IDX Exchange-DS 35

Contents

Introduction

Goal: Predicting California Property Close Prices

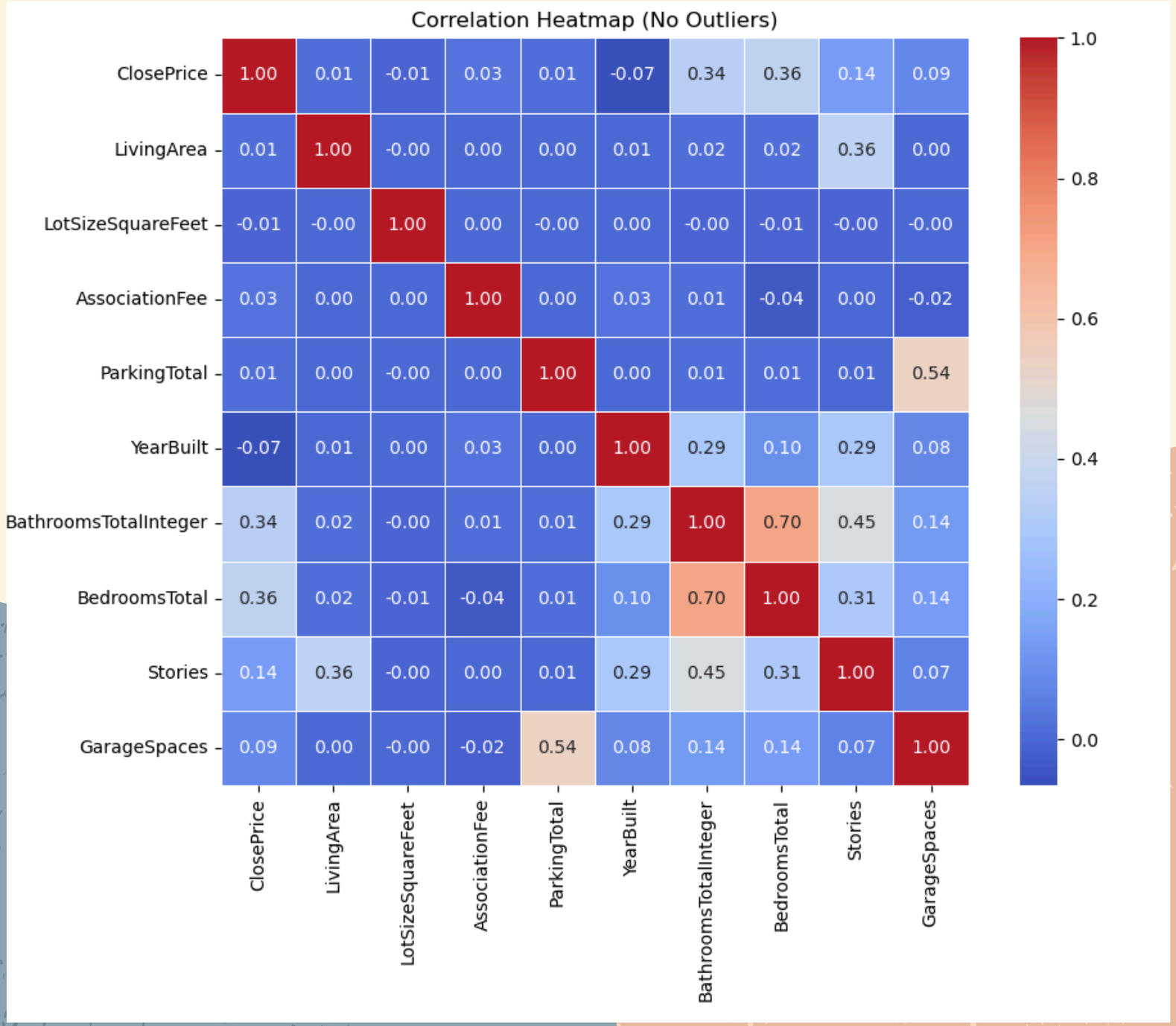
Focus: Single Family Residences & Residential

Data: Historical transaction data from CRMLs

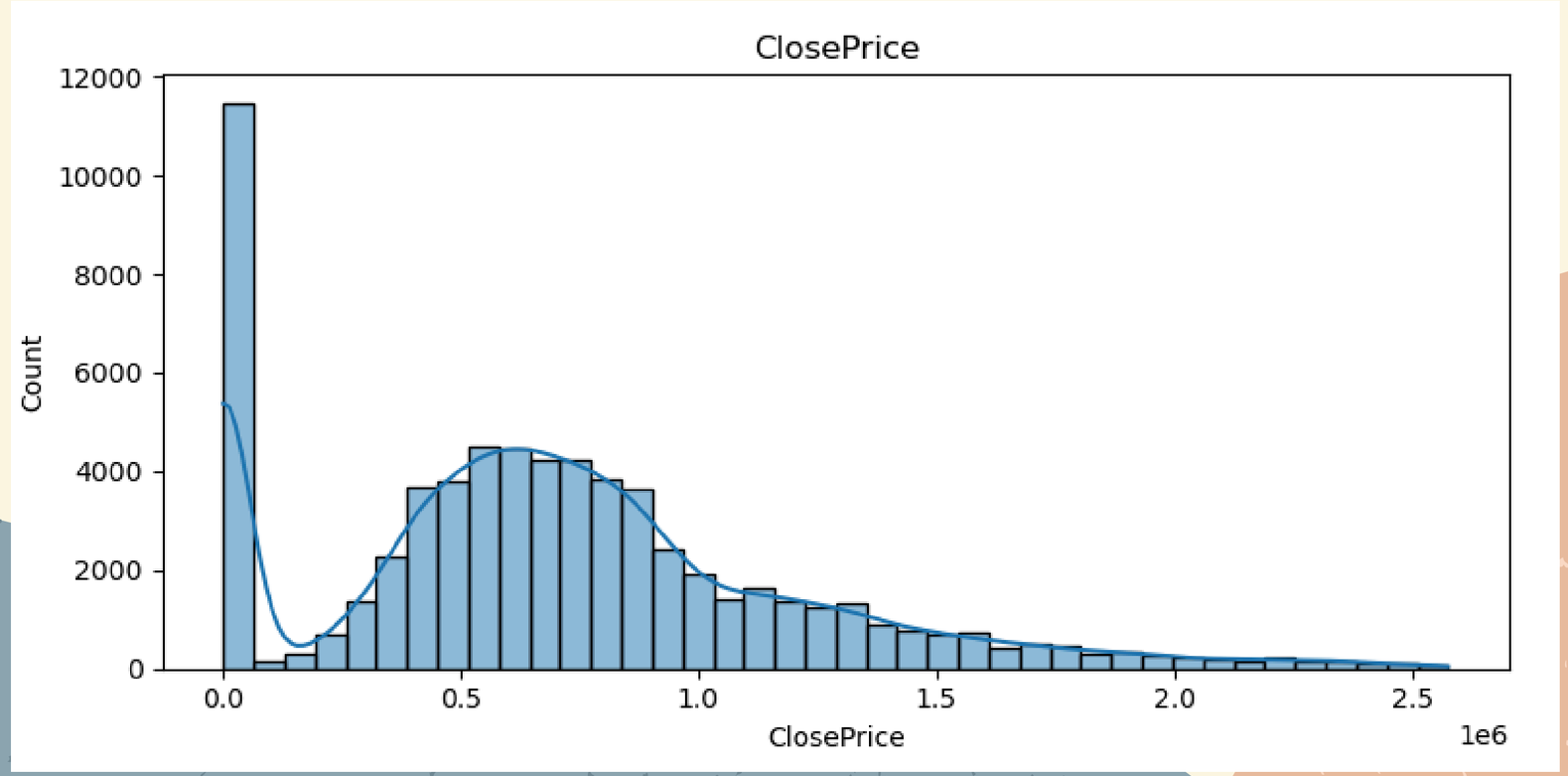
Our Models:

