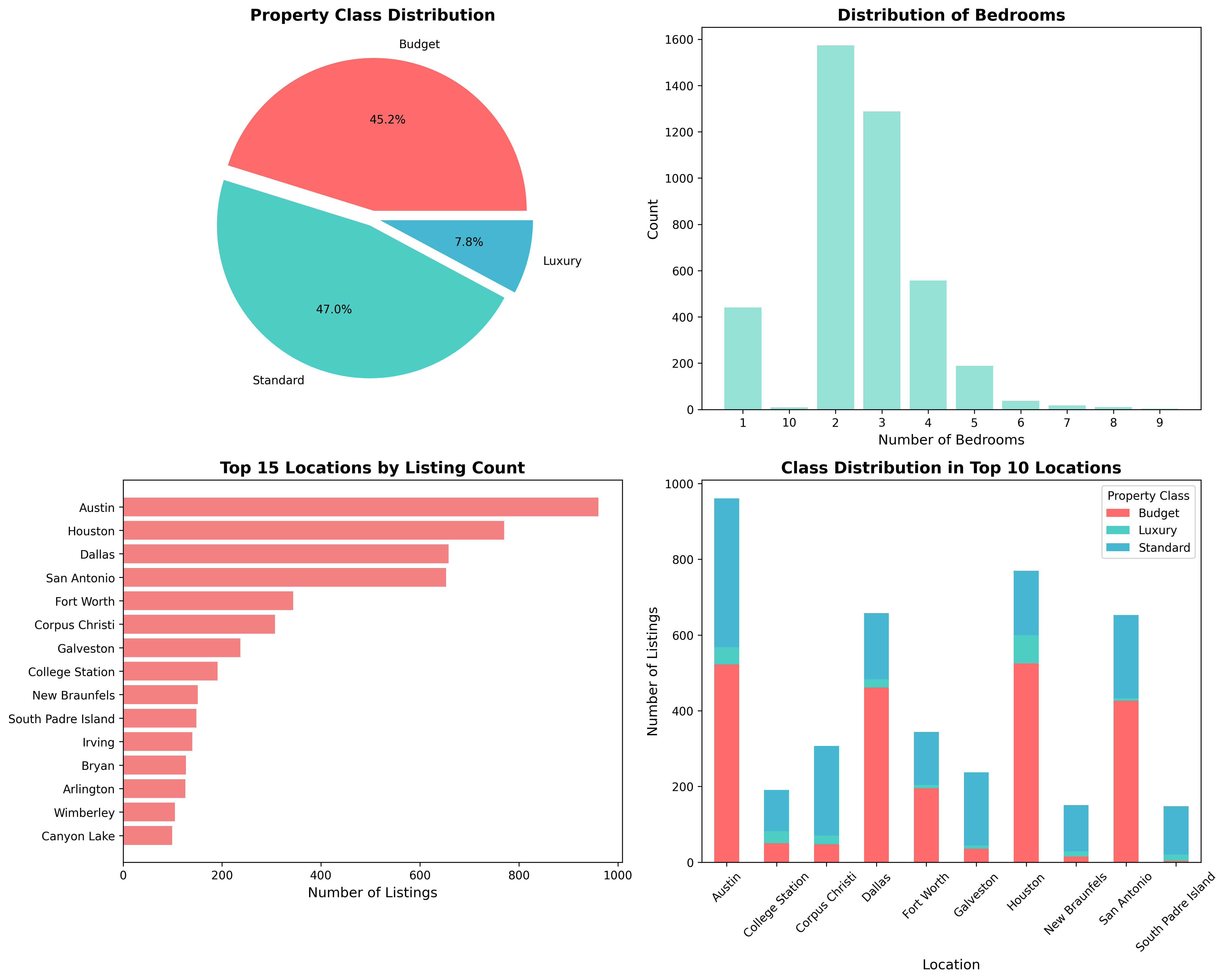
**Property Classification Technical Test - Summary**

