Zillow's Home Value Prediction

Kaggle Competition

Team Regression Aggression

Yu-Han Chen, Mike Ghoul, Julia Goldstein, Andre Toujas

Outline

1. **Machine Learning Preparation**

    Introduction & Objectives

    Workflow Chart

    Exploratory Data Analysis

    Imputing Missing Values

    Feature Selection & Engineering

2. **Model Iteration**

    Initial Model Results

    Model-Based Feature Selection

    Additional Model Results

3. **Summary**

    Insights & Future Work

Machine Learning Preparation

- ✓ Introduction & Objectives
- ✓ Workflow Chart
- Exploratory Data Analysis
- Imputing Missing Values
- Feature Selection & Engineering

Introduction & Objectives

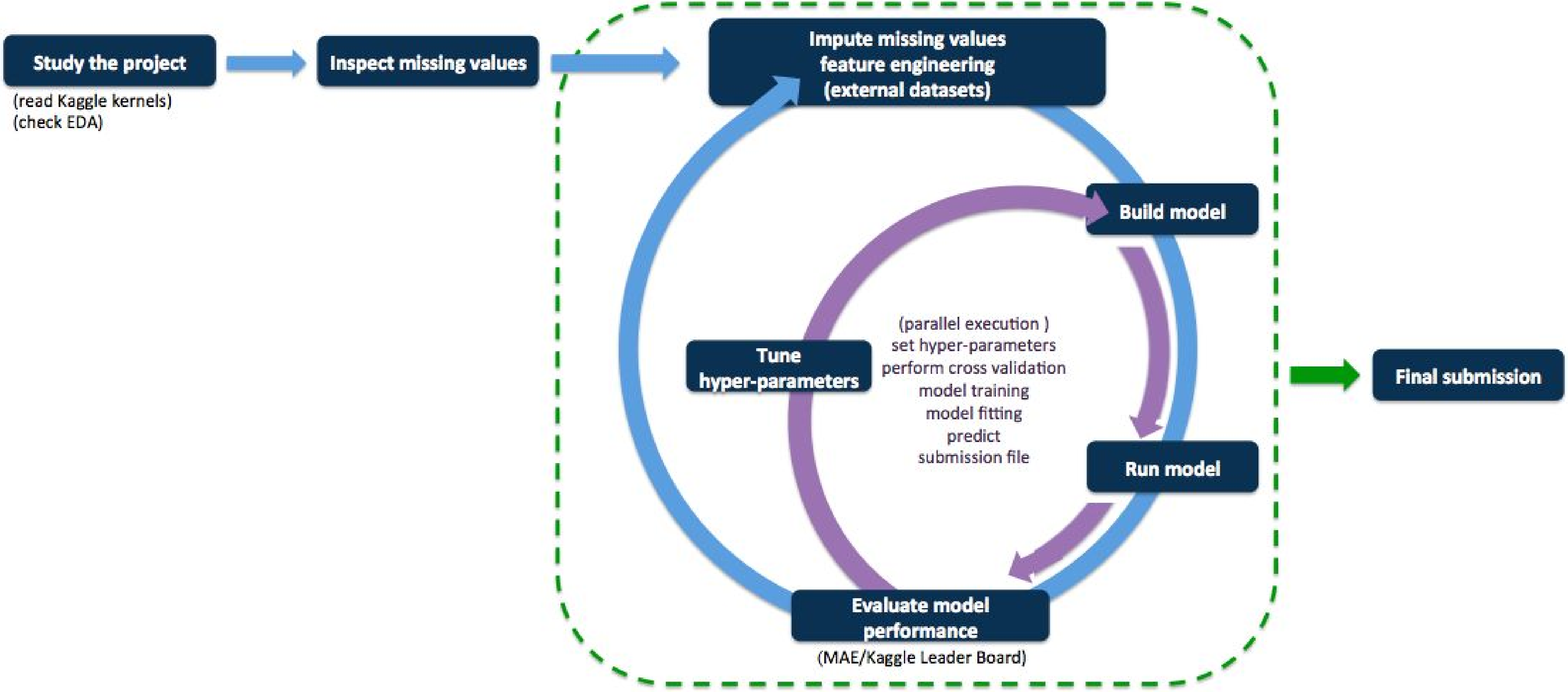
Competition Objective:

Utilize machine learning techniques to accurately predict logerror from Zestimate vs. actual sales price.

Project Objectives:

- ✓ Understand the machine learning process flow.
- ✓ Practice using and tuning different machine learning algorithms.
- ✓ Try a variety of imputation and feature engineering techniques to improve models.
- ✓ Learn how to work within a team on a data science project .

Workflow Chart



Study the project

(read Kaggle kernels)
(check EDA)

Inspect missing values

Impute missing values
feature engineering
(external datasets)

Build model

Run model

Evaluate model performance

(MAE/Kaggle Leader Board)

Tune hyper-parameters

(parallel execution)
set hyper-parameters
perform cross validation
model training
model fitting
predict
submission file