1. LightGBM
2. XGBoost
3. MLP Regression
4. Catboost

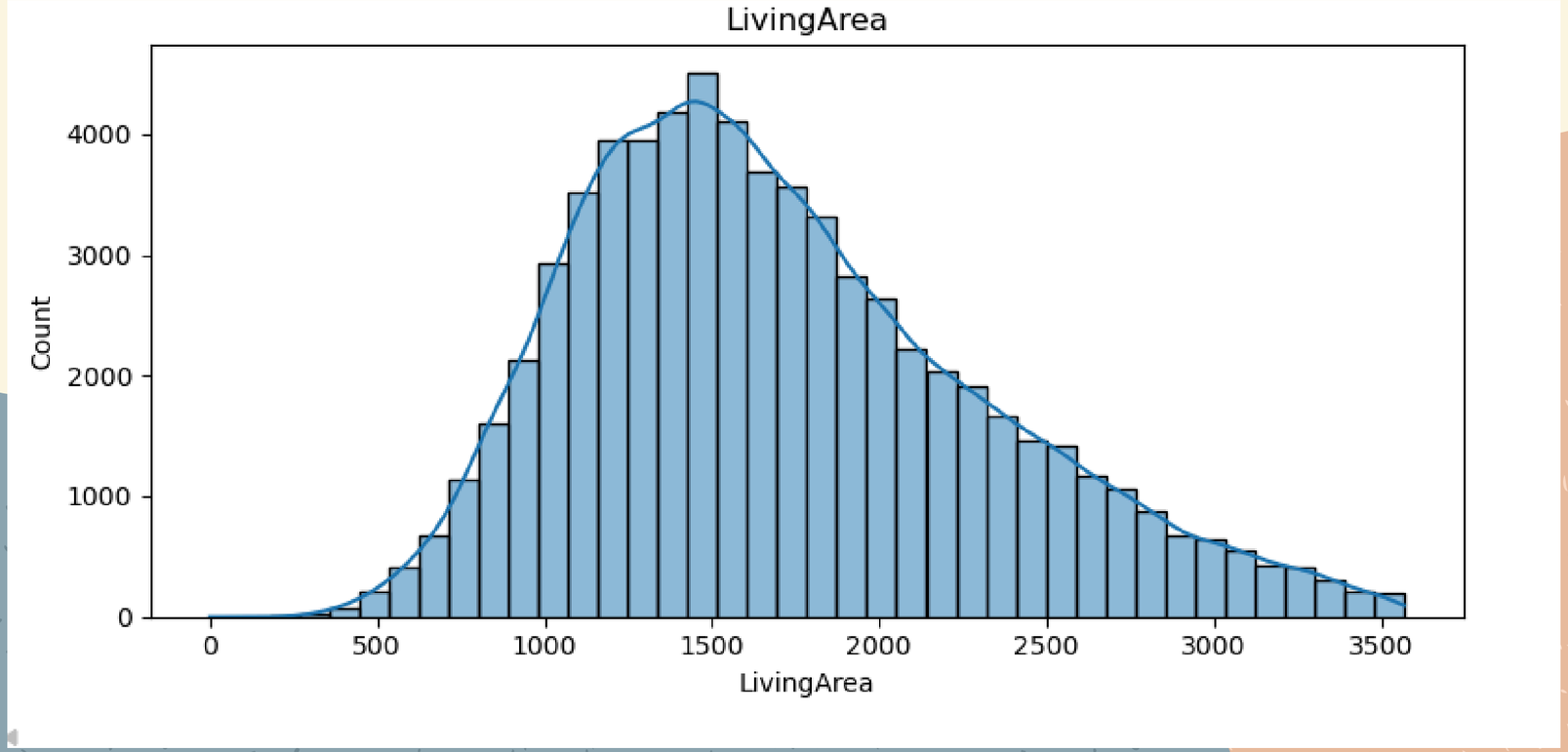
Correlation table



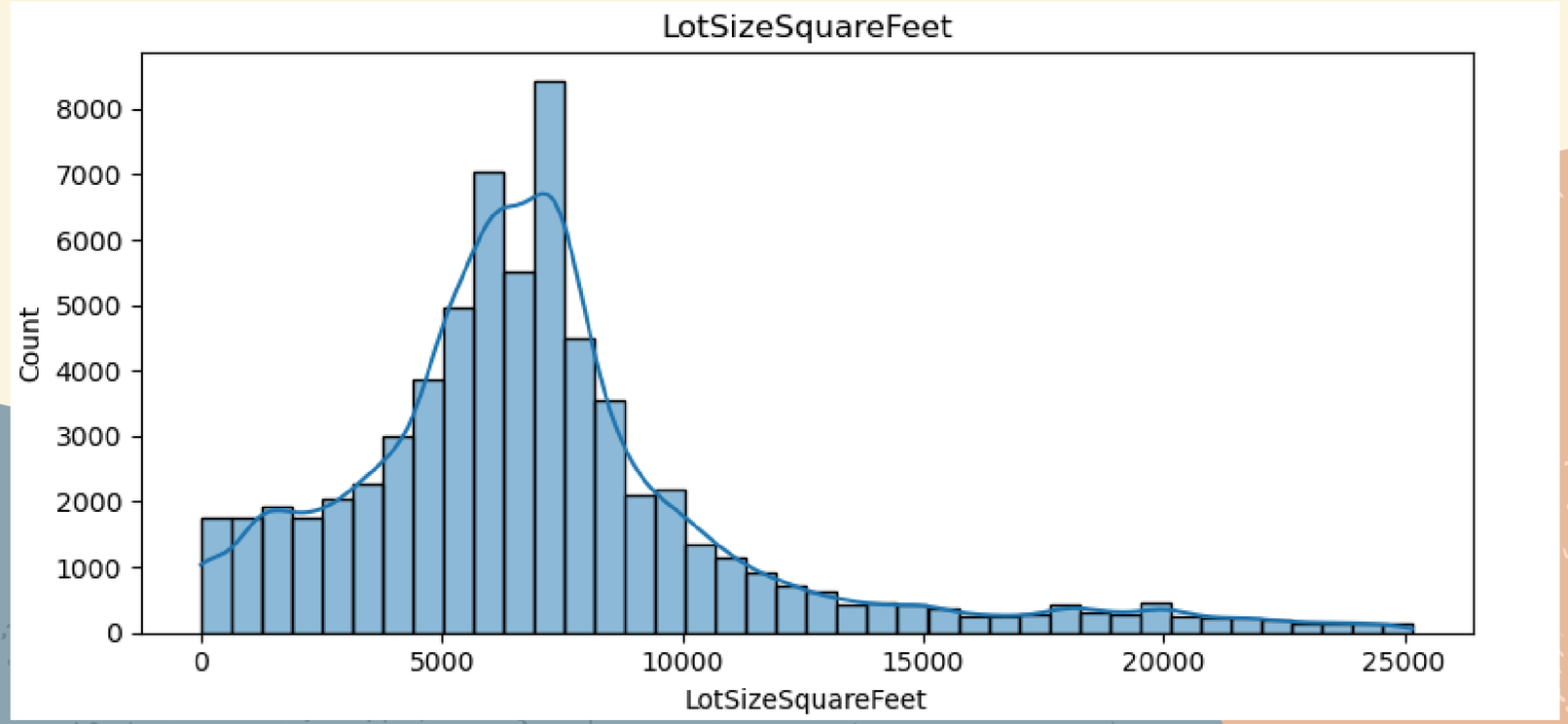
Close price Distribution



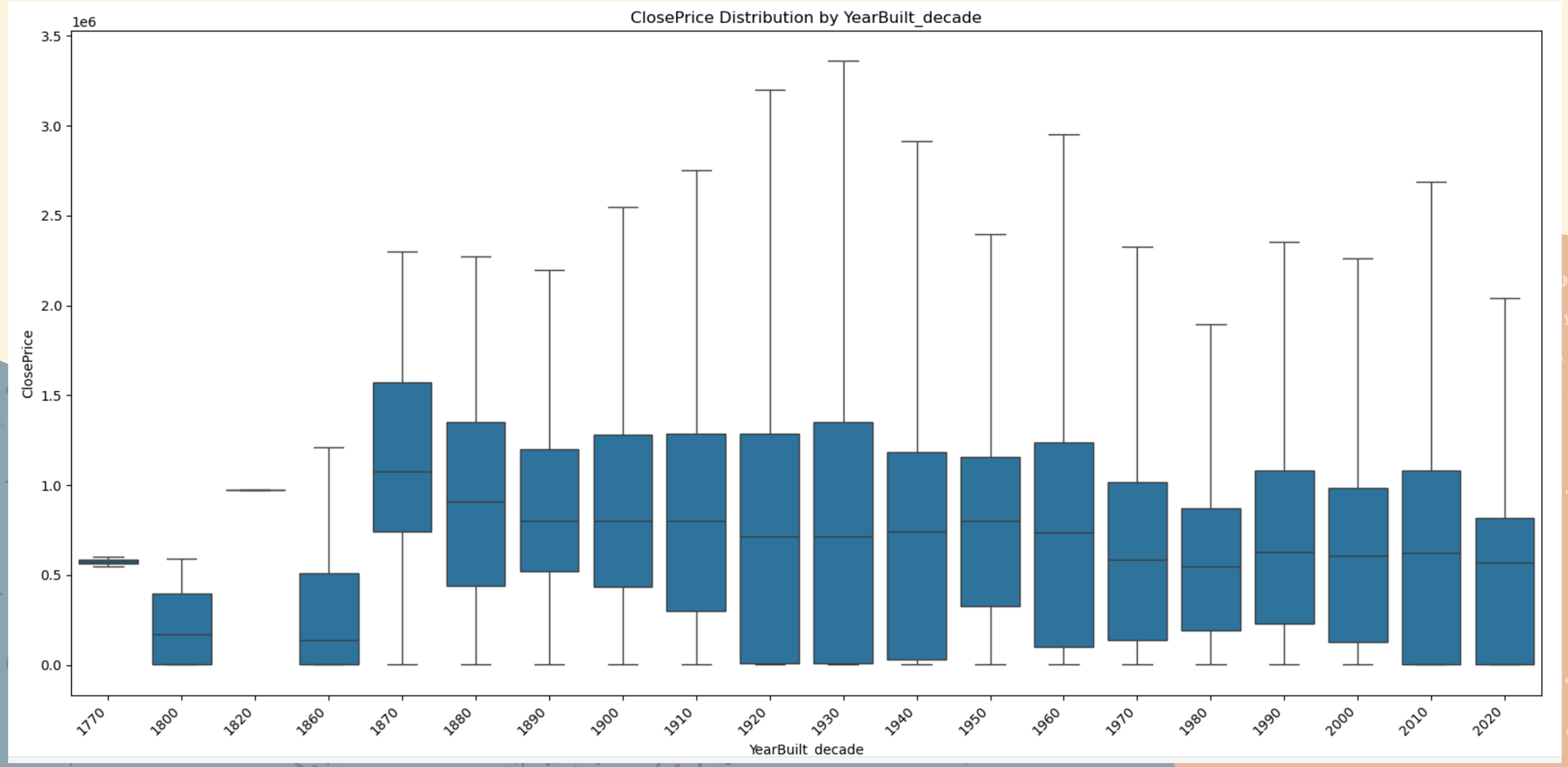
Living Area Distribution



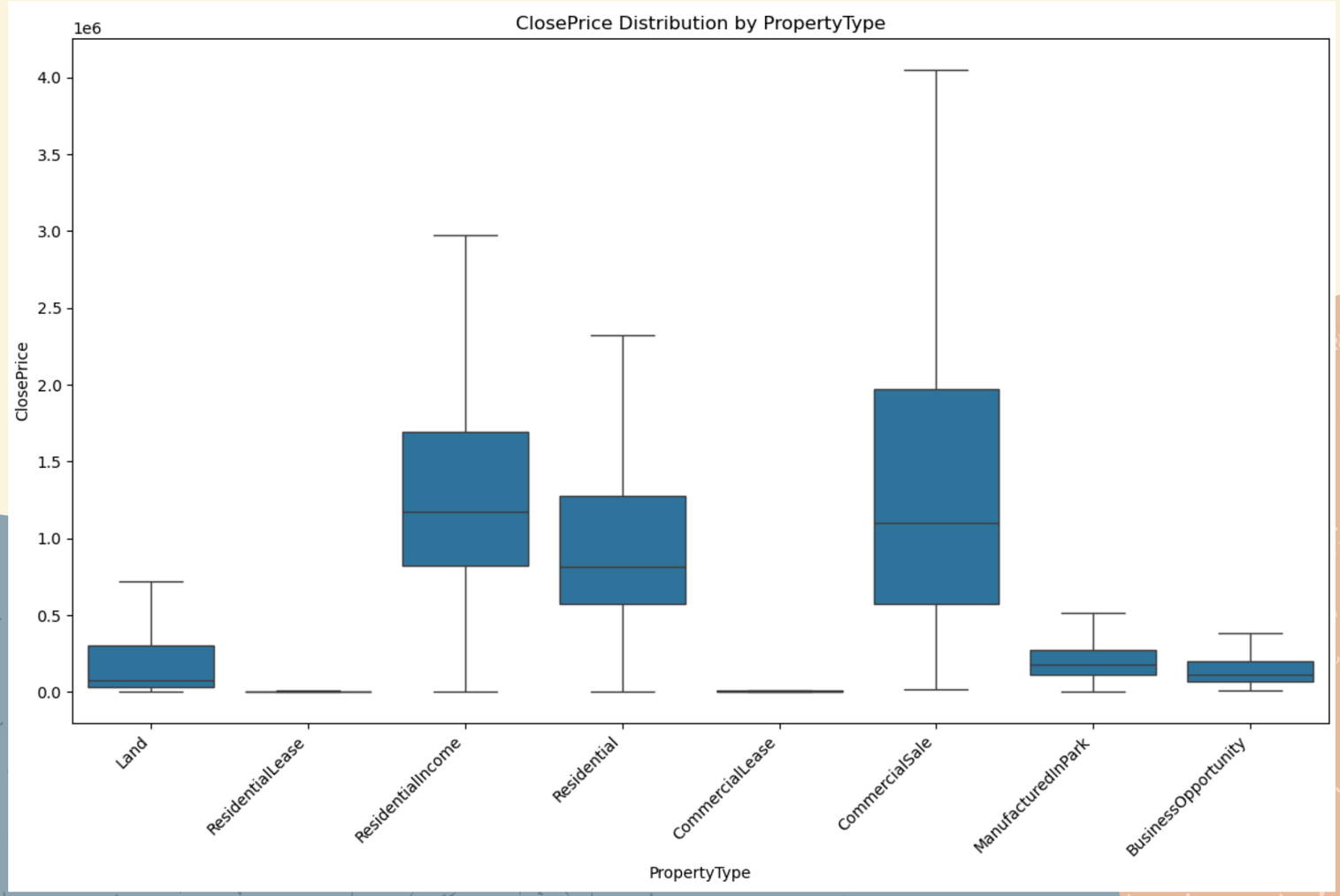
Lot Size (ft²)Distribtuion



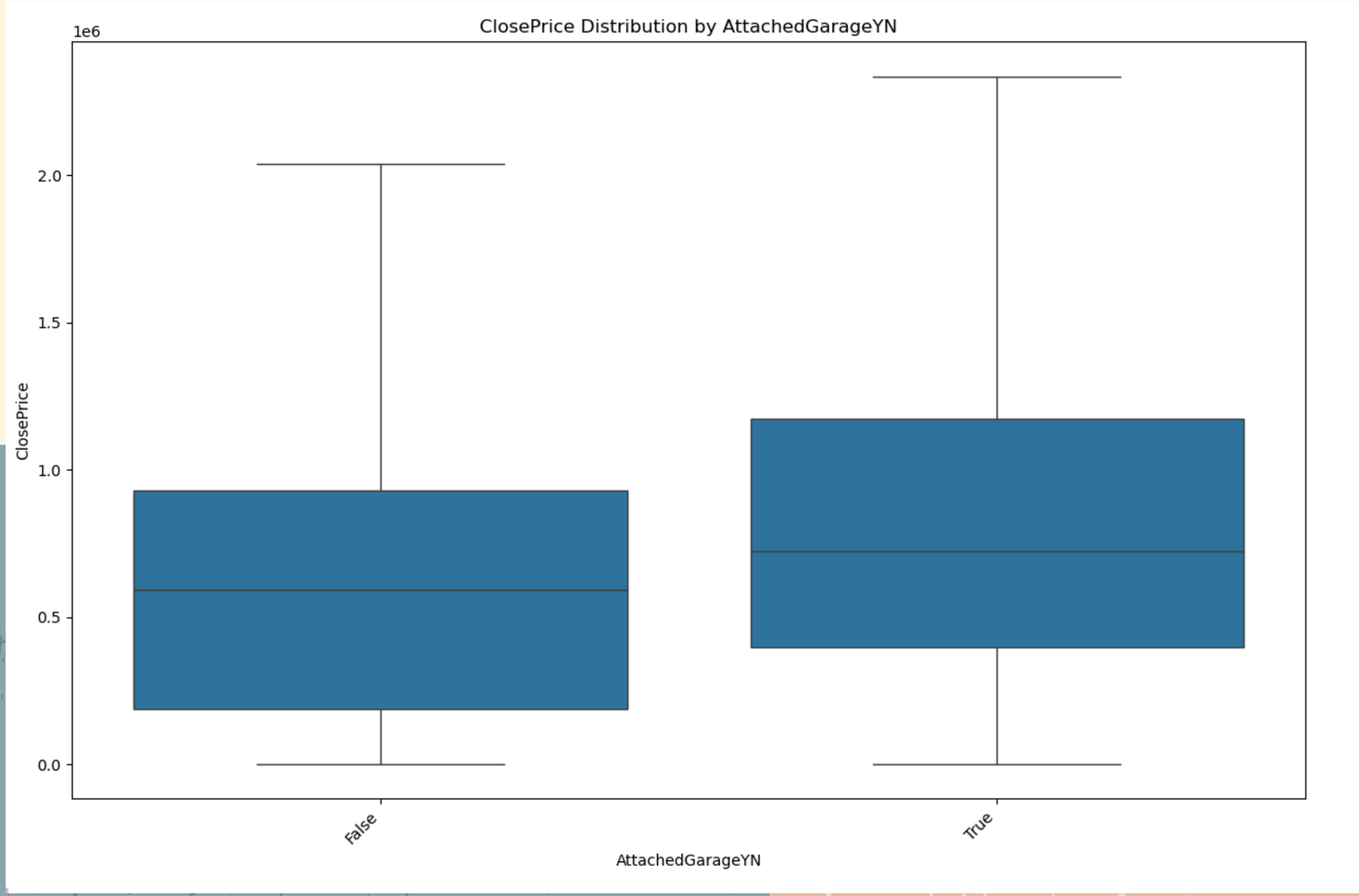
Close Price by YearBuild



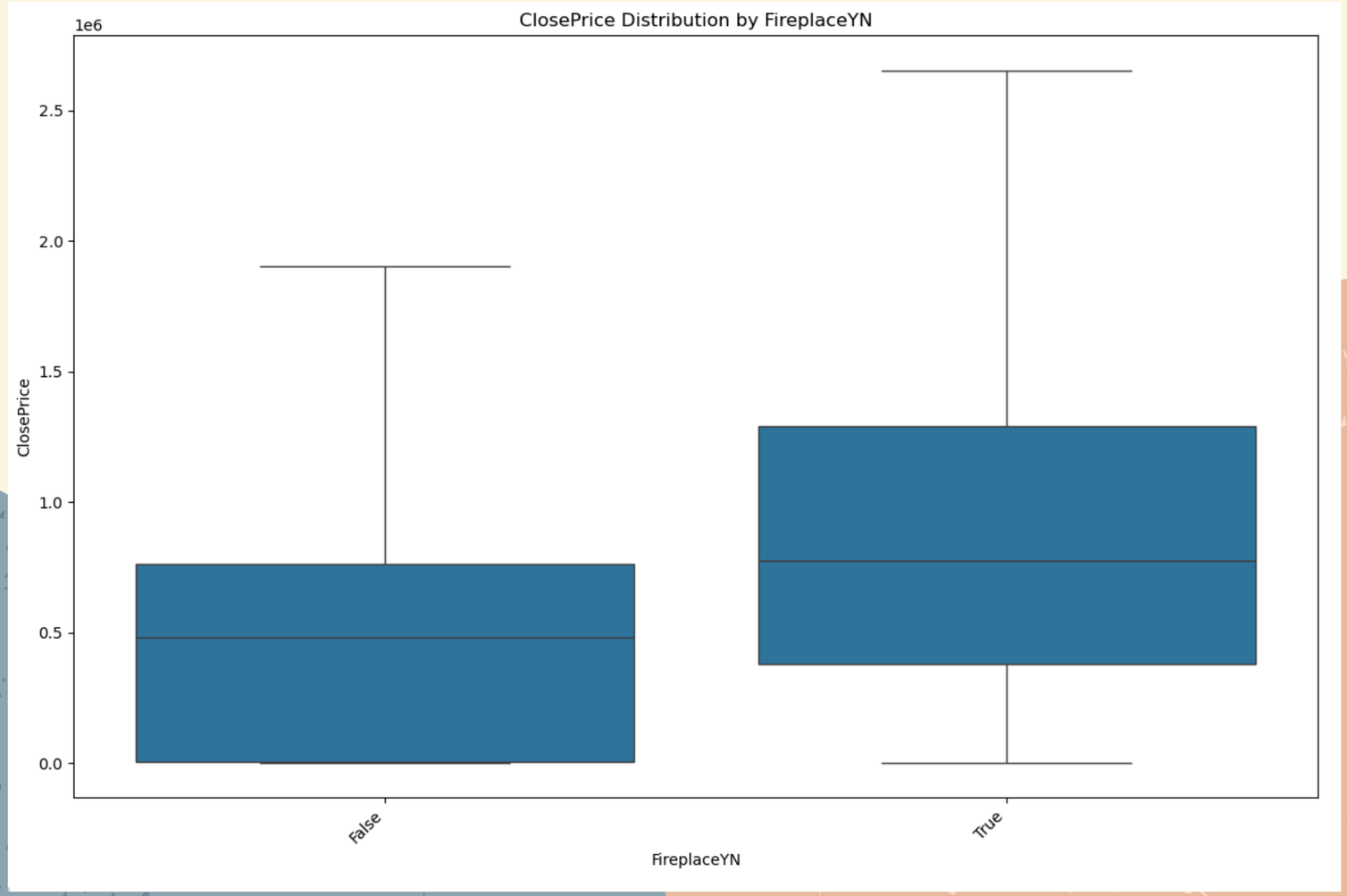
Close Price by Property type



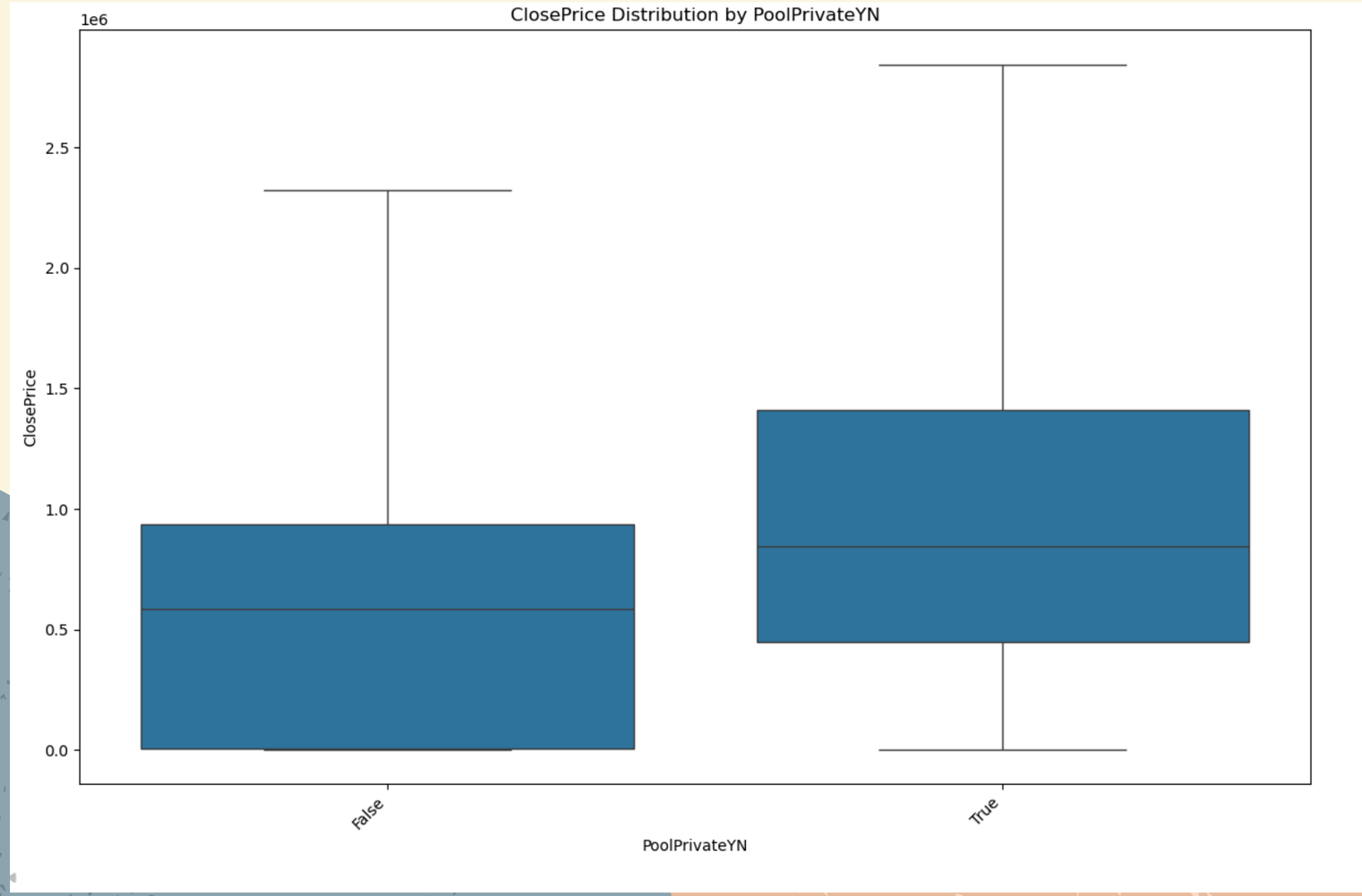
