@ NYC Data Science Academy

(23rd August 2017)

Kaggle Competition: Mission Zillow

by

Team Entropy

Janet Hu, Wei Liu, Yadi Li, Shivakumar Ranganathan, Suyash Chopra

1 Overview

2 Exploratory Data Analysis

3 Data Visualization & Analysis

4 Conclusions

5 Future Work

1 Overview

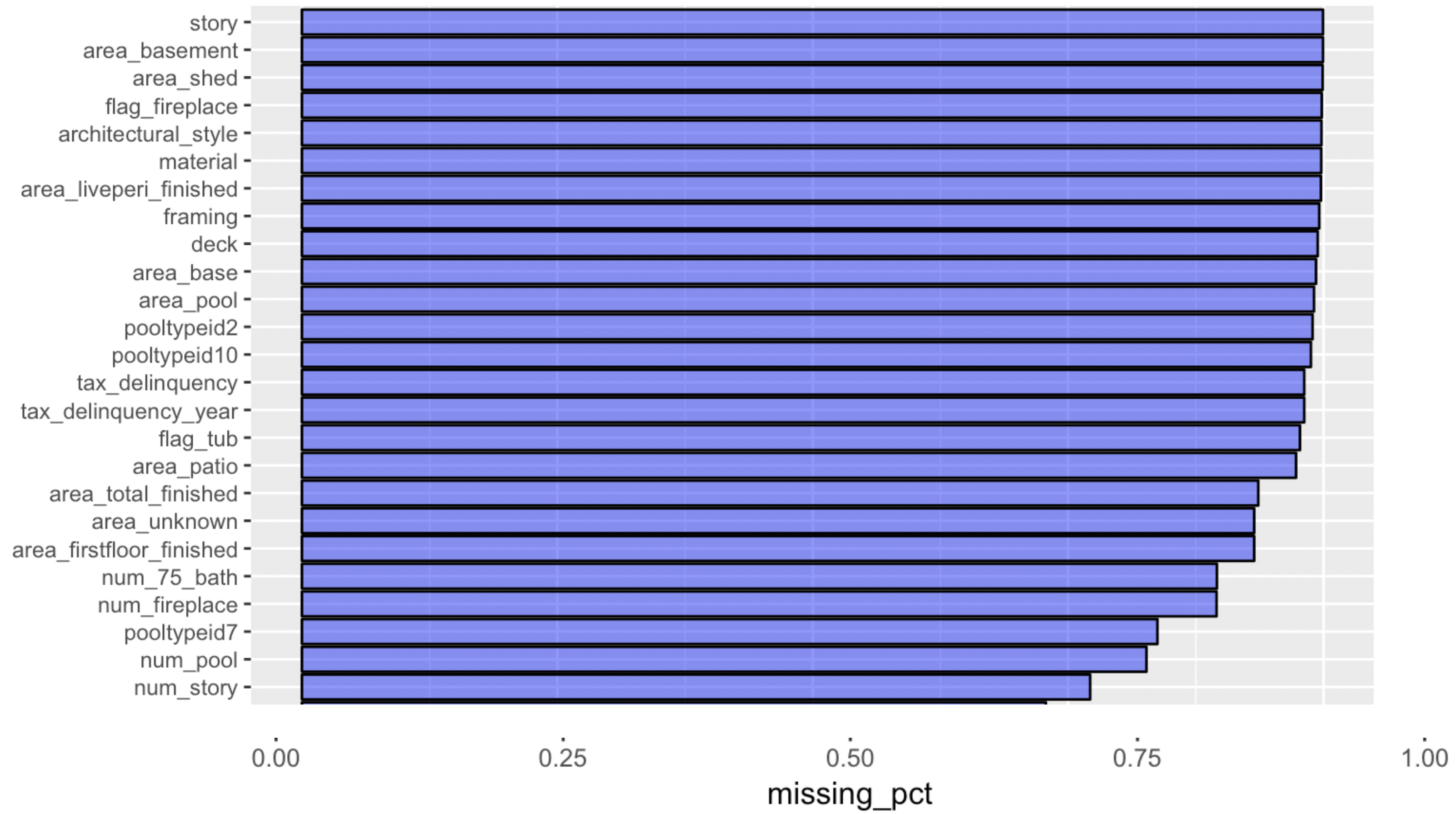
- Kaggle competition: Develop a Machine Learning algorithm that makes predictions about the future sale prices of homes (better than Zestimate?)

- Objective (Round-1): Develop a model to predict logerror

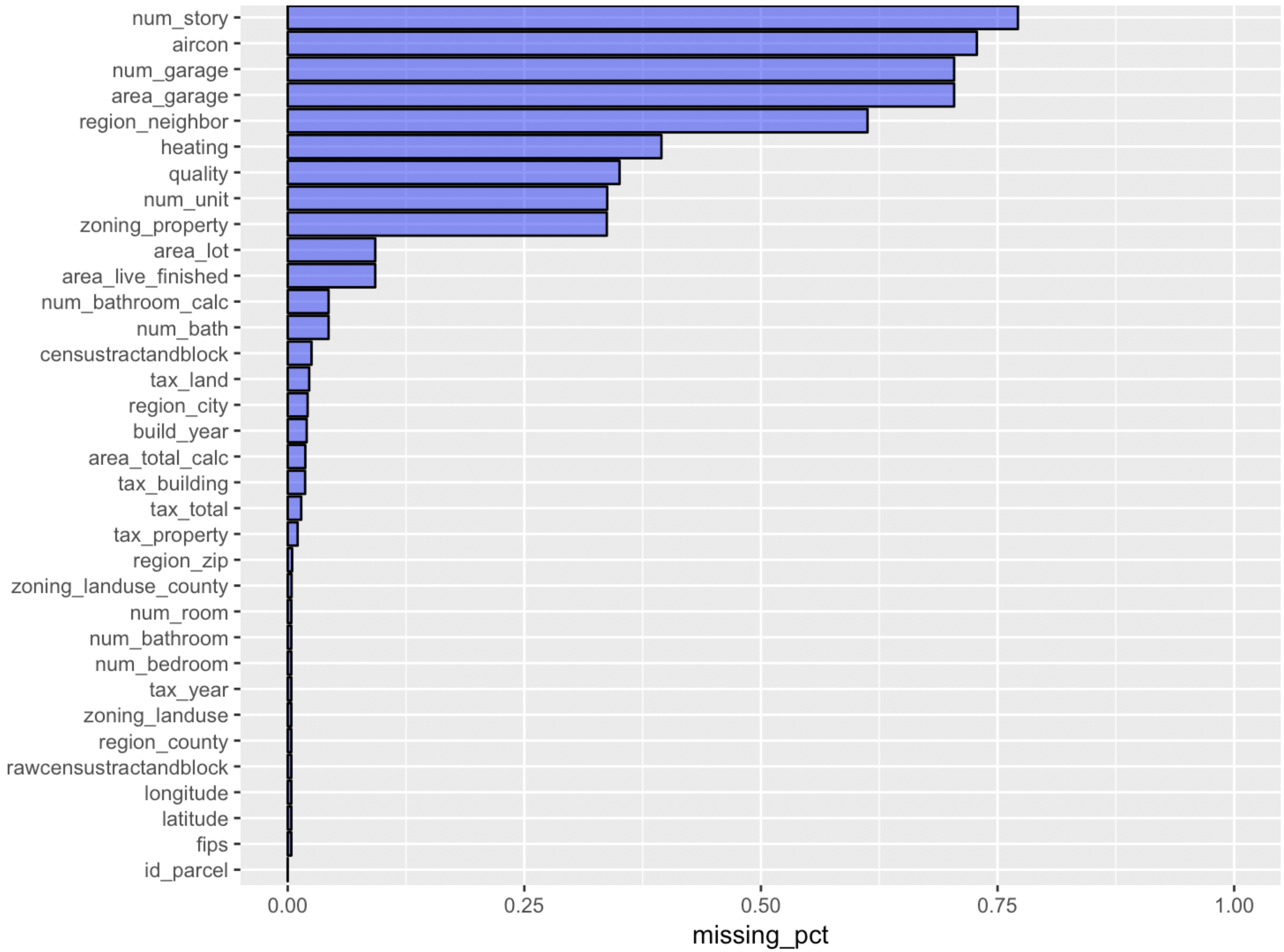
$$logerror = log(Zestimate) - log(SalePrice)$$