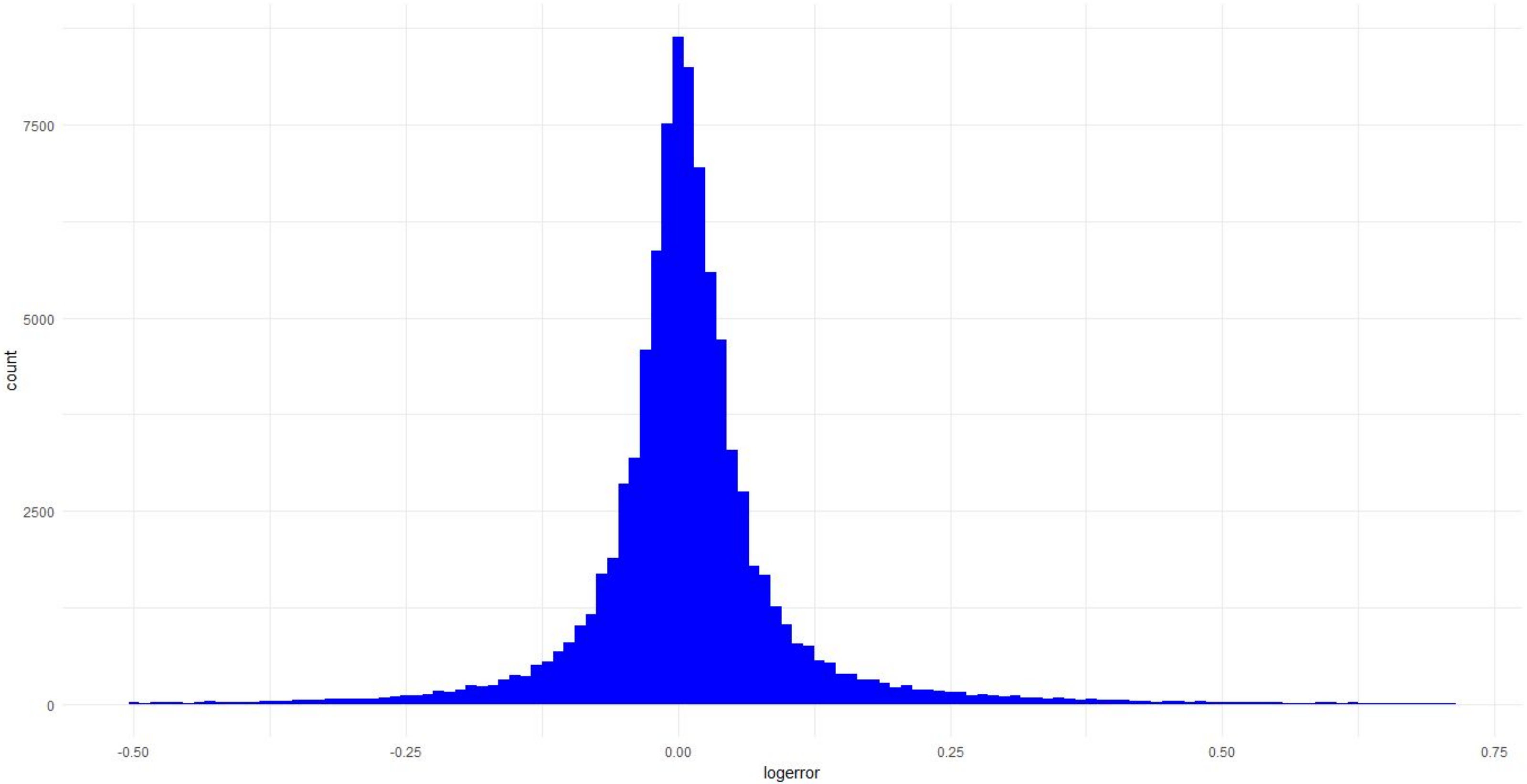
Final submission

Machine Learning Preparation

- Introduction & Objectives
- Workflow Chart
- ✔ Exploratory Data Analysis
- Imputing Missing Values
- Feature Selection & Engineering