Close Price by Attached Garage



Close Price by Fireplace



Close Price by Private Pool



What is LightGBM?

Overview	How It Works	Advantages
• Light Gradient Boosting Machine • Decision Tree–based Algorithms • High-performance Gradient Boosting Framework • Developed by Microsoft	• Boosting rounds • Small decision trees • Correcting errors • Learning patterns	• Speed • Efficiency • Scalability • Large datasets

Feature Engineering & Spatial Intelligence

KNN Neighborhood Features	Domain-Specific Feature Engineering
Added location intelligence beyond city/ZIP → Used latitude/longitude	Created real-estate ratios
Generated neighborhood price statistics (mean, median, quartiles)	Captured real estate behavior (size, age, discounting, tax burden)
IMPACT:	Boosted R^2 and reduced MAPE

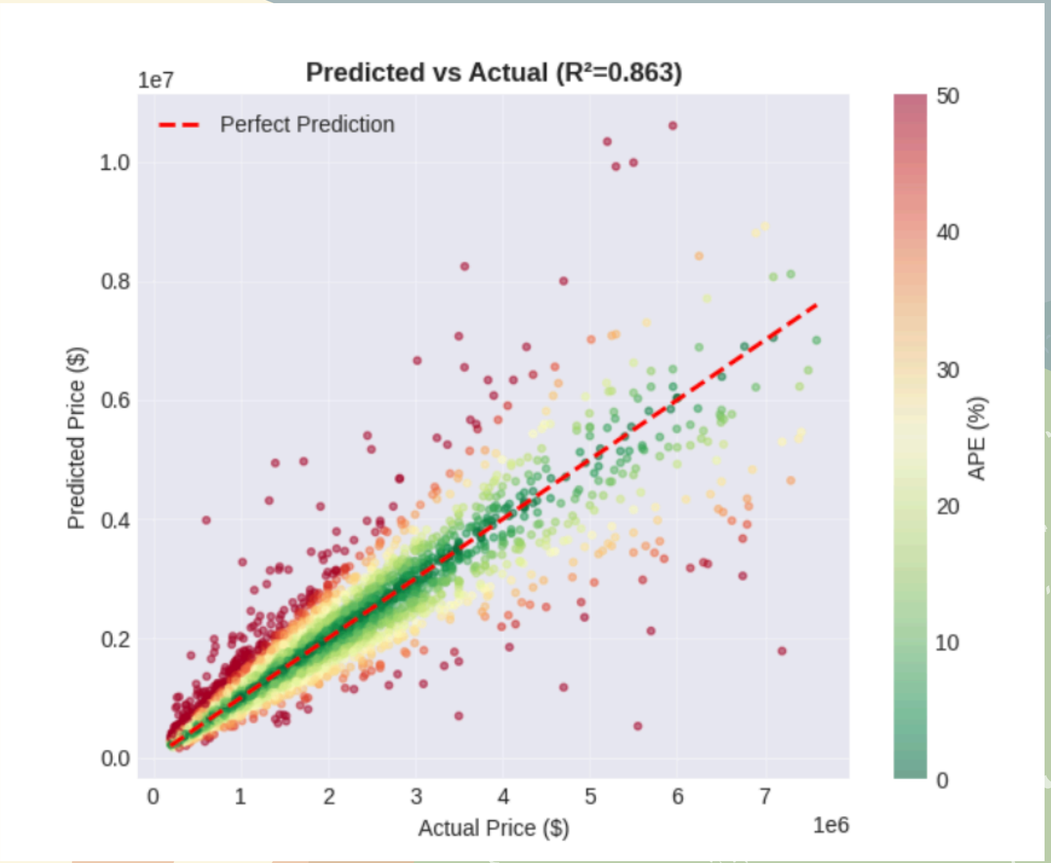
Hyperparameter Tuning with Optuna

- Performed an automated search to minimize validation MAPE
- Used early stopping to prevent overfitting
- Fined-tune key parameters:
  - num_leaves, max_depth, min_child_samples, min_gain_to_split
  - learning_rate, feature_fraction, bagging_fraction
  - reg_alpha, reg_lambda

→ Achieved the lowest MAPE.

Results : Key Metrics

- R^2 = 0.86 → strong signal capture in noisy real-estate data
- MAPE ≈ 12% → reliable valuation accuracy.
- Half of all homes within ~8% error (MdAPE) → model is consistently accurate across most properties, not just on average.



Why chose XGBoost?