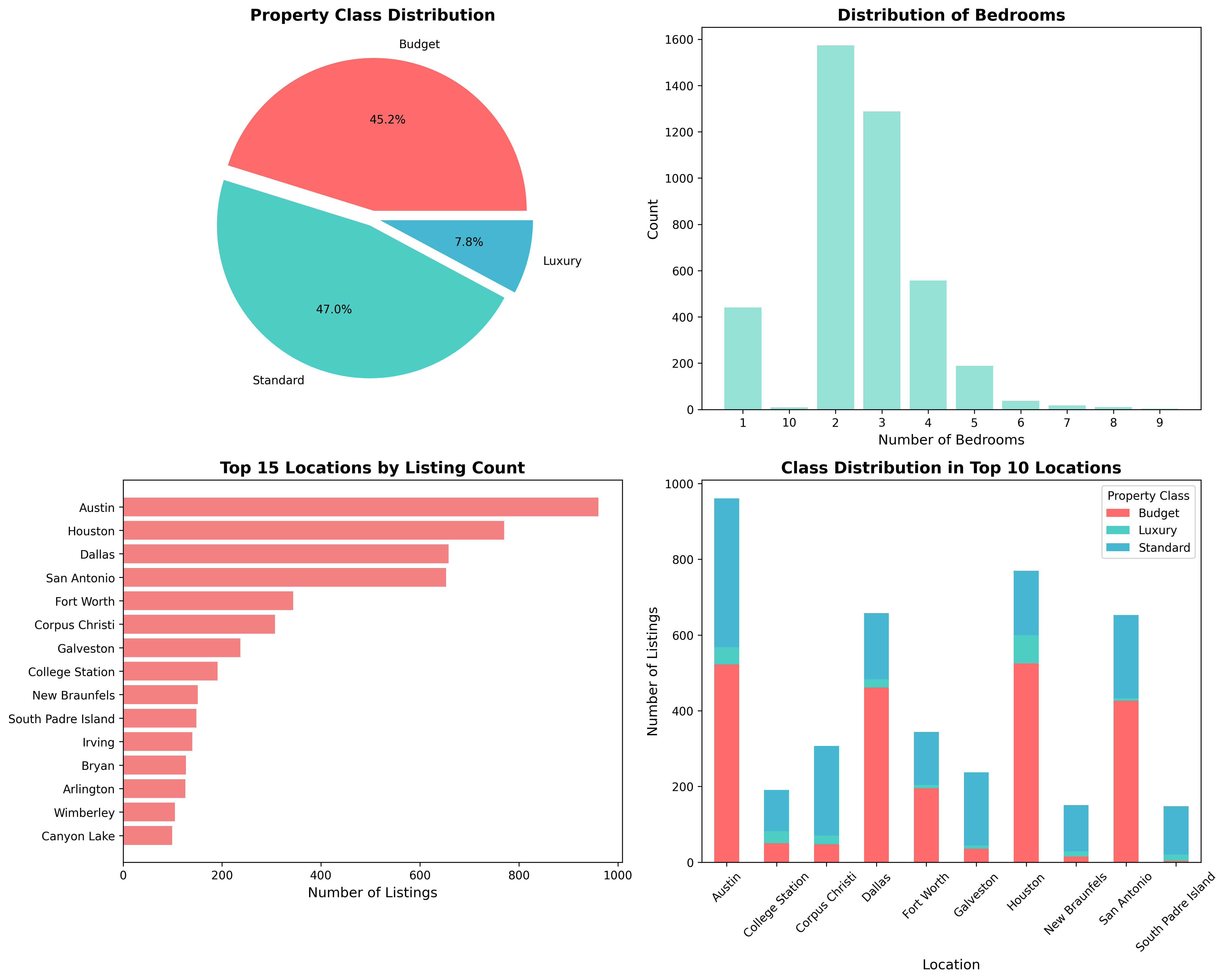
# 1. Exploratory Data Analysis Summary

**Key Findings:**

* **Dataset**: 9,198 Texas property listings which range from December 2008 to December 2016
* **Class Distribution**: Budget (45.2%), Standard (47.0%), Luxury (7.8%)
* **Missing Data**:
  + 730 missing bedroom values (7.9%)
  + 1,608 invalid coordinates (17.5%)
  + "Studio" text in bedroom field which needs to be converted to a number
  + 1 Unknown location
* **Geographic Coverage**: 467 unique locations
* **Top Locations**: Houston, Austin, Dallas and San Antonio have the highest count of listings in the provided dataset