- Approach: Team Entropy's strategy included the following—
  - Exploratory Data Analysis (EDA)
  - Data Imputation
  - Implementing a slew of Machine Learning algorithms including:
    a) Logic based methods (by observing the given data)
    b) Elastic net regularization (Ridge and Lasso),
    c) Tree based models (Gradient Boosting Machine, Random Forest, Extreme Gradient Boosting)
    d) Automatic Machine Learning (h2o)

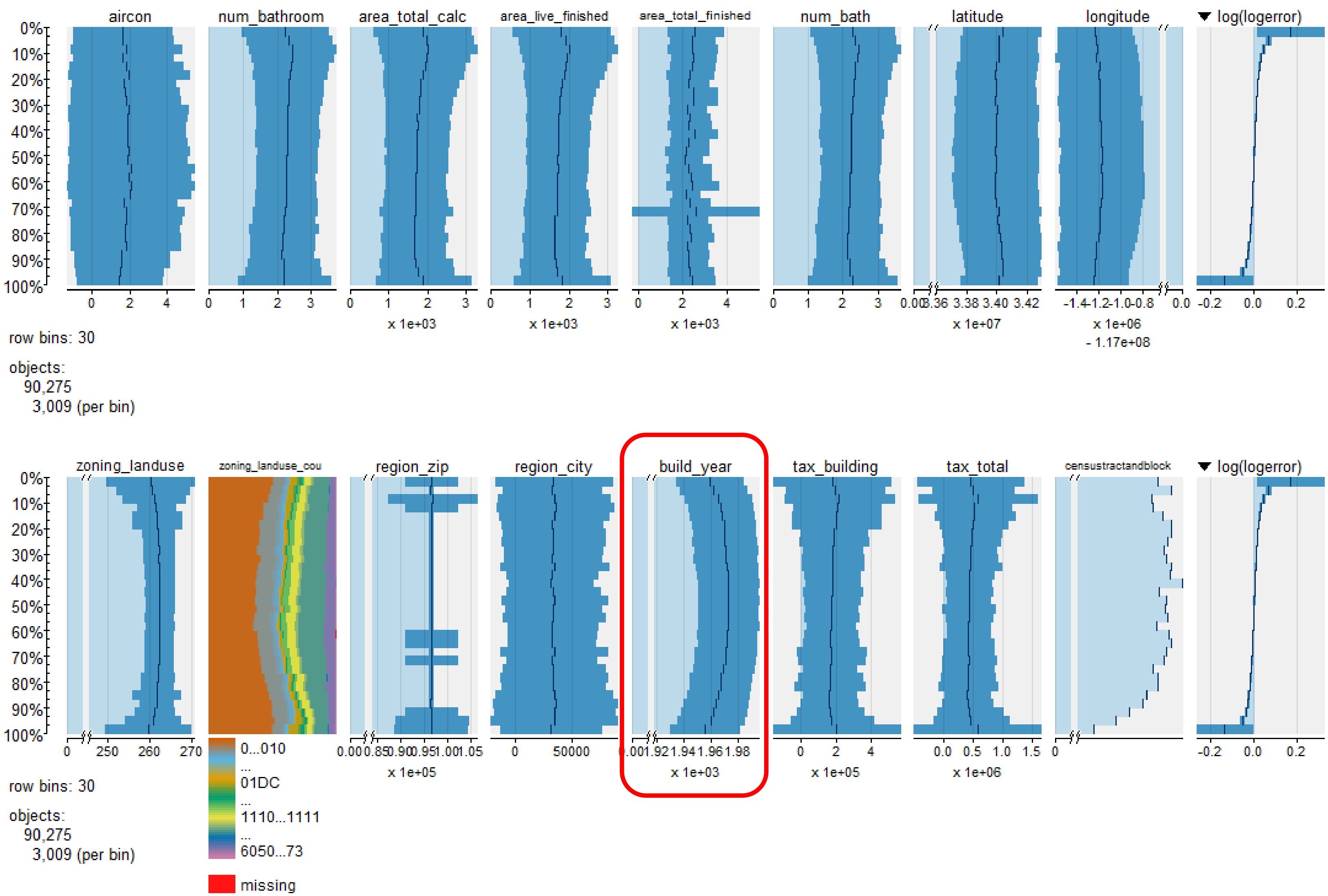
2 EDA:Analysis of Missing Data



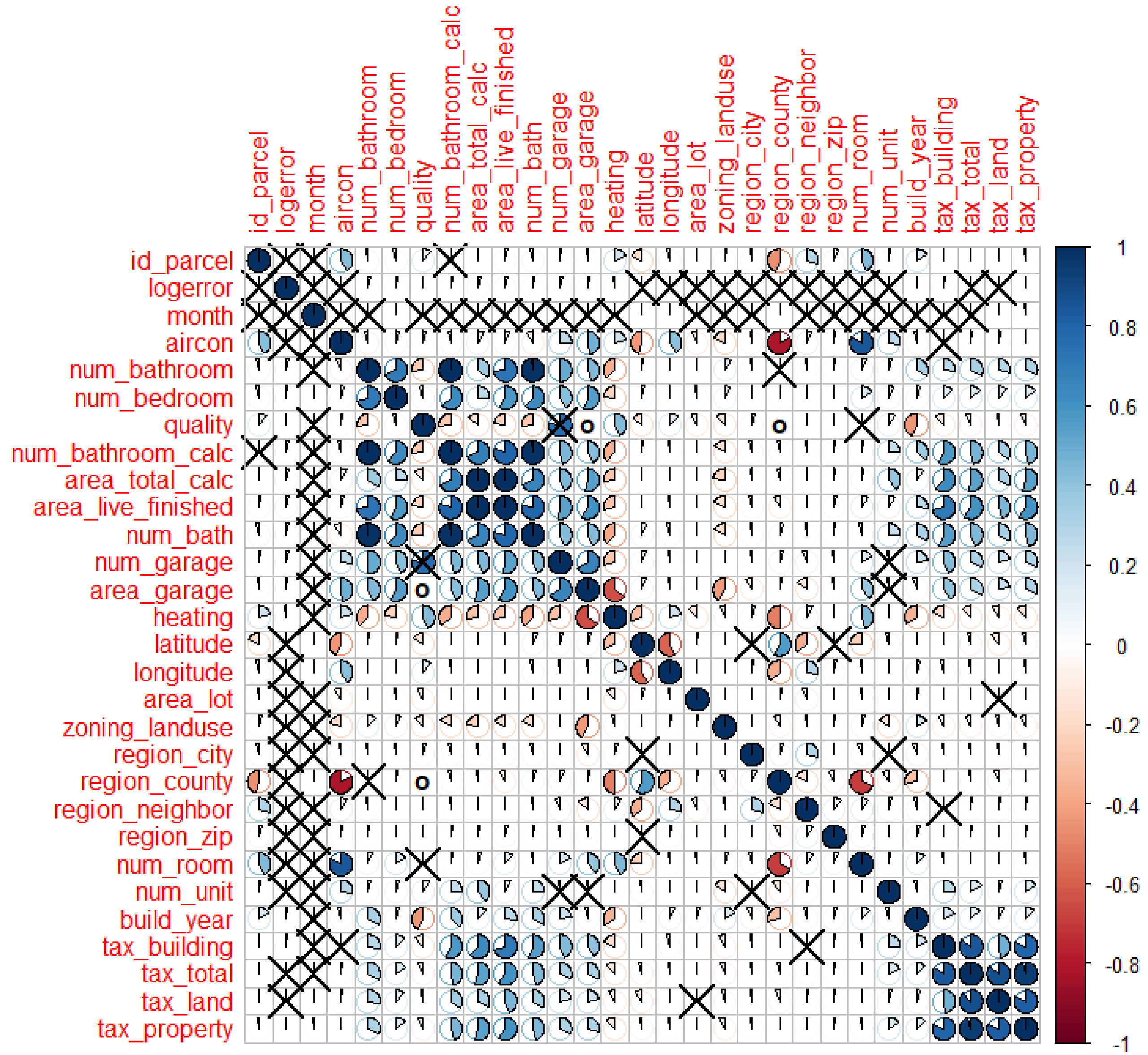
2 EDA:Analysis of Missing Data



2 EDA: tabplot



2 EDA: corrplot



3 Data Imputation:round-1

Drop NA > 95%

- Architectural_style
- Area_basement
- Framing
- Deck
- Area_liveperi_finished
- Area_total_finished
- Area_base
- Flag_tub
- Area_pool
- Pooltypeid10
- Pooltypeid2
- Story
- Material
- Tax_delinquency
- Tax_delinquency_year
- Censustractandblock

Multicollinearity

- Region_neighbor
- Rawcensustractandblock
- Zoning_property
- Fips
- area_unknown
- num_bathroom_calc
- area_firstfloor_finished
- area_live_finished

Random

- Area_garage
- Area_lot
- Build_year
- Longitude
- Latitude
- Region_county
- Region_zip

3 Data Imputation:round-1

Impute by Making Factor

- Aircon
- Heating
- Num_pool