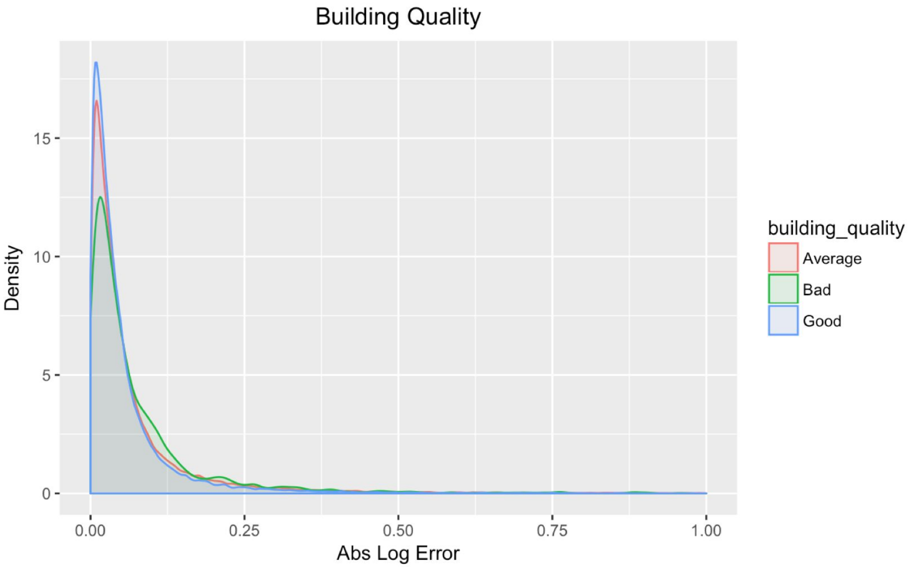
Exploratory Data Analysis

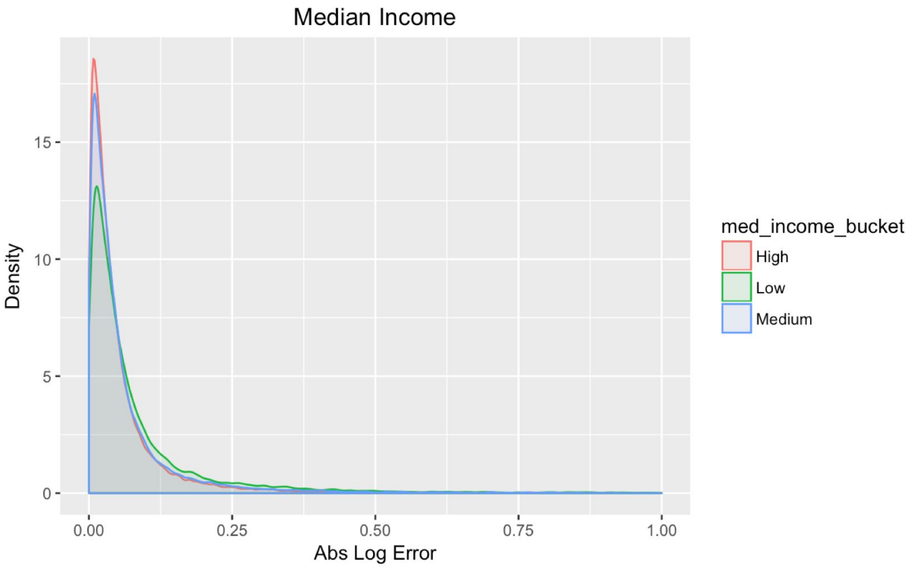
Distribution of Log Error



Exploratory Data Analysis

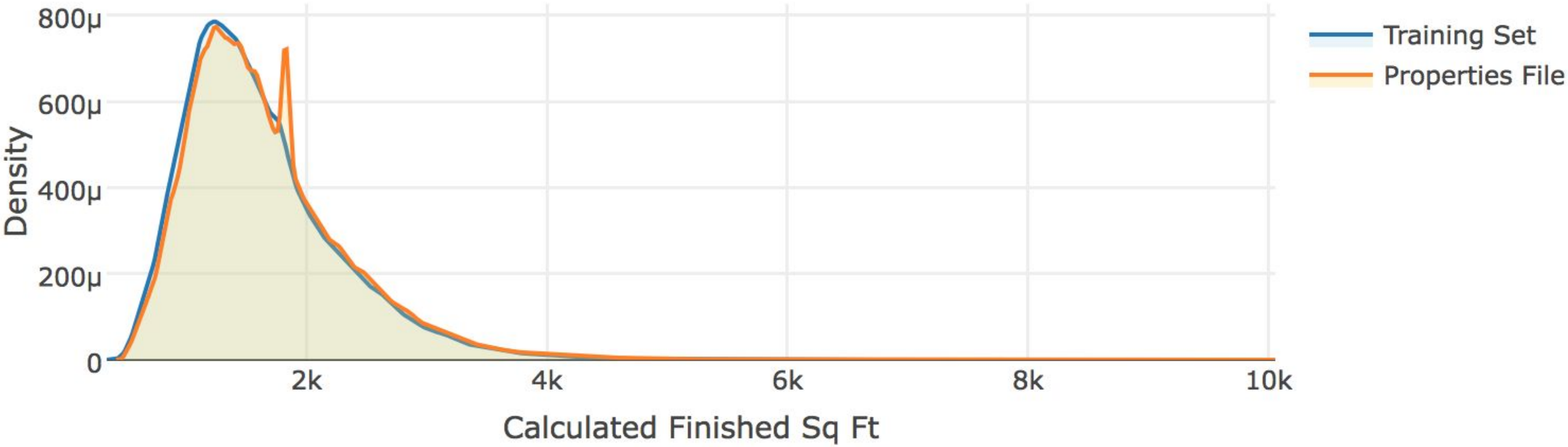
Comparison of Absolute Log Error Distribution



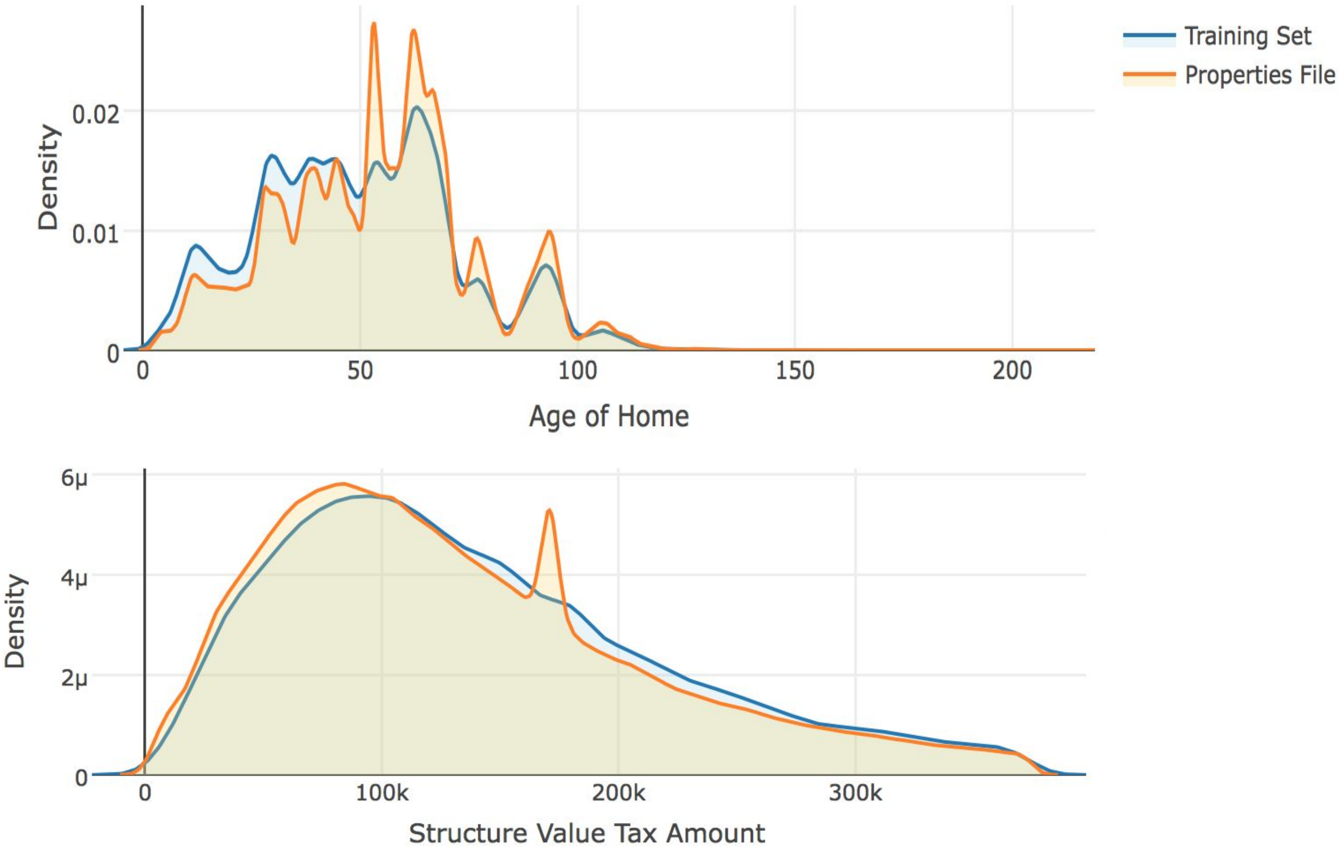


Exploratory Data Analysis

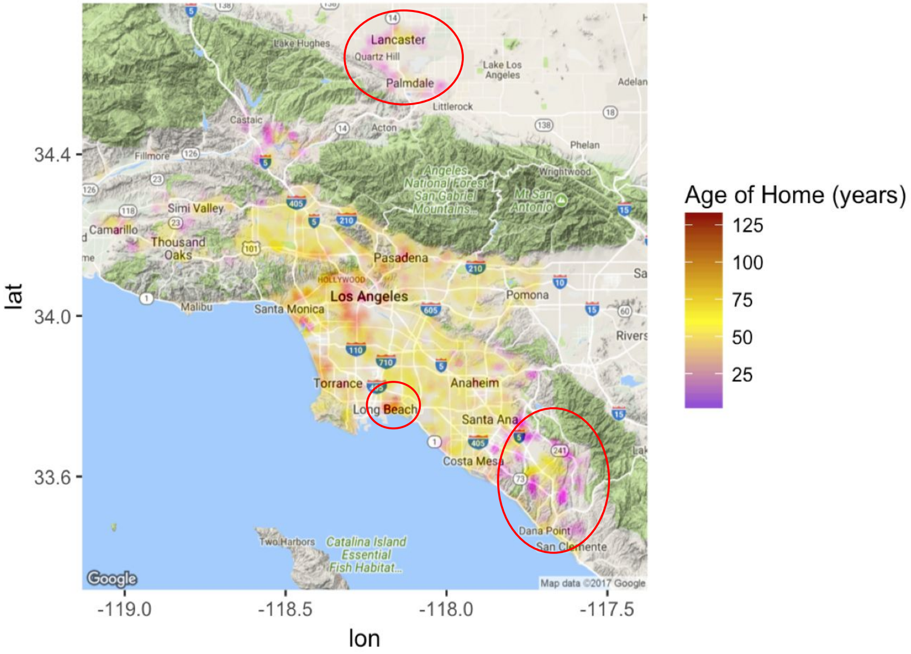
Comparison of Distribution Within Training & Properties Data