The following graphs have been generated to get an idea of the provided dataset





# 2. Machine Learning Model

**Data Cleaning Approaches**

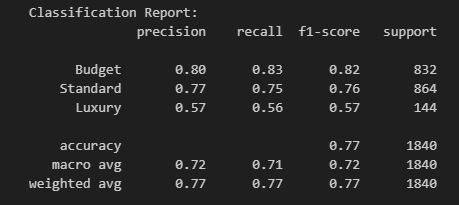
* Set ‘Studio’ Bedrooms to 0
* Used the Median bedroom count to fill all blank bedroom entries
* Updated all -9999 latitude and longitude coordinates to the average latitudes and longitudes as per the locations provided
* Updated all -9999 coordinates with the overall mean (if not populated from the step before)

**Feature Engineering**

* Created the following features to assist the model
  + Luxury words – counting luxury words
  + Budget word - counting budget words
  + Location popularity – counting location appearance
  + Listing month, Listing year – from splitting the Date value into numerical data

**Model Choice**: Random Forest Classifier due to the different types of data. The Random Forest classifier also can provide probability estimates per class.

**Technical Model Performance**: 77% accuracy on validation set from an f1 score perspective. Further details can be found in the figure below in the classification report. While the model does have a low ‘Luxury’ f1-score, the reasoning could be due to the median bedrooms used which (discussed below) significantly influences the models’ decision



**Top 10 Prediction Feature Importance:**

1. Bedrooms: 31.79%
2. Latitude: 12.83%
3. Longitude: 11.13%
4. Location Popularity: 7.96%
5. Description Length: 6.96%
6. Description Word Count: 6.16%
7. Title Length: 5.32%
8. Location (Encoded): 5.00%
9. Listing Month: 4.41%
10. Title Word Count: 3.22%

**For Non-Technical Stakeholders**: The model behaves as a real estate agent who looks at multiple property features and estimates the likelihood it's Budget, Standard, or Luxury. It's correct 77% of the time, which is quite reliable. For each property, you'll get three percentages that add up to 100%, the percentages show the models confidence in how likely it thinks the property will be for one of the 3 classes – Budget, Standard or Luxury.

This can be applied to unknown properties to determine if they would be classified as a budget, standard or

# 3. College Station Analysis

**Answer: YES, College Station has significantly more luxury properties**

* College Station: 16.15% luxury
* Rest of Texas: 7.66% luxury
* Statistical significance: p-value < 0.001 (extremely significant)
* This means the difference is real and not due to chance

The reason for College Station having significantly more luxury properties could be due to the University. Since it is a university town – there may be a lot of properties that would have multiple rooms to act as student accommodations. This would also mean extensive renovations would be done to these properties to attract students for lodging

# 4. Future Enhancements

The following future enhancements to the model can be considered to improve overall accuracy

* Hyper Parameter tuning – this was applied with a first iteration attempt
* Better imputing process of bedrooms to not just use the overall median but to rather consider imputing based of location median as some areas could have larger plots of land (town vs city) and that could influence the bedroom count which is a strong key feature for the model
* When determining bedrooms, first split the training data into the core training data and validation data to not skew results instead of only splitting for training the model

# 5. Technical Appendix

**Code Requirements:**

* Python 3.7+
* Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn, scipy

**Model Parameters:**

* Algorithm: Random Forest
* Trees: 200
* Max depth: 15
* Class weights: Balanced
* Random state: 55 (for reproducibility)