- Pooltypeid7
- Num_75_bath
- Flag_fireplace
- Num_story

Random with Top 4 Levels

- Quality
- Num_bathroom
- Zoning_landuse
- Num_bedroom
- Num_unit

Impute by Mean

- Tax_total
- Area_total_clc

3 Data Imputation:round-2

-999

- "architectural_style","area_basement","num_bathroom" ,"num_bedroom","framing","quality" "num_bathroom_calc","deck" ,"area_firstfloor_finished" "area_total_calc","area_live_finished" "area_liveperi_finished","area_total_finished" ,"area_unknown", "area_base" , "fips","num_fireplace" ,"num_bath" ,"num_garage" , "area_garage" , "flag_tub","latitude","longitude","area_lot","area_pool" ,"pooltypeid10" ,"pooltypeid2","zoning_landuse_county","zoning_property" ,"rawcensustractandblock" ,"region_city" ,"region_neighbor" ,"num_room" , "story","material" ,"num_unit" , "area_patio" , "area_shed" ,"build_year" , "tax_building" , "tax_total" , "tax_year" , "tax_land" , "tax_property" , "tax_delinquency" ,"tax_delinquency_year","censustractandblock"

-1

- aircon, Heating, Zoning_landuse, Region_county, Region_zip, Num_75_bath, Flag_fireplace, Num_pool, Pooltypeid7, Num_story

other

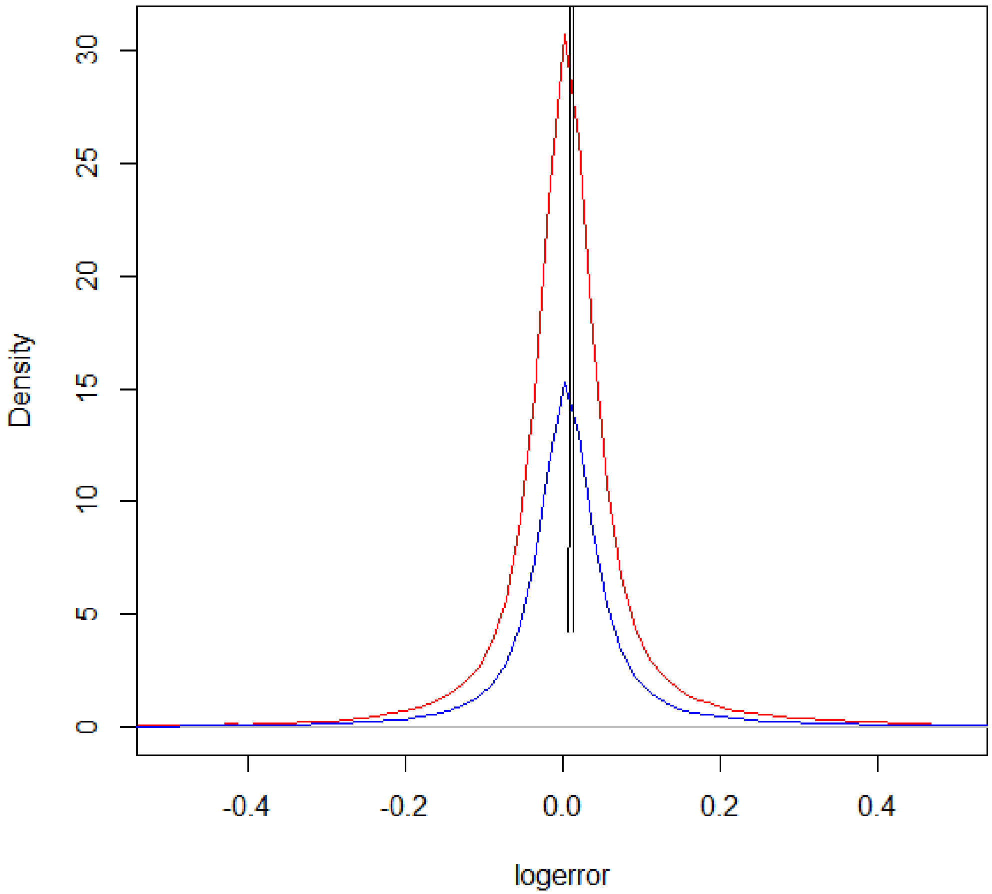
- use number to replace category content
- Change all the columns to numeric
- Scale all the dataset