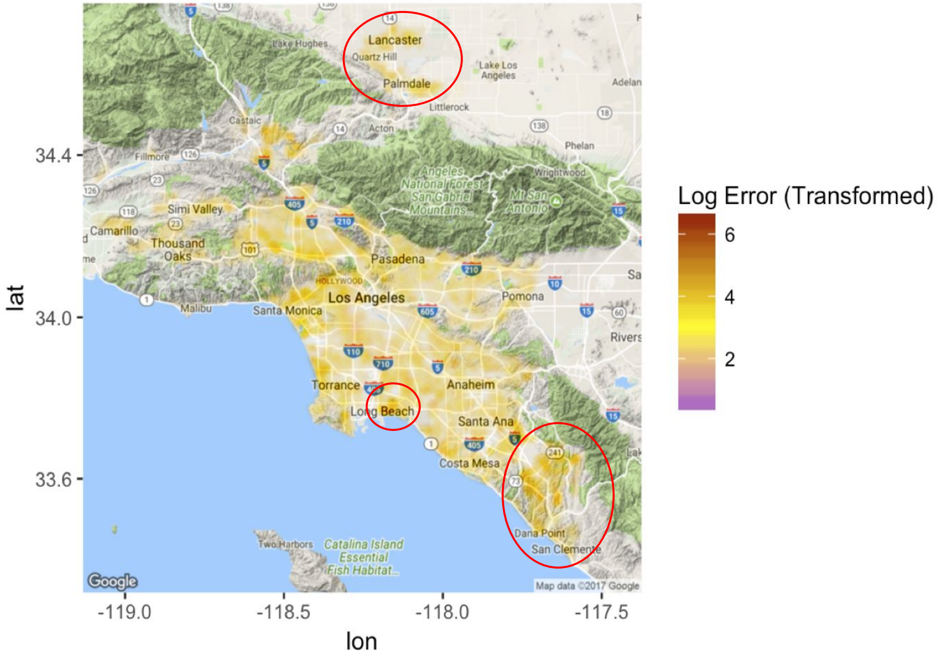


Exploratory Data Analysis

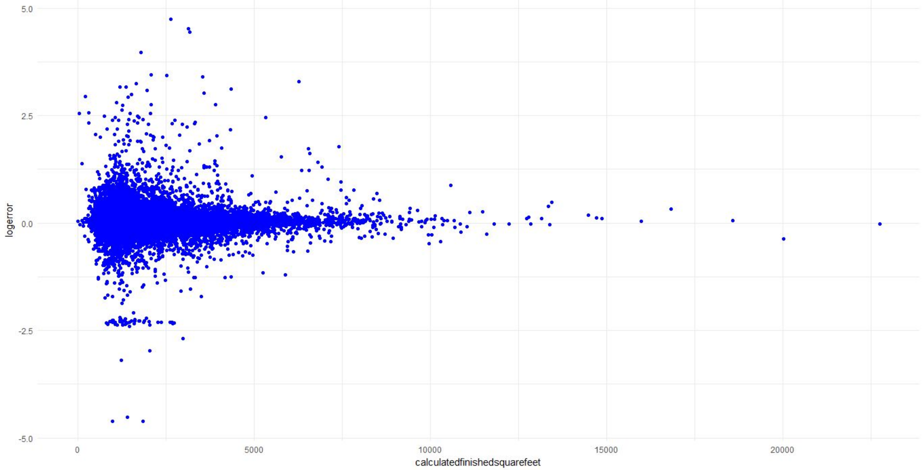


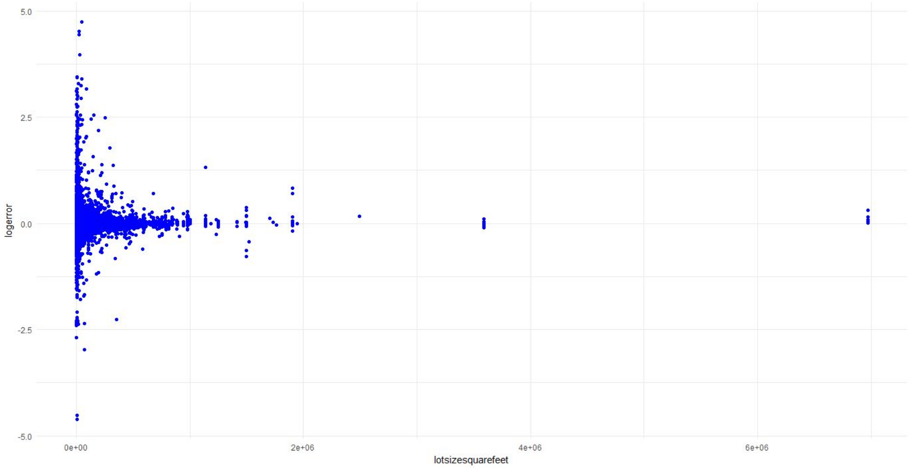
Exploratory Data Analysis

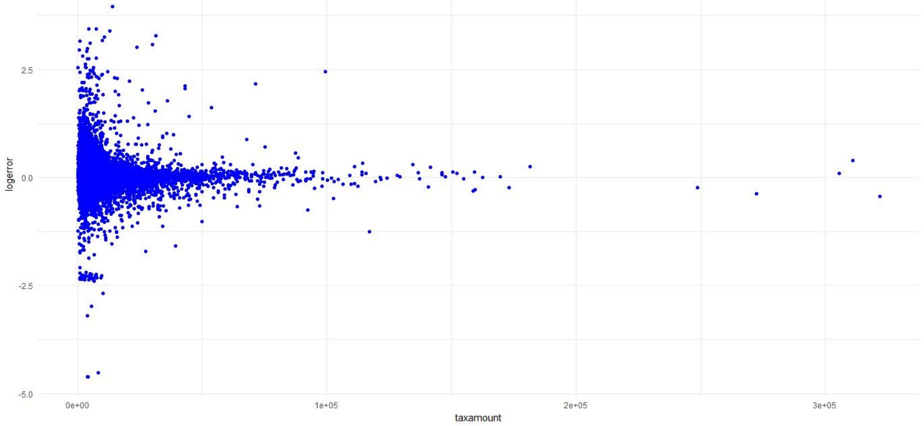