Overview	How It Works	Advantages
• Tree-based Gradient Boosting Model • High-performance XGBoost Regressor • Designed for structured/tabular prediction tasks • Widely used in industry and Kaggle competitions	• Ensemble of decision trees built sequentially • Each tree corrects residual errors from previous trees • Uses gradient boosting to optimize loss • L1/L2 regularization to reduce overfitting • Handles mixed features and missing values efficiently	• Strong performance on non-linear, high-dimensional tabular data • Captures complex feature interactions naturally • Robust with engineered features (geo, text, categorical, etc.) • Built-in regularization improves generalization • Provides feature importance for interpretability

Data Wrangling

Removed Extreme Outliers	• Identified unrealistic prices (multi-million anomalies). • Such extremes distort model learning and increase variance. • Applied percentile rules to trim top & bottom extreme values. • Result: more stable, more generalizable XGBoost performance.
KNNImputer for Numerical Features	• Real estate data exhibits local similarity (area, bedrooms, location). • KNNImputer uses nearest neighbors to fill missing numeric values. • Preserves natural variability vs. flattening via mean/median imputation. • More realistic feature distribution → better model quality.
“Missing” Category for Categorical Features	• Categorical missingness can be informative on its own. • Added explicit “Missing” category instead of mode imputation or dropping rows. • Enables model to learn patterns associated with missing values. • Well-aligned with XGBoost’s strong categorical handling.

Feature Engineering

Numeric Scaling

- Numeric features were processed using median imputation to handle missing values.
- After imputation, features were standardized using StandardScaler to normalize scales and reduce variance differences.
- This prepares numeric inputs for consistent handling by the XGBoost model and prevents features with larger ranges from dominating splits.

Text Vectorization

- Textual features (Flooring and Levels) were transformed using TF-IDF vectorization (max 100 features each).
- Converts raw text into numeric vectors representing token importance, allowing the model to leverage descriptive property information effectively.
- Ensures that repeated important terms contribute more to feature representation, enhancing predictive power.

Geospatial Encoding

- Categorical location features (State, County, City) were encoded using hierarchical target encoding, replacing each category with the mean ClosePrice for that group.
- Within each City, properties were grouped using KMeans clustering on (Latitude, Longitude) to generate City_Cluster labels.
- This captures both local pricing patterns and neighborhood-level spatial clusters, helping the model distinguish areas with systematically different prices.

Model Tuning

- Performed automated hyperparameter optimization with Optuna to minimize MdAPE (Median Absolute Percentage Error)
- Integrated early stopping in every trial to prevent overfitting
- Optimized key XGBoost parameters impacting tree structure, sampling, regularization, and learning rate

Achieved the lowest MdAPE.

Basis Results:

- R^2: 0.8038
- MAPE: 22.77%
- MdAPE: 14.97%

Best Results:

- R^2: 0.8111
- MAPE: 22.21%
- MdAPE: 14.36%

Conclusion:

There is no significant improvement. It could be the limitation of the model and the included features.

Next Step:

Exploring Features

Ratios:

- Price per sqft = ClosePrice / LivingArea
- Lot-to-living ratio = LotSizeArea / LivingArea

Further analysis on price by region:

- Include median and variance for each district.

Also, it could be the limitation of explainability on XGBoost.

We can explore on other advanced model as well such as CatBoost and MLP. Or mix models by ensemble.

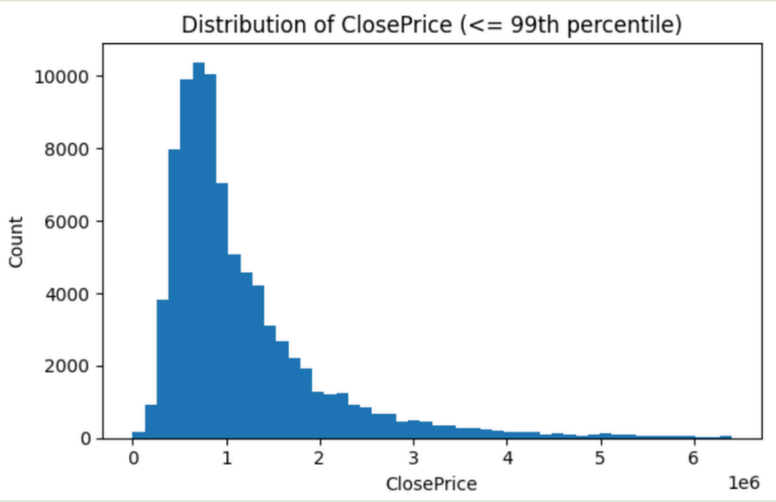
MLP Introduction

What is MLP Regression?
Why choose MLP Regression?

Definition	Advantage	Disadvantage
• Multilayer Perceptron, a kind of **neuron network** with multiple layers • Input → layer 1 → layer 2 → … → output • Nonlinear activation function • Can be used for classification and regression tasks	• Flexible design, suitable for ANY function • Strong accuracy with large datasets • Regression, Classification, Pattern recognition, … • Housing price are often non-linear	• Large data & demanding computation resources • Highly hyper-parameter choices sensitive • “Black-box” model • Local minimum risk

Feature Engineering

Outliers	• Filter both **Price** and **Price Per Square Feet** • **Upper bound**: min(Tukey bound, quantile bound) - stable results • Price < $10000
Feature Building	• **Additional variables**: Property Age, Baths per Bed, Lot per Living, log(Living Area)**2, log(Living Area)*Total Bathrooms • **Out-of-fold target encoding with smoothing**: deal with geographic features (City, Zip, ...) • **Multi-scale RBF** (radial basis function): transfer longitude and latitude variables to distances to related centers • **Out-of-fold KNN**: the impact of prices in neighborhood
Final Input	• Total number of features: **188**



MLP Model

Model structure: Input → Dense(512) → Dense(256) → Dense(128) → Output(1)

Output(1) is **log(Price)**

Model Overview	Training	Post Training
• Dense -> ReLU • BatchNorm -> Dropout -> L2	• Early stopping + LR scheduler	• Smearing Correction
Dense + ReLU brings: 1. Non-linearity; 2. Model stability; 3. Computation efficiency; 4. Approximation to any function. **BatchNorm** - layer normalization, brings: 1. Faster convergence; 2. Overfitting prevention - **layer level**. **Dropout** brings: 1. Overfitting prevention - **neuron** level. **L2** brings: 1. Overfitting prevention - **weight** level.	**Early stopping** brings: 1. Best performance checkpoint; 2. Overfitting prevention. **Learning rate** scheduler brings: 1. Fine-tune around the minimum.	**Smearing correction** brings: 1. Downward bias caused by exponentiation log-predictions

Results

R^2	87.15%
MAPE	12.79%
MdAPE	8.72%

- **The model performs well on the cleaned dataset**
- **BUT …**

Model performance on Full dataset:

R^2	0.32%
MAPE	13.2%
MdAPE	8.83%

Price Segment:

0k - 500k	**R^2**	-0.58
500k-1000k	**R^2**	0.33
1000k-infk	**R^2**	0.75

- **Model Interpretation**

  Can we interpret machine learning models?

  SHAP Method – Complex machine learning models interpretation

  SHapley Additive exPlanations, a game-theoretic approach to explain the output of **any machine learning model** by calculating the **marginal contribution** of a feature to the prediction.