4 Machine Learning Models & Results

- Logic based methods (by observing the given data)

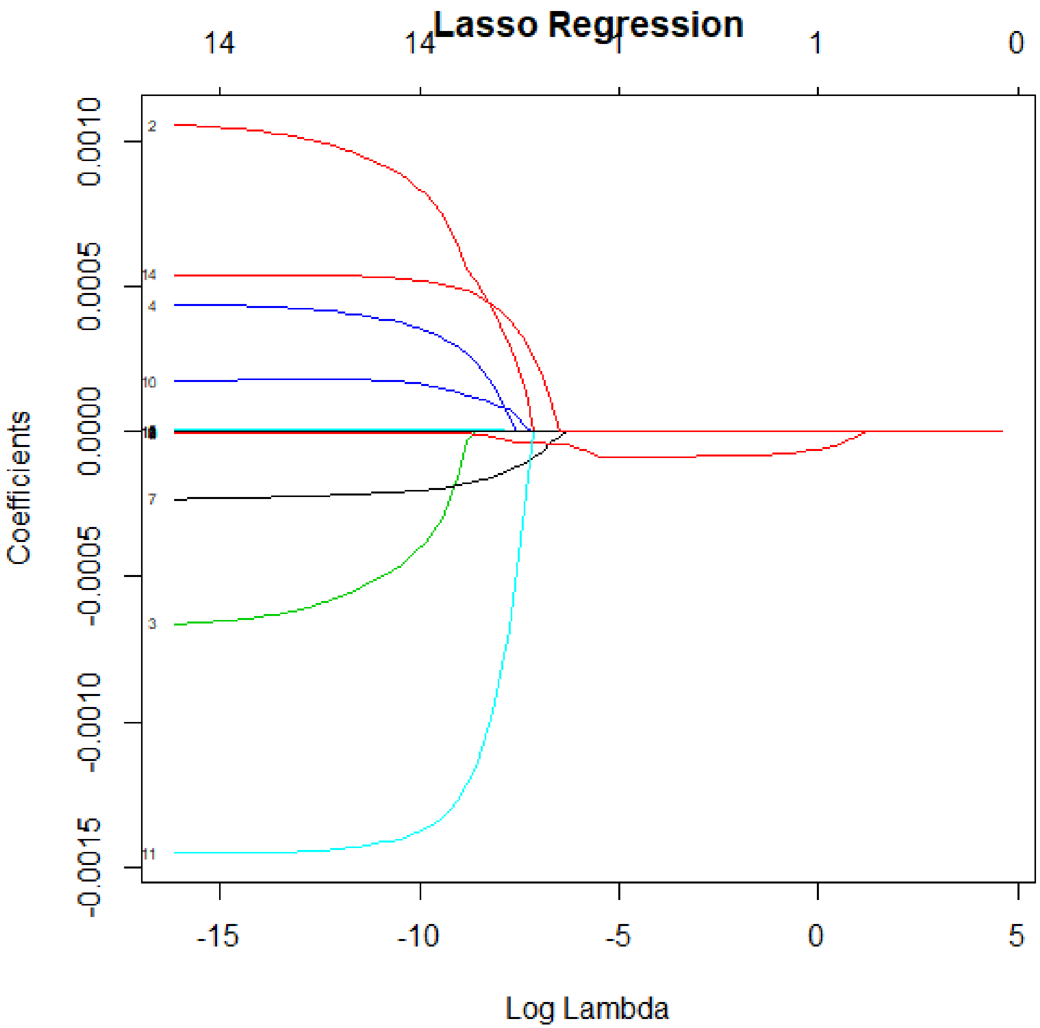
  a) Prediction using the mean value of logerror [Kaggle Score: 0.0651279]

  b) Prediction using the distribution of logerror [Kaggle Score: 0.1075059]



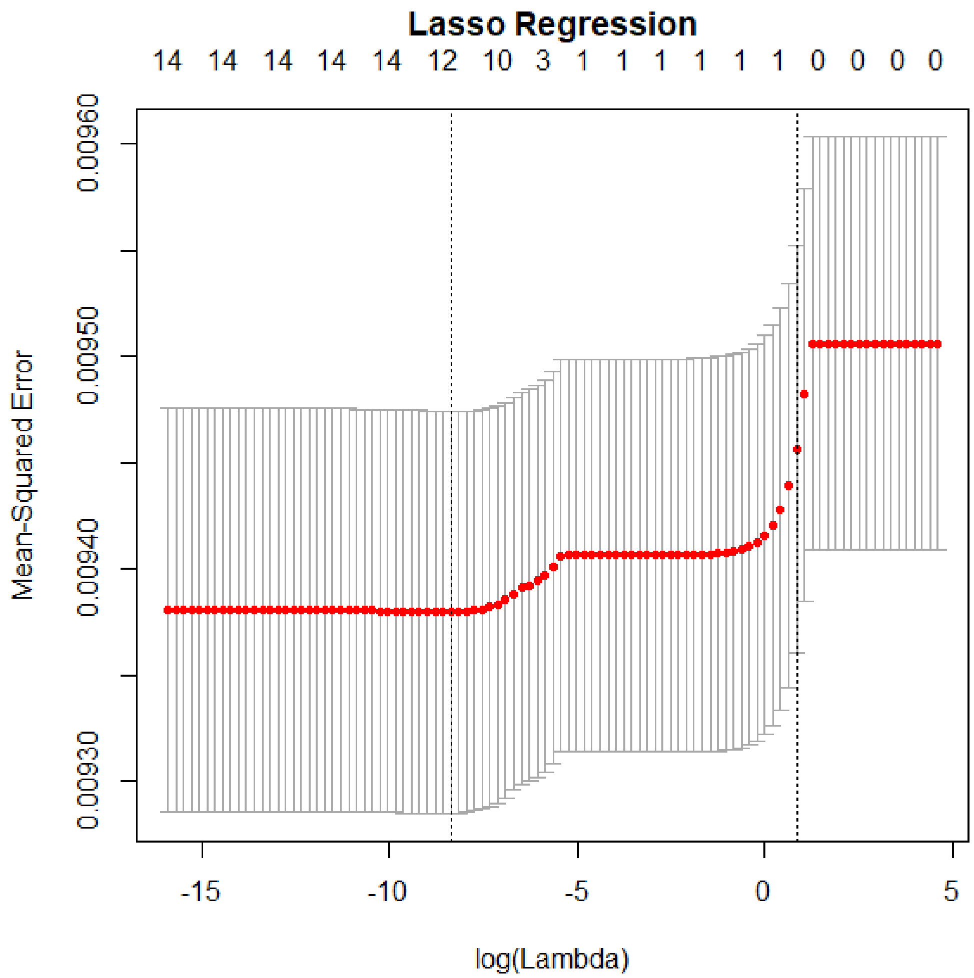
4 Machine Learning Models & Results

- Elastic net regularization (Ridge and Lasso)