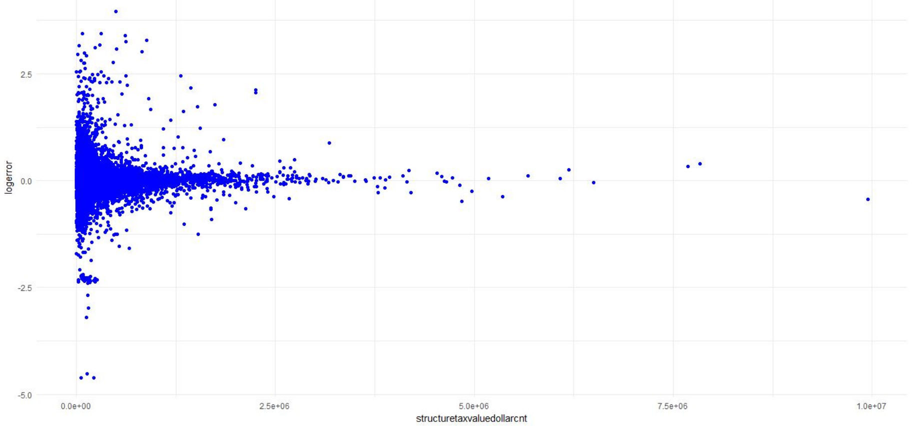


Exploratory Data Analysis

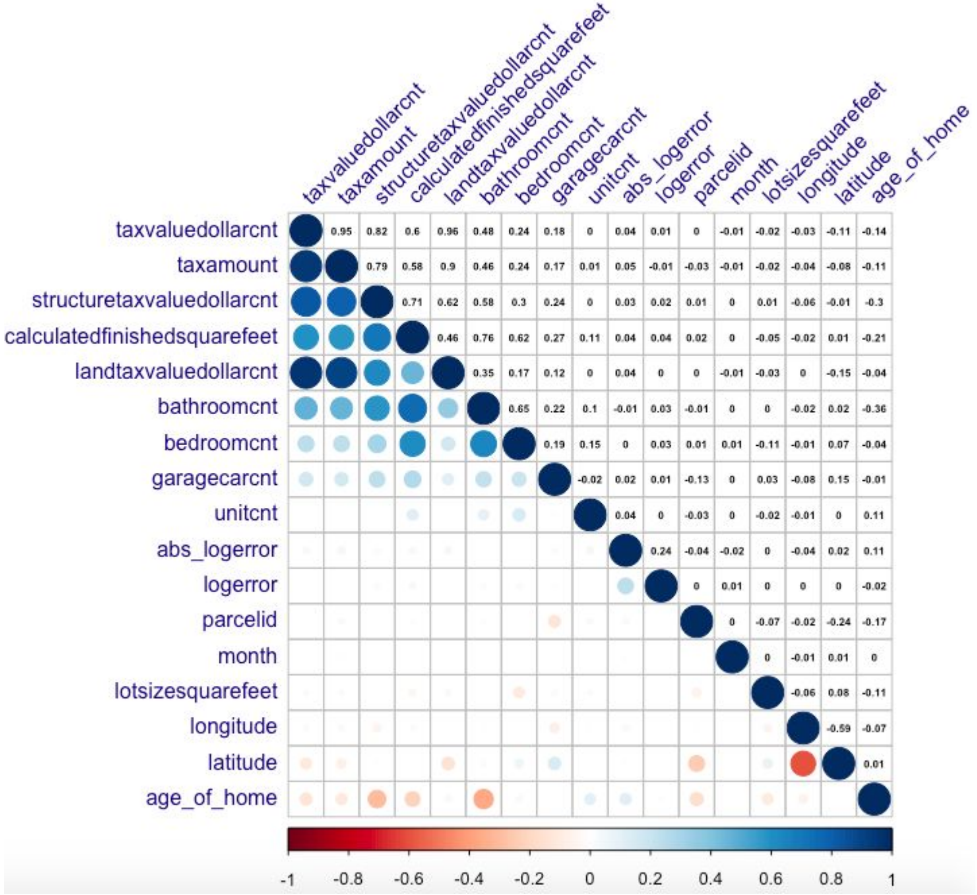


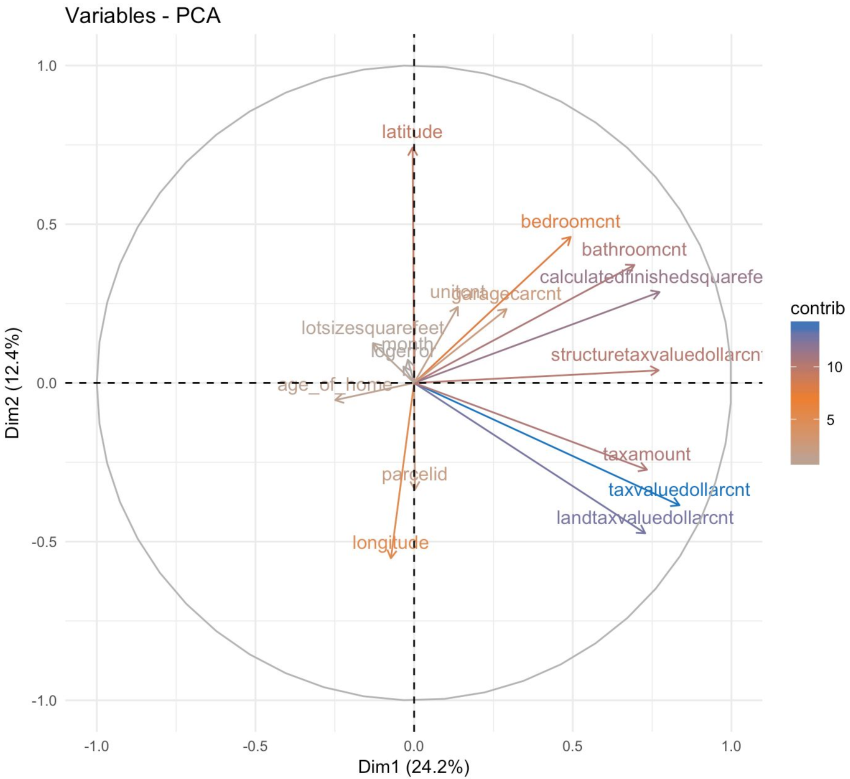






Exploratory Data Analysis

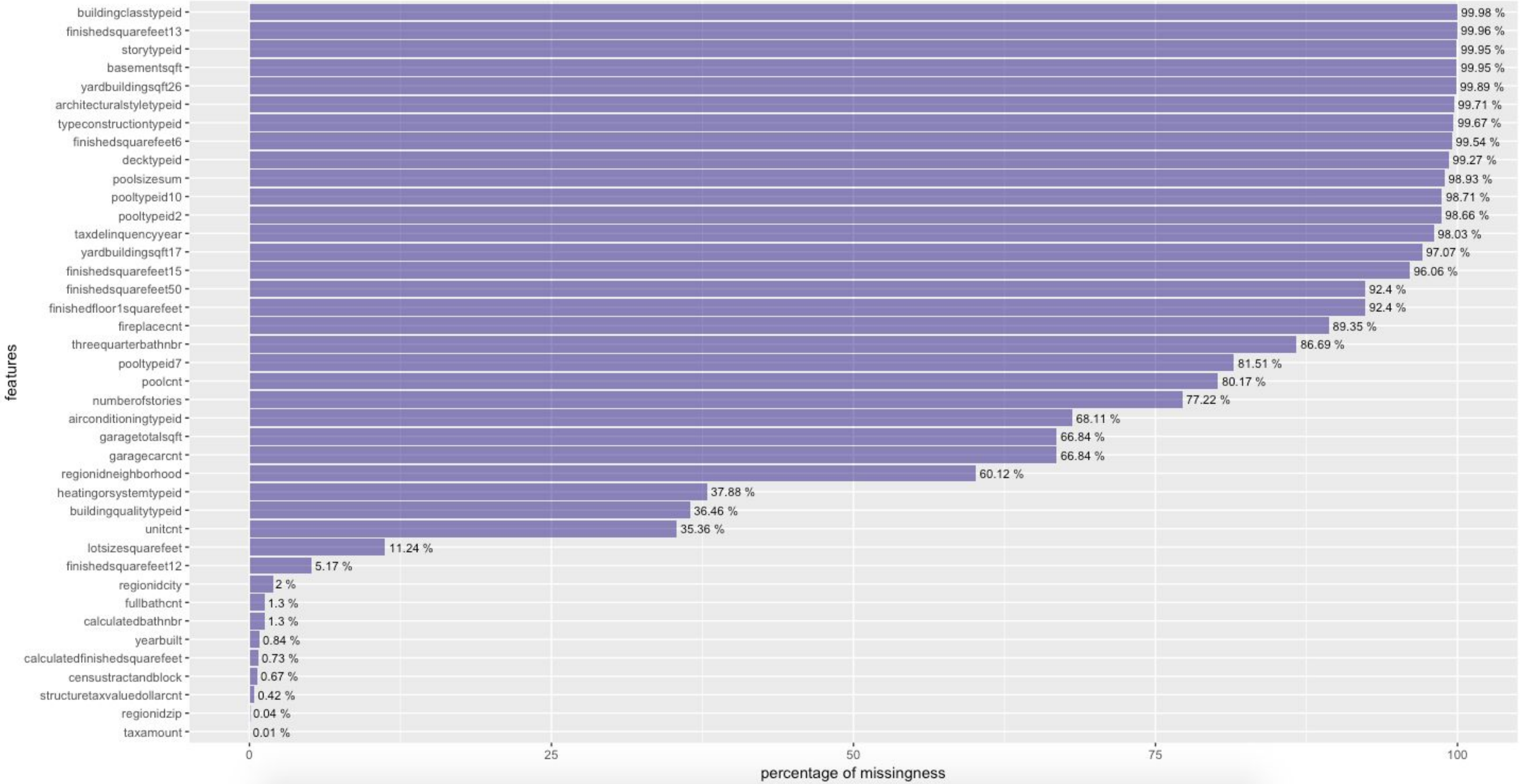




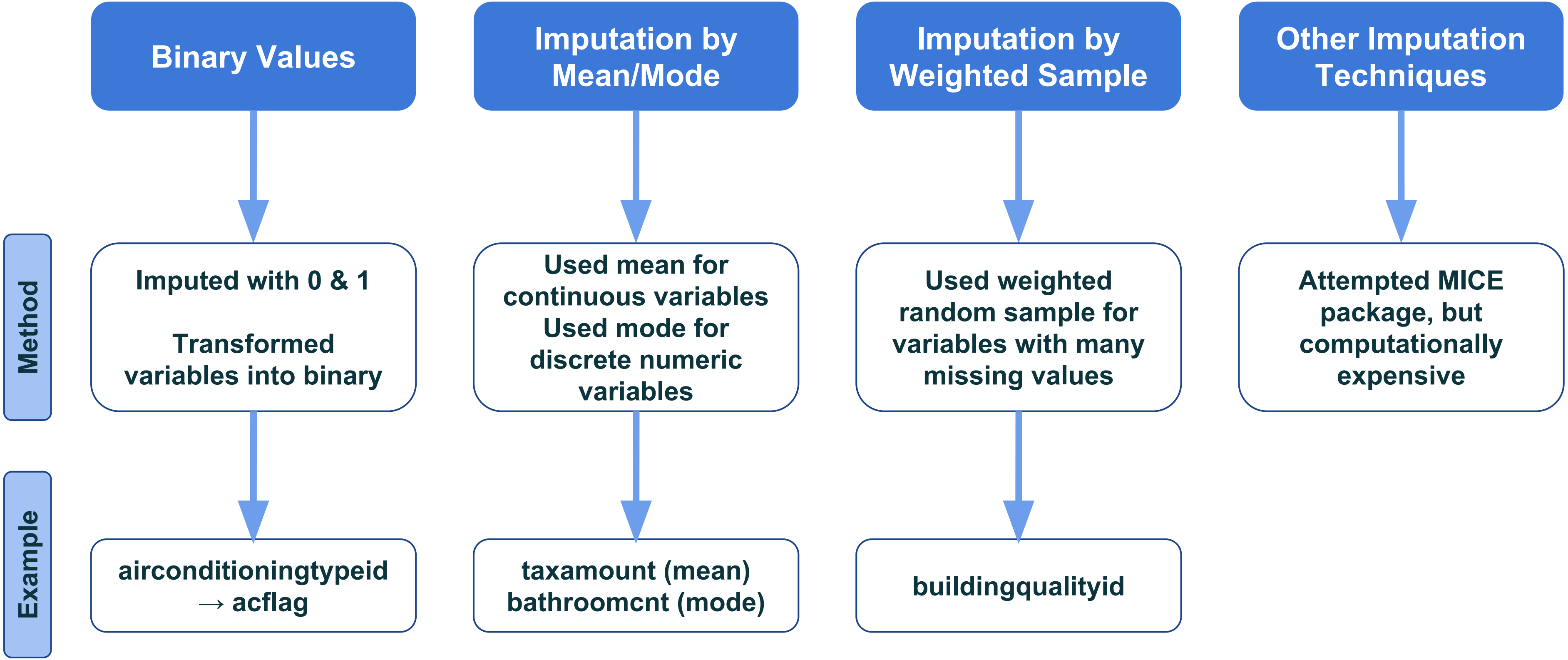
Machine Learning Preparation

- Introduction & Objectives
- Workflow Chart
- Exploratory Data Analysis
- ✓ Imputing Missing Values
- ✓ Feature Selection & Engineering

Missingness



Imputation: Base Case



	Binary Values	Imputation by Mean/Mode	Imputation by Weighted Sample	Other Imputation Techniques
Method	Imputed with 0 & 1 Transformed variables into binary	Used mean for continuous variables Used mode for discrete numeric variables	Used weighted random sample for variables with many missing values	Attempted MICE package, but computationally expensive
Example	airconditioningtypeid → acflag	taxamount (mean) bathroomcnt (mode)	buildingqualityid	

Feature Selection

We dropped several variables based on:

1

Extreme Missingness

Variables with over 95% missingness and no feasible way to determine the correct value (*e.g., architecturalstyletype*)

2

Duplication

Variables with same/similar information captured by other variables (*e.g., fips and zipcode*)

3

Zero Variance

Variable with the same value across all observations (*e.g., assessmentyear*)

Feature Engineering

We created several new features, including:

1. Total Room Count
2. Transaction Month
3. Age of Home
4. Value Ratio
5. Building Quality
6. Property Type

After imputing, dropping variables, and adding new features, we had **18** variables left to predict log error prior to our first run.

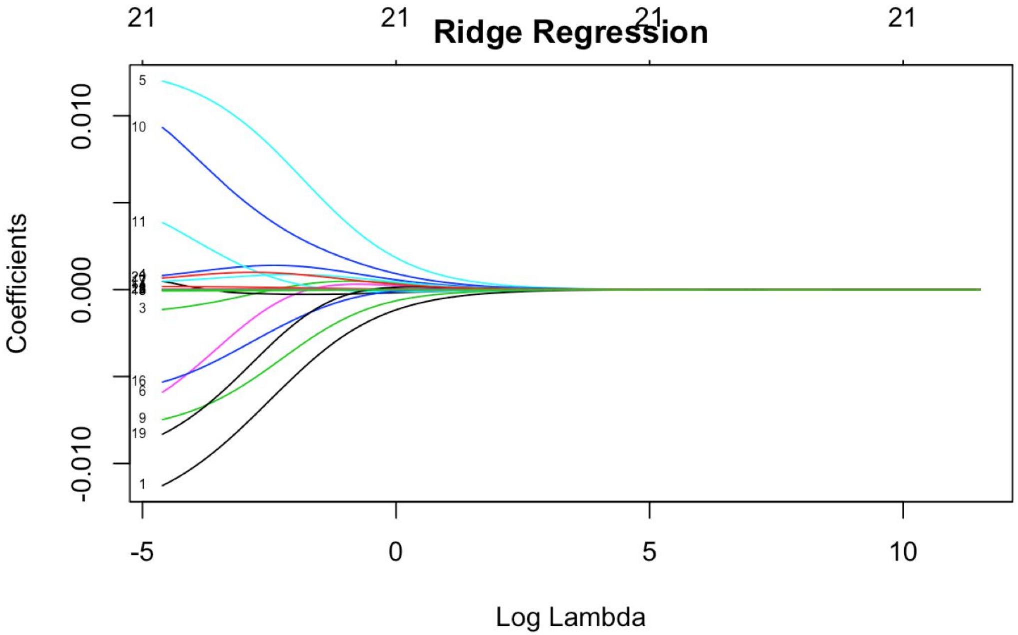
Model Iteration

✔ Initial Model Results

Model-Based Feature Selection

Additional Model Results

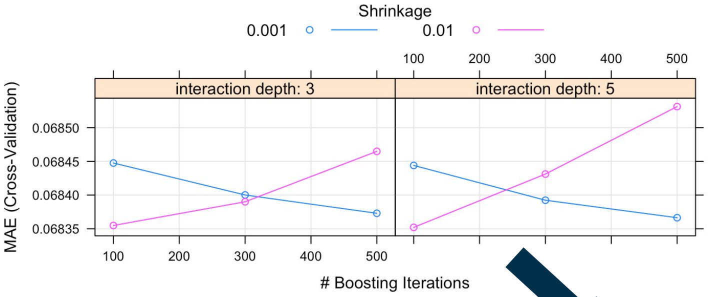
Model Results: Linear Regression

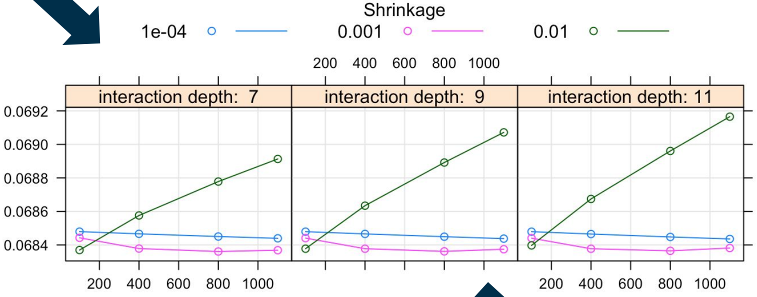


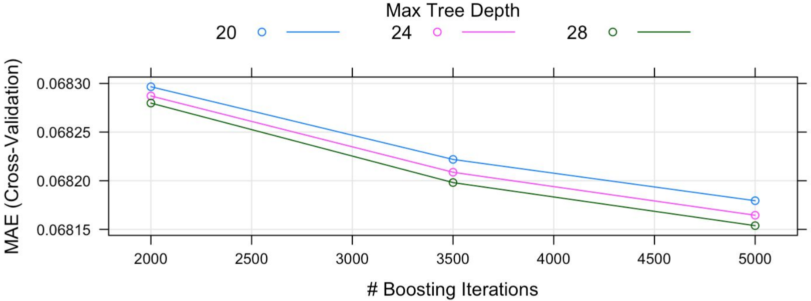
AIC
Calculated Finished Sq Ft
Land Tax Value
Has a Pool
Average Tax Metric
Latitude
County
Structure Tax
Age of Home
Living Area Metric

BIC
Calculated Finished Sq Ft
Land Tax Value
Has a Pool
Average Tax Metric

Model Hyperparameter Tuning







Tuning Insights

➔ Smaller shrinkage + higher tree count improved score

➔ Higher shrinkage + higher tree count hurt score

➔ Smallest shrinkage did not produce optimal results

Model Results

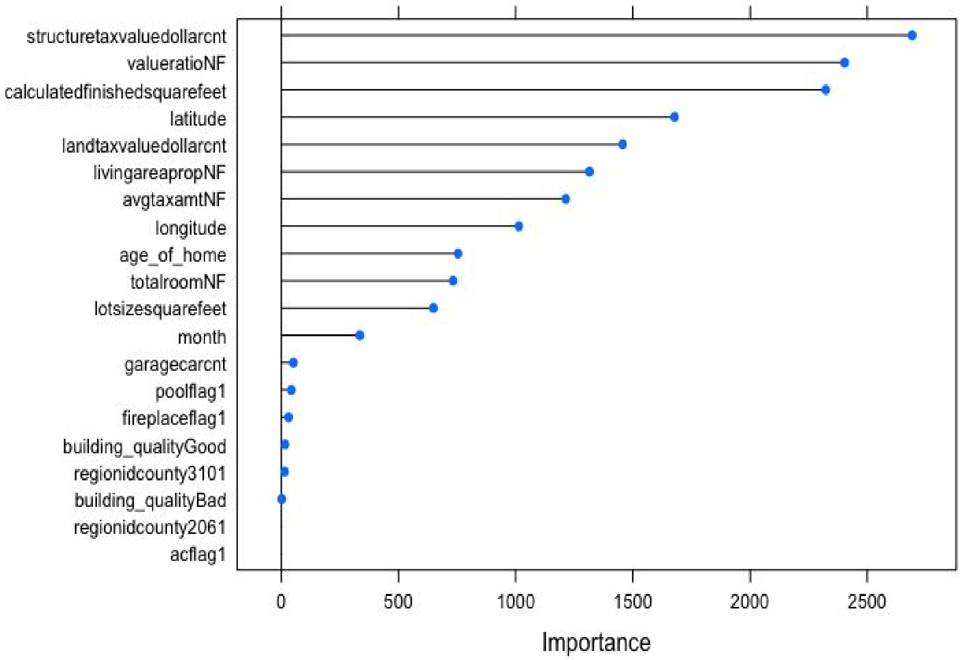
Model Type	Hyperparameters	Test Error (MAE)	Kaggle Error (MAE)

Gradient Boosting	Depth = 28
	N. Trees = 5000
	Shrinkage = .0001
		0.0679711	0.0649745

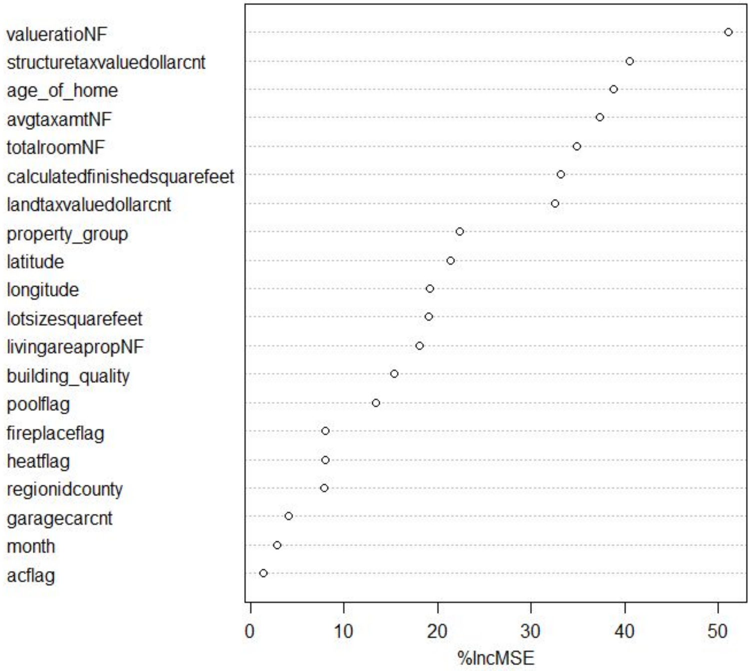
Multiple Linear Regression	Performed Ridge and Lasso	0.06747453	0.0650787
Random Forest	Mtry = 6 N. Trees = 500	0.07052706	0.0689792

Model Results: Random Forest & Boosting

Random Forest



Gradient Boosting



Model Iteration

- Initial Model Results
- ✓ Model–Based Feature Selection
- ✓ Additional Model Results

Additional Strategies

For our second attempt, we wanted to see if any of the following changes would improve our models:

- **Trimming Features by Importance:** Selected 11 variables that were important across model types.
- **Additional Feature Engineering:** Created more features based on most important variables.
- **Adding External Features:** Added Census Bureau data based on census tract of each observation.
- **Imputing Differently:** Removed the weighted random sampling method and used a simpler strategy.
- **Handling Outliers:** Replaced outliers with the median observation or removed them completely.
- **Ensembling:** Investigated whether aggregating results from several models improved score.

Model Results: Additional Strategies

Improved Model

✔ Adding 3 features based on existing data

✔ Using a simpler imputation strategy

Did Not Improve Model

✘ Using a reduced set of 11 “most important” variables

✘ Adding features from external Census Bureau data

✘ Imputing or removed outliers

✘ Ensembling models

Model Results: Final Submissions

For our final model runs, we combined methods that improved our score on their own (imputing differently and adding more features).

Model Type	Strategies Used	Hyperparameters	Test Error (MAE)	Kaggle Error (MAE)

Random Forest	New imputation
	Mtry through cross validation
	Iterative approach for N.Trees
		Mtry = 1
		N.Trees = 4000
			0.06773457	0.0648092

Boosting	New imputation
	Three new features
		Depth = 18
		N. Trees = 600
		Shrinkage = .001
			0.06855073	0.0649608

Summary

- ✔ Insights
- ✔ Future Work

Insights

Specific Project Insights:

- Imputation that seems the most “logical” does not yield the best result.
- Feature engineering proved to add value.
- Agile machine learning allowed us to run models more effectively and efficiently.
- Reducing the number of features produced worse results.
- Random Forest produced the best results over all other model types.

Insights

Overall Machine Learning Insights:

- Understand how long it will take to fit a model before running it.
- Limit the number of parameters to check during cross validation when working with large data sets.
- Set up a modeling process flow from the beginning, rather than diving in.

Future Work:

- Continue to engineer features.
- Use PCA to select variables.
- Apply different ML algorithms (e.g., SVR, Clustering).
- Try other imputation techniques.