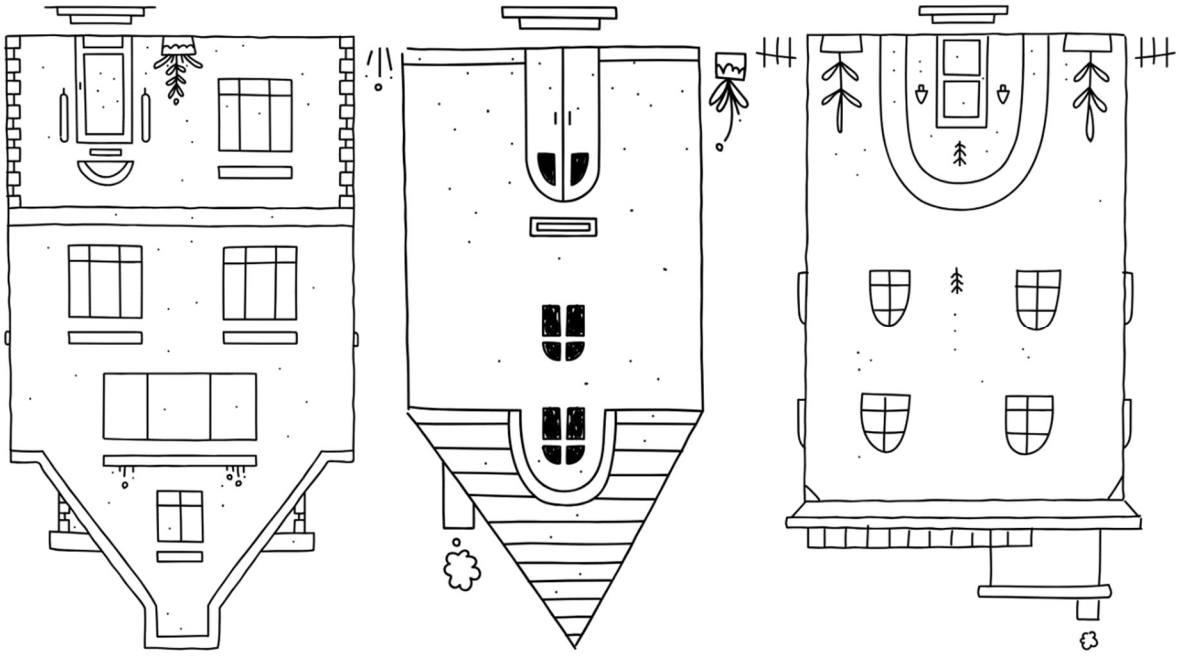
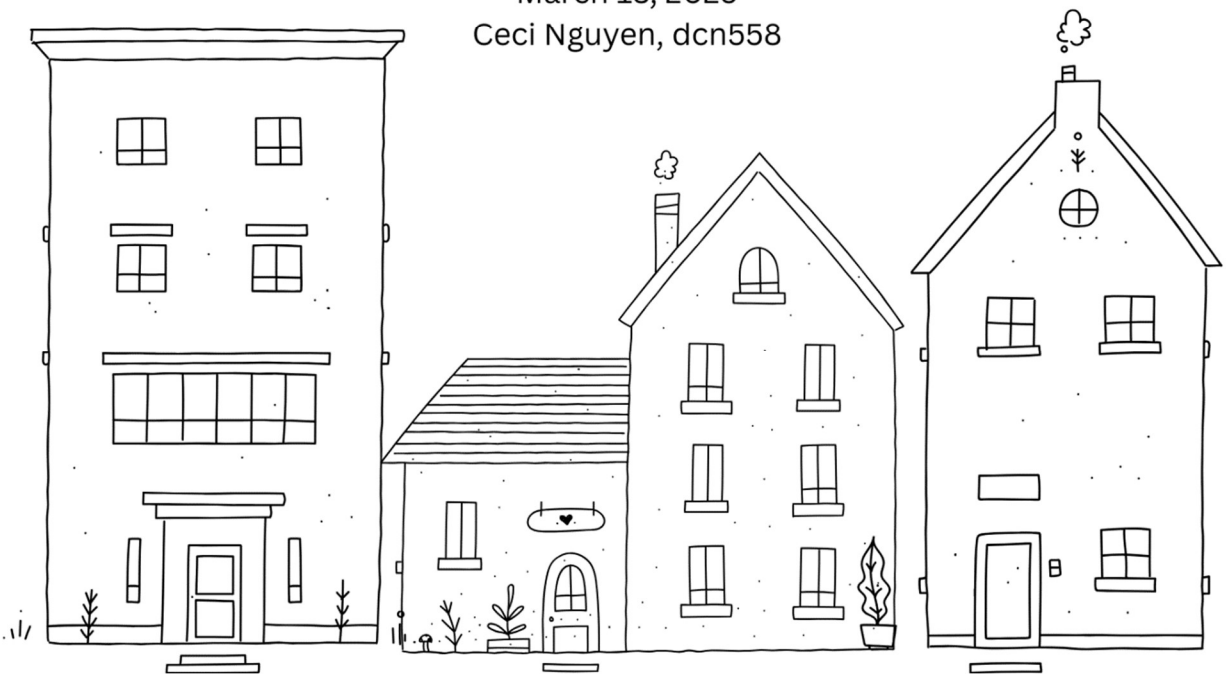


GOLDEN STATE GRIDS

Data Science in Housing Trends

Project 2, COE379L
March 13, 2025
Ceci Nguyen, dcn558



Exploratory Data Analysis

To begin, we examined the shape and size of the dataset, revealing that it is quite extensive. Next, we analyzed the data types of each variable and found that all are in float64 format, except for price_above_median, which is in int64. Since these data types are appropriate for analysis, no conversions were needed. Additionally, we checked for duplicate rows and found none, confirming that our dataset is clean.

A statistical summary of the dataset provided key insights into the distributions of various features:

Features	Key insights
Median Income (MedInc)	The mean and median are close, suggesting a relatively normal distribution. However, the maximum value of 15.00 is significantly higher than the mean of 3.87, indicating some high-income areas.
House Age (HouseAge)	The oldest houses are 52 years old, while the median age is around 29 years.
Average Number of Rooms (AveRooms)	The mean number of rooms per household is approximately 5.43, with a maximum value of 141, suggesting some extreme values.
Average Number of Bedrooms (AveBedrms)	This variable follows a similar pattern to AveRooms, but with a mean of 1.09.
Population	The population per block varies widely, with a mean of 1429 but a maximum value of 35682, highlighting potential outliers.
Average Occupancy (AveOccup)	The mean occupancy per household is 3.07, with a maximum of 1243, indicating high-density areas.

Several univariate visualizations were performed, including histograms and box plots. We observed that most features had a right-skewed distribution, indicating that transformations may be necessary for better model performance.

Classification Techniques

The dataset was split into training (80%) and testing (20%) sets while maintaining the proportion of each class in the dependent variable. The K-Nearest Neighbors (KNN), Decision Tree Classifier, Random Forest Classifier, and AdaBoost Classifier models were

implemented. To improve model performance, hyperparameter tuning was conducted using grid search and k-fold cross-validation techniques, with k set to 5. This approach was chosen to ensure a balanced trade-off between computational efficiency and robust performance estimation. Standardization was also applied where necessary to enhance classification accuracy.

The evaluation of each model based on accuracy and AUC-ROC scores provided the following results:

Model	Accuracy	AUC-ROC
KNN	0.8303852677489701	0.9135695786222366
Decision Tree	0.848073661255149	0.9147468830671072
Random Forest	0.8899927307971892	0.958446413742311
AdaBoost	0.86188514659559	0.9355917240640748

Random Forest and the Stacking Model exhibited the best performance, with Random Forest achieving an accuracy of 0.8899 and an AUC-ROC of 0.9584, while the Stacking Model slightly outperformed it in AUC-ROC. This suggests that both models are highly reliable for predicting price_above_median, with Stacking demonstrating a slight edge in capturing overall predictive power. Confusion matrices were generated for all models to assess their classification effectiveness. The results indicated that ensemble learning techniques, particularly Random Forest, provided superior results compared to individual classifiers.

Model Evaluation and Recommendation

Among all models tested, Random Forest emerged as a strong standalone model with high predictive power and a balance between computational efficiency and performance. However, the Stacking Model slightly outperformed Random Forest in AUC-ROC, demonstrating its ability to combine multiple classifiers effectively. For deployment, Random Forest is recommended due to its interpretability, efficiency, and minimal need for additional tuning. Compared to the Stacking Model, Random Forest offers a more straightforward explanation of feature importance, making it easier to interpret. Additionally, it requires less computational power and training time, making it a more practical choice for large-scale applications. Regarding evaluation metrics, AUC-ROC is the most important for this problem since it accounts for both true positive and false positive rates, making it more reliable than accuracy in cases of class imbalance.