4 Machine Learning Models & Results

- Elastic net regularization (Ridge and Lasso)



4 Machine Learning Models & Results

- Elastic net regularization (Ridge and Lasso) [MAE: 0.05723179, Kaggle: 0.0649128]

$$\log error = \sum_{i=1}^{12} \beta_i * x_i$$

Variable Name	Coefficient	Value
id_parcel	β_1	2.27642E-10
num_bathroom	β_2	1.70640E-03
quality	β_3	3.04056E-04
area_total_calc	β_4	5.12690E-06
latitude	β_5	-2.66799E-04
longitude	β_6	-1.93938E-05
area_lot	β_7	5.86367E-09
num_room	β_8	7.91091E-05
num_unit	β_9	-1.20248E-03
build_year	β_{10}	2.35871E-07
tax_total	β_{11}	-5.70986E-09
month	β_{12}	4.02626E-04

4 Machine Learning Models & Results

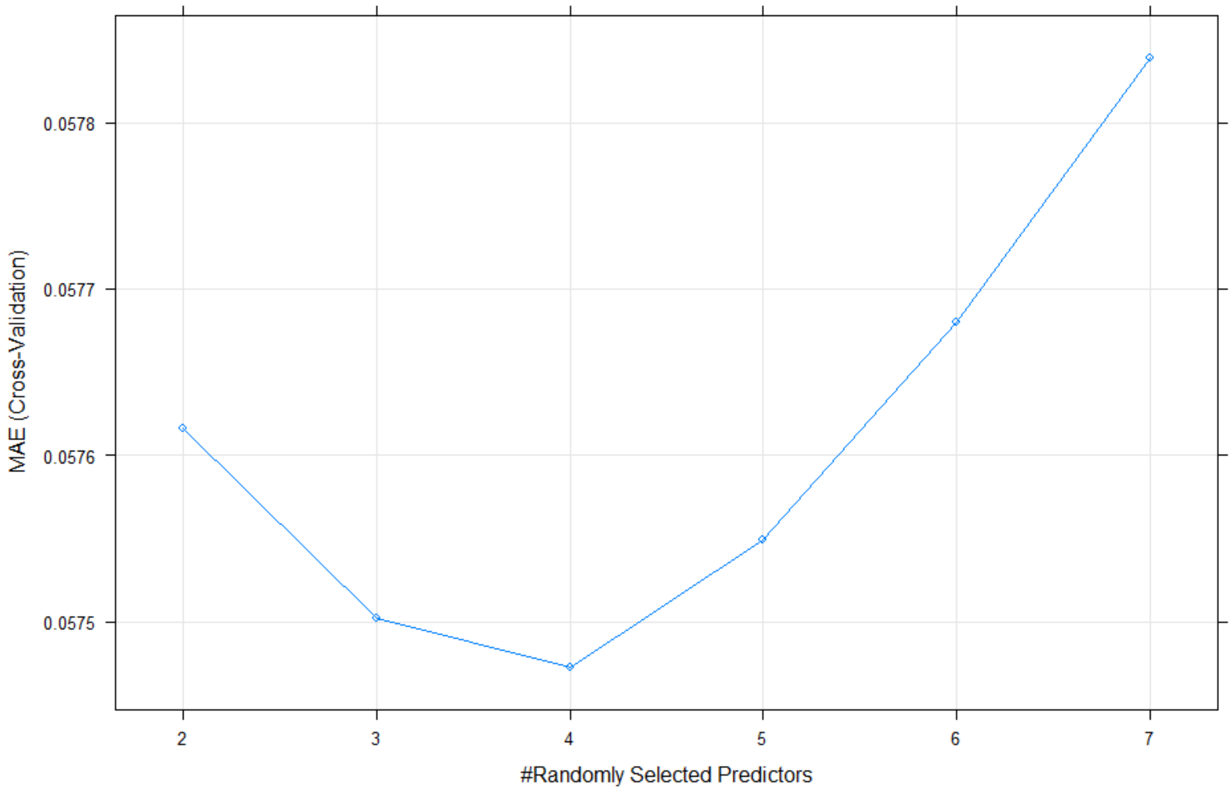
- Tree based models (Random Forest)

Hyperparameters

- ntree
- mtry
- nodesize
- maxnodes

4 Machine Learning Models & Results

- Tree based models (Random Forest)

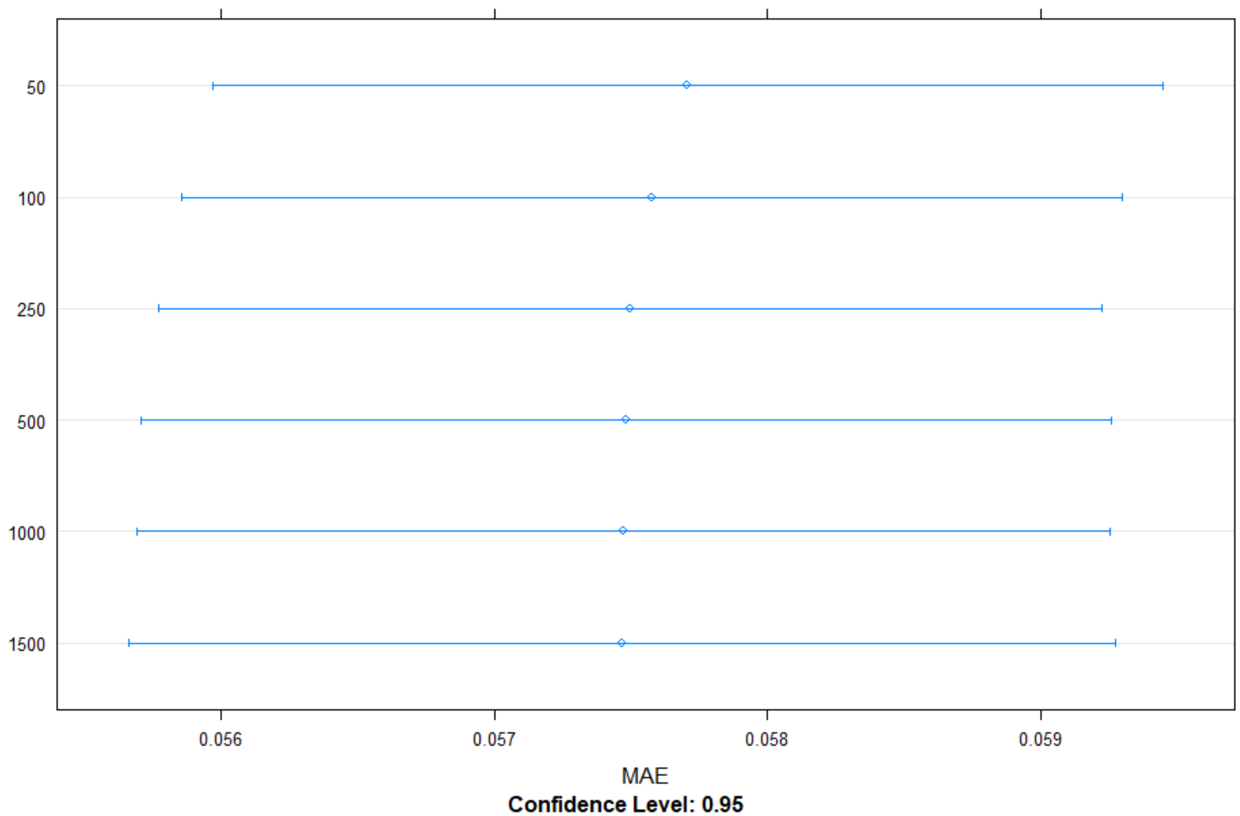


mtry	MAE
2	0.05761628
3	0.05750186
4	0.05747278
5	0.05754893
6	0.05768034
7	0.05783864

Grid search of mtry – imputed set 1

4 Machine Learning Models & Results

- Tree based models (Random Forest)



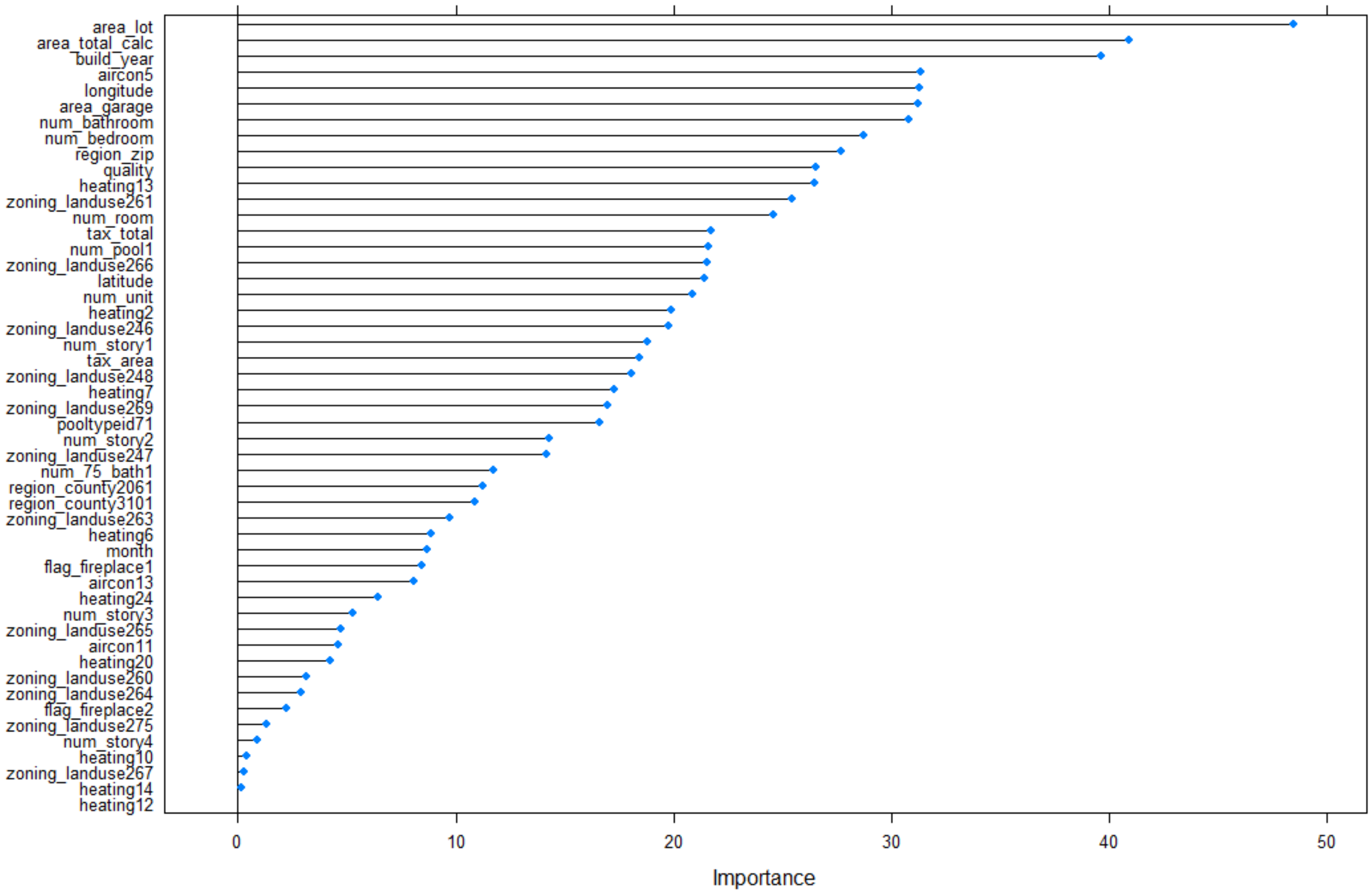
Ntree	MAE
50	0.05770776
100	0.05757854
250	0.05749845
500	0.05748335
1000	0.05747443
1500	0.05746732

ntree = 800 was used to train the final model

Manually tuning number of trees – imputed set 1

4 Machine Learning Models & Results

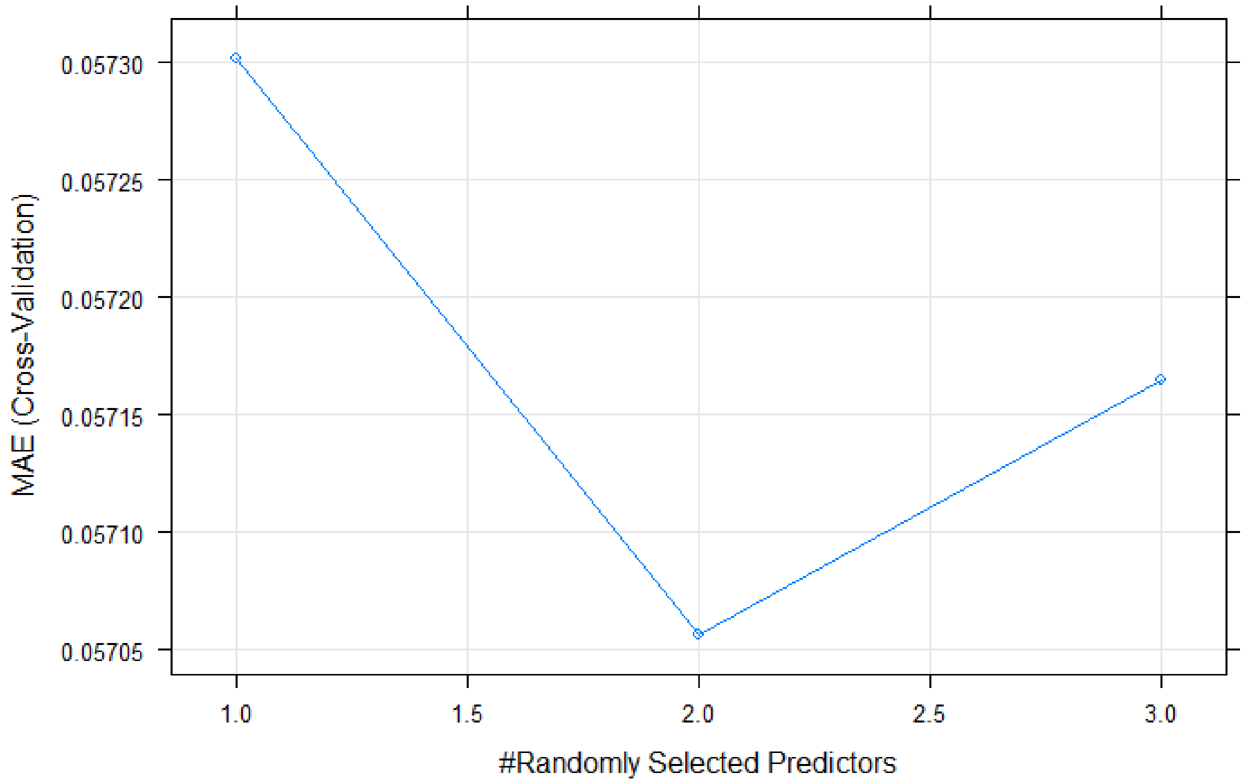
- Tree based models (Random Forest)



Variable importance plot – imputed set 1

4 Machine Learning Models & Results

- Tree based models (Random Forest)

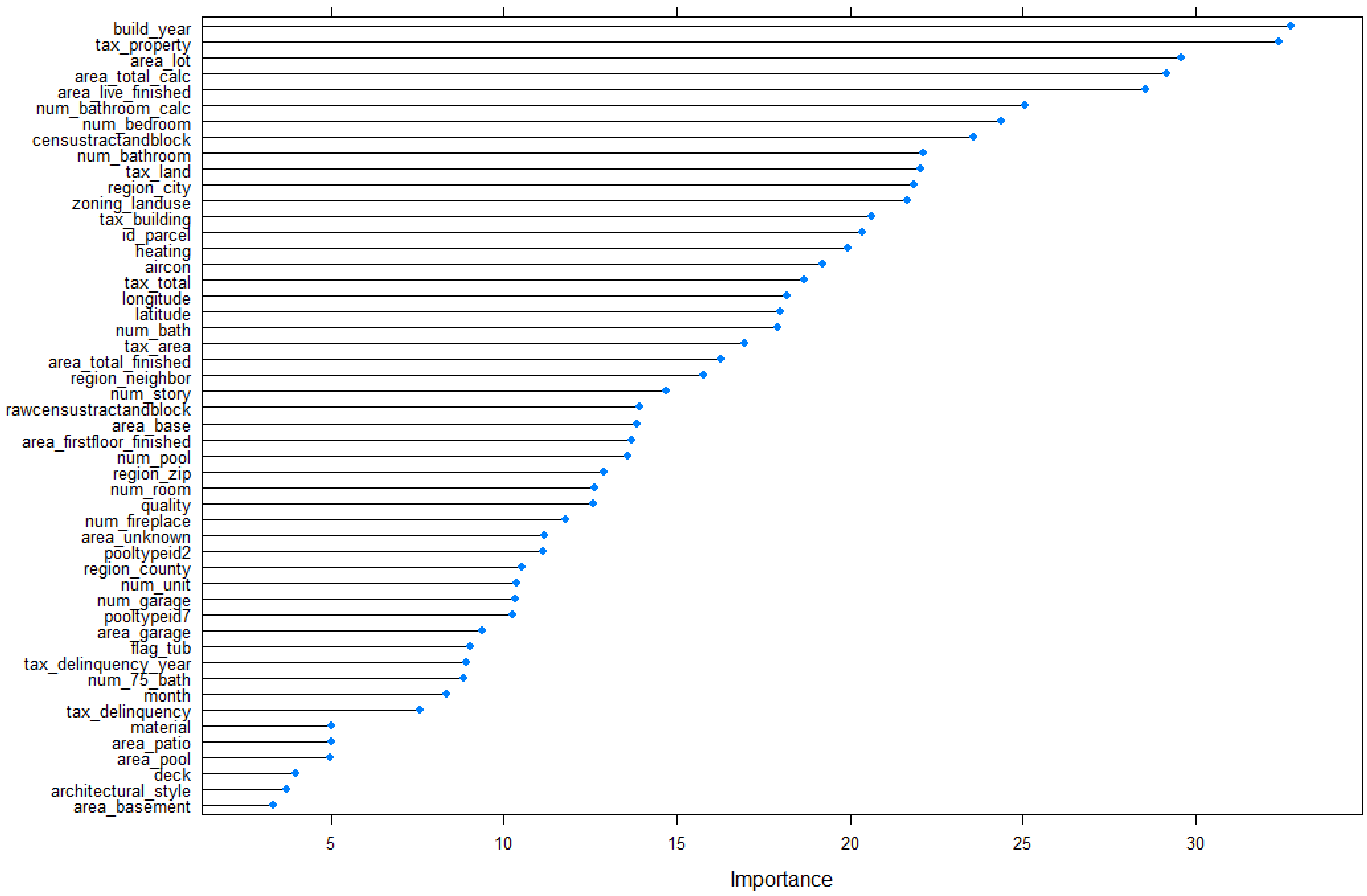


mtry	MAE
1	0.05730122
2	0.05705683
3	0.05716474

Manually tuning number of trees – imputed set 2

4 Machine Learning Models & Results

- Tree based models (Random Forest)



4 Machine Learning Models & Results

- Tree based models (Random Forest)

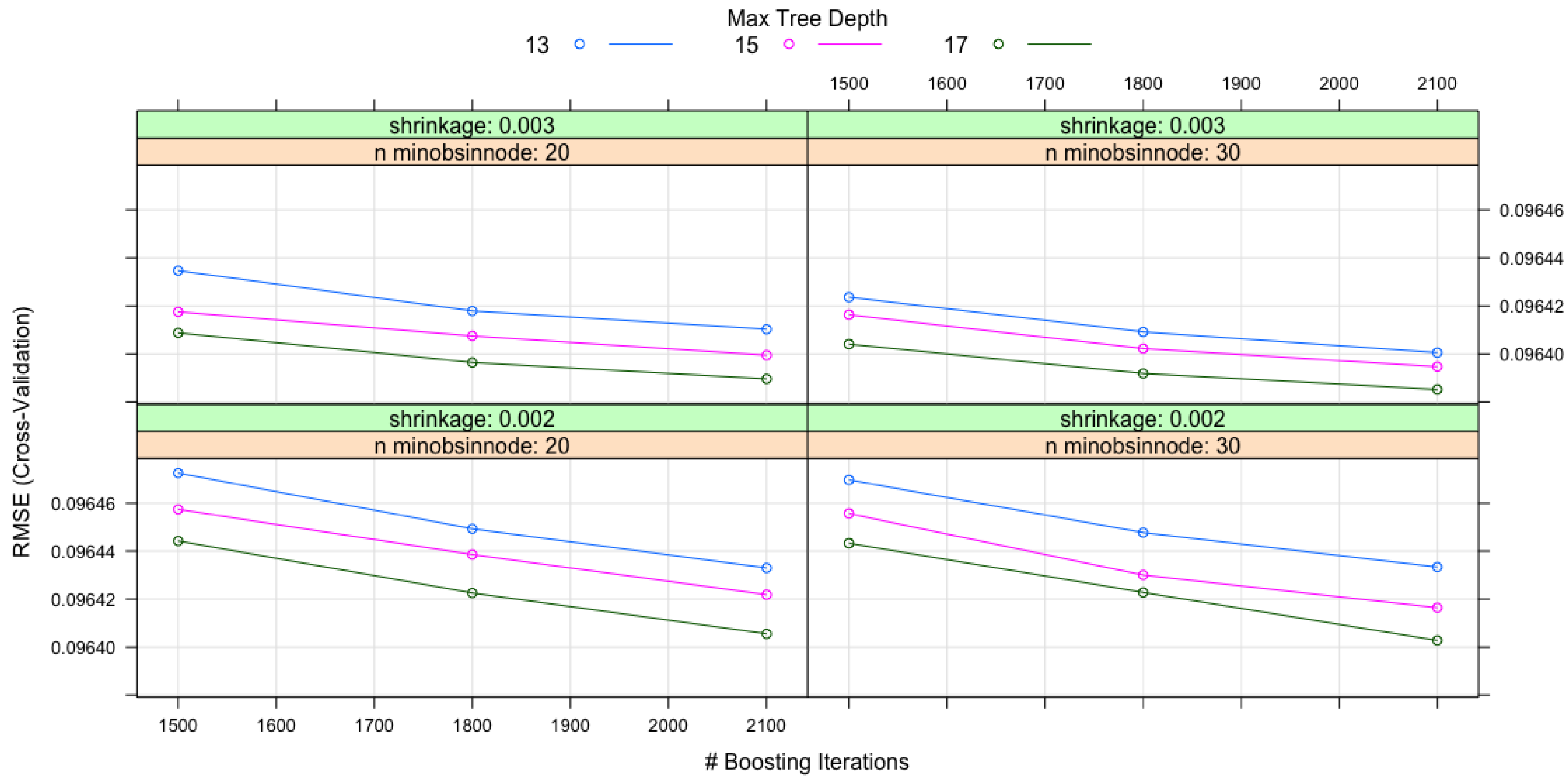
Summary

Impute set	Local CV MAE	Kaggle score
1	0.05730982	0.0647866
2	0.05738716	0.0646149

- **Pros:** Good performance, parameters relatively easy to tune
- **Cons:** SLOW

4 Machine Learning Models & Results

- Tree based models (GBM)

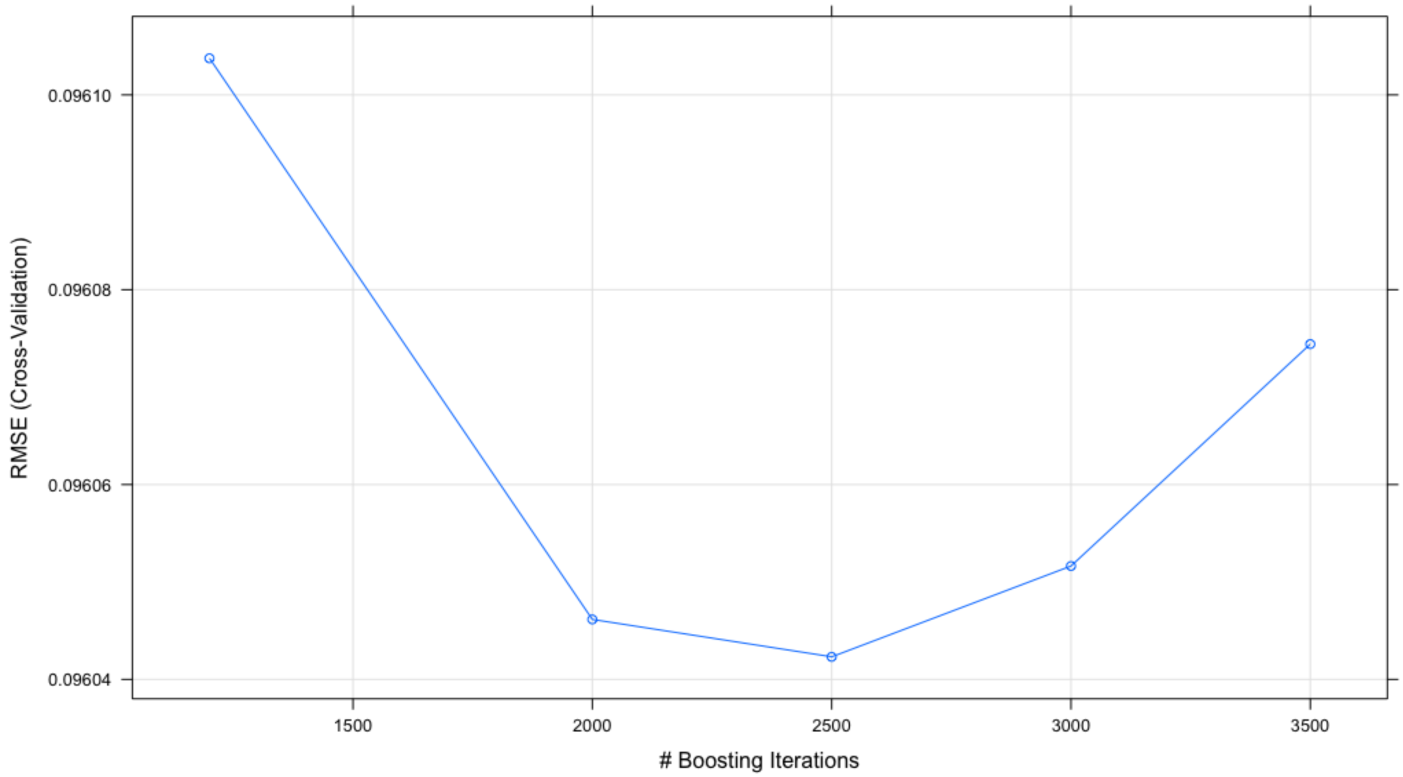


n.trees	2100
depth	17

shrinkage	0.003
minobsinnode	30

4 Machine Learning Models & Results

- Tree based models (GBM)