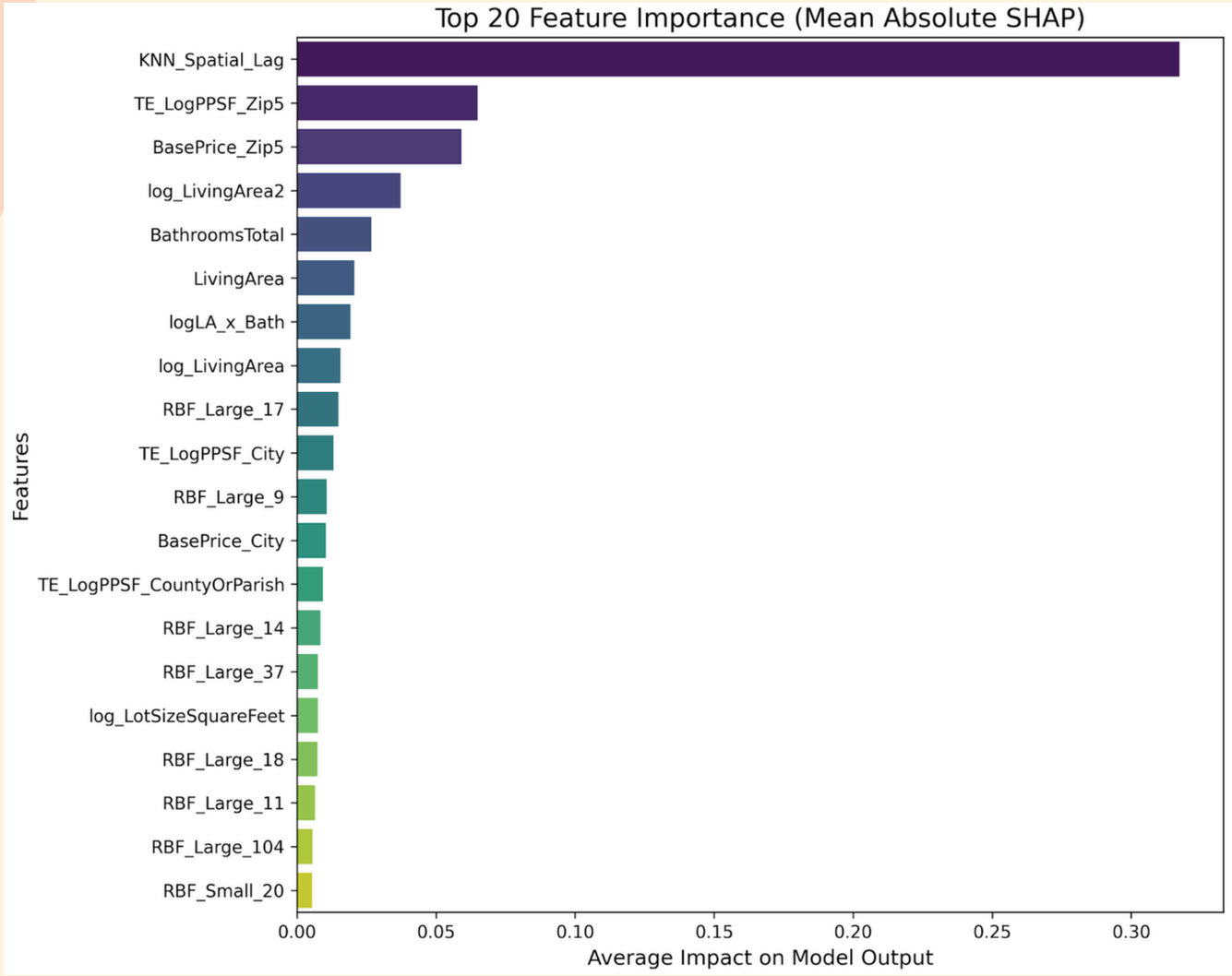
Advantage:

1. Consistency
2. Directionality

Results

This model prioritizes location over physical attributes.

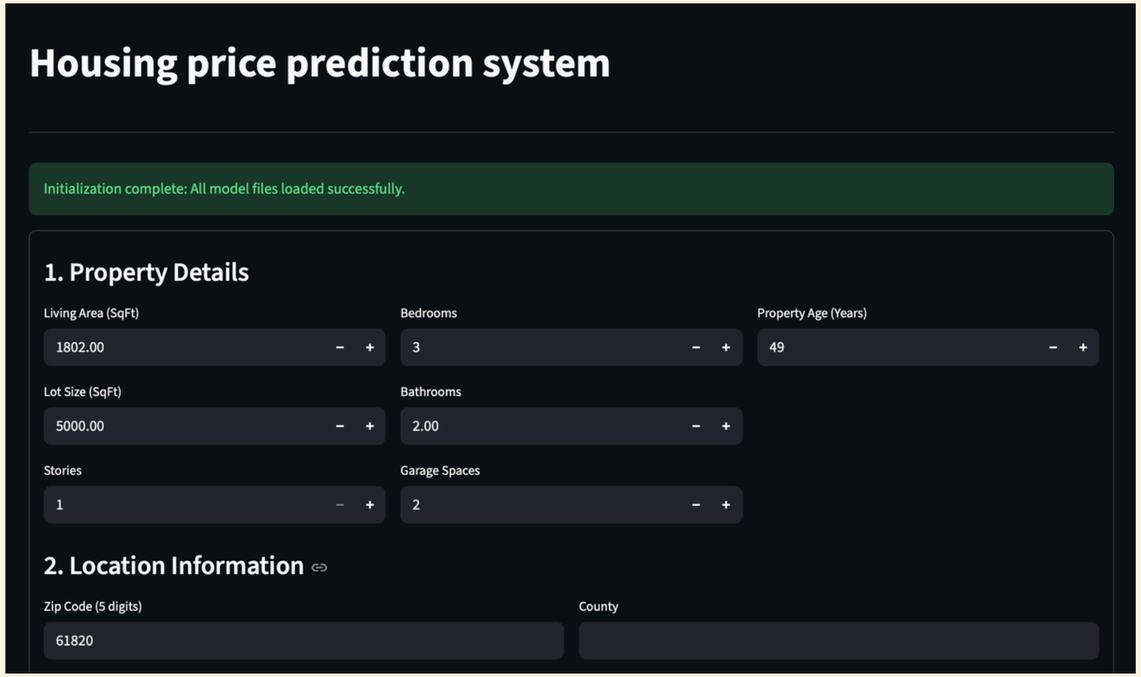
- **Dominant Feature:** Spatial Lag
- **Macro-Location:** Zip Code
- **Physical Attributes:** Living area, Bedrooms, ...



- This is a high performance model on cleaned data and high-end properties but with limitations on generalization.
- Lower-priced homes have higher variance in property condition that out current data doesn't capture.

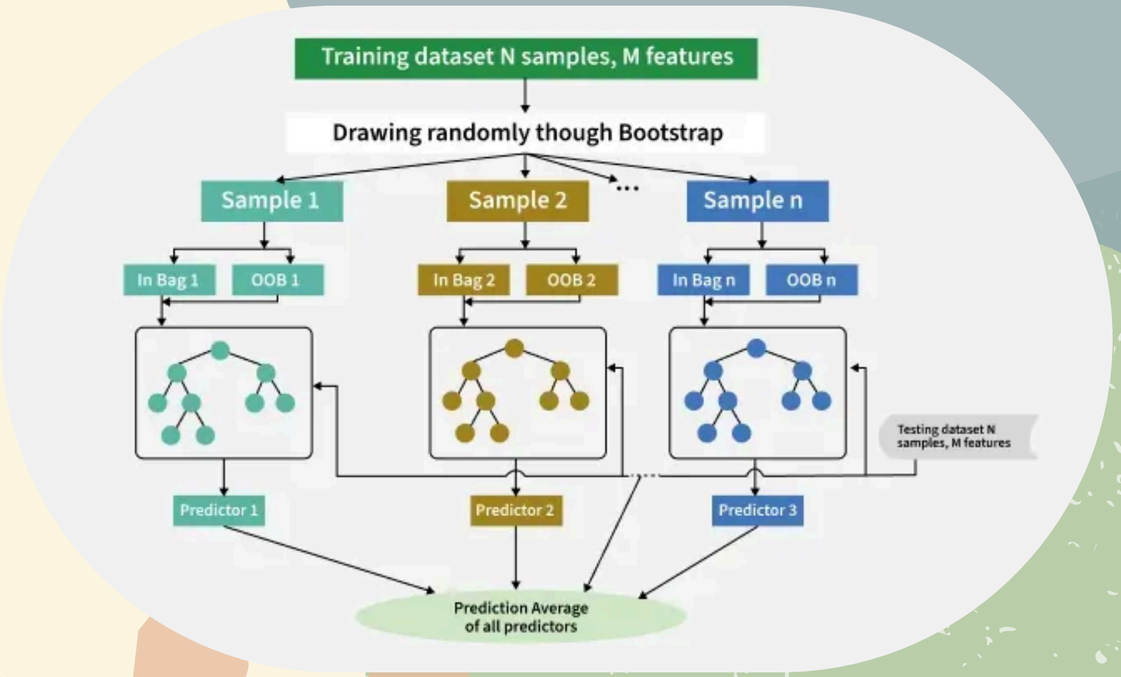
Future steps

- **Segmented modeling:** Develop separate models for the Luxury Segment (>1M) and the Economy Segment (<500k).
- **Unstructured data integration:** Incorporate NLP on listing descriptions or Computer Vision on property photos.
- **Robust loss functions:** Experiment with Huber Loss or Quantile Regression instead of standard MSE.
- **Streamlit App:** A real useable model.



What is CatBoost?

- Gradient Boosting framework by Yandex
- Optimized for tabular data with categorical features
- Handles categorical variables automatically
- Strong performance with minimal preprocessing



Advantages

Automatic Handling of Categorical Features

Symmetric Tree Structure

Supports GPU Training and Missing Values

Disadvantages

Slower on Very Large Datasets

Fewer Hyperparameter Tuning Options

Limited Interpretability

Data Preprocessing

Filtering	Cleaning	Pruning	Target Transformation
• Retained only Residential and SingleFamilyResidence properties	• Converted string-based numeric fields to numeric types	• Removed listing-related fields • Kept only samples with sale prices within the 0.5%–99.5% quantile range	• Applied log1p(ClosePrice) → y_log to reduce skewness in price distribution

Model Iteration

Difference:

- R^2 ↑ 0.62
- MAPE ↓ 6%
- MdAPE ↓ 5%

v1: Baseline

- CatBoost with external data (Income, Schools).
- Result: Poor performance due to missing external data.
  - $R^2 \approx 0.25$
  - MAPE ≈ 17%
  - MdAPE≈ 12%

v2: Spatial

- Engineered "Mean Neighbor Price (20 nearest)".
- Result: Massive accuracy boost.
  - $R^2 \approx 0.86$
  - MAPE ≈ 11%
  - MdAPE≈ 8%

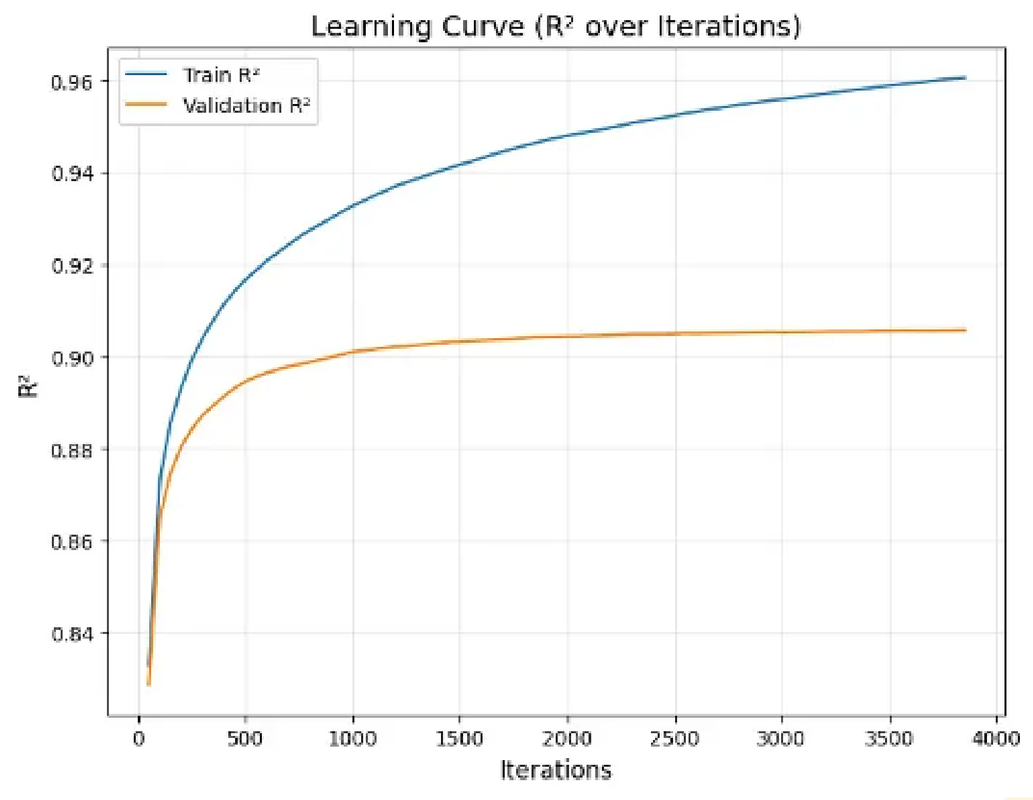
v3: Remote-Aware

- Hybrid approach separating remote vs. urban areas.
- Result: Limited gain due to sparse remote data.

v2.1: Final

- Refined v2 with stricter California-only filtering.
- Result: Most robust performance.
  - $R^2 \approx 0.87$
  - MAPE ≈ 11%
  - MdAPE≈ 7%

Model Diagnostics



Residual Patterns	• Residuals are mostly centered around zero, indicating no strong systematic bias. • High-priced properties exhibit a funnel-shaped pattern, where prediction variance increases with price. • A small number of extreme outliers are observed (approximately ±2M to ±6M), suggesting difficulty capturing rare or highly unique homes.
Learning Curves (RMSE)	Training and validation RMSE decrease together, showing no signs of overfitting.
Learning Curves (R^2)	Validation R^2 increases steadily and remains close to the training curve, indicating good generalization and stable learning behavior.

Next Steps / Future Work

- **Increase historical data for remote areas**
  → Helps reduce errors caused by small sample size and distinct distribution patterns in remote regions.

- **Build a dedicated model for high-priced properties**
  → Since luxury homes have unique characteristics, a specialized model may capture their pricing patterns more accurately.

Models Performance Comparison

(Training: 2025.01–2025.08; Testing: 2025.09)

Model	R^2	MAPE	MdAPE
MLP	87.15%	12.79%	8.72%
LightGBM	85.74%	12.14%	7.75%
XGBoost	81.11%	22.21%	14.36%
CatBoost	86.82%	11.29%	7.71%

Advantages & Disadvantages

Model	Advantages	Disadvantages
MLP	• Powerful nonlinear learning • highly flexible	• Needs large data- easily overfits • low interpretability
LightGBM	• Very fast • scalable • handles missing values	• Sensitive to tuning • weak categorical support • higher overfitting risk
XGBoost	• Strong tabular performance • good regularization • handles missing values	• Heavy training & tuning cost • risk of overfitting with poor features
CatBoost	• Best for categorical data • strong defaults • stable	• Slower on very large data • fewer tuning options

Common Feature Engineering

- **Geospatial / Neighborhood Features**
  KNN + coordinates to create neighborhood price/distance features
  → boosts R^2 and lowers MAPE.

- **Target Log Transformation**
  Apply log1p to ClosePrice → reduces skew and stabilizes training.

- **Outlier Filtering**
  Use price bounds or percentile/IQR rules
  → remove extremes and improve model stability.

Future Steps

Data Enhancements	Modeling Enhancements	Validation Improvements
• Expand dataset (Sep–Dec 2025) for seasonality • Add macroeconomic signals (rates, CPI, unemployment) • Enrich location intelligence (schools, commute distance)	• Experiment with ensemble stacking (LightGBM + XGBoost + LSTM). • Apply deeper explainability methods (SHAP, PDPs). • Deploy the model as an interactive Streamlit/dashboard app.	• Implement time-based cross-validation for realistic forecasting. • Generate regional error heatmaps. • Identify geographic overpricing/underpricing clusters.

What We've Learned

- Recognized that data cleaning drives model quality.
- Observed that feature engineering often outperformed algorithm changes.
- Improved model accuracy using neighborhood + ratio features.
- Strengthened teamwork through structured collaboration.

Thank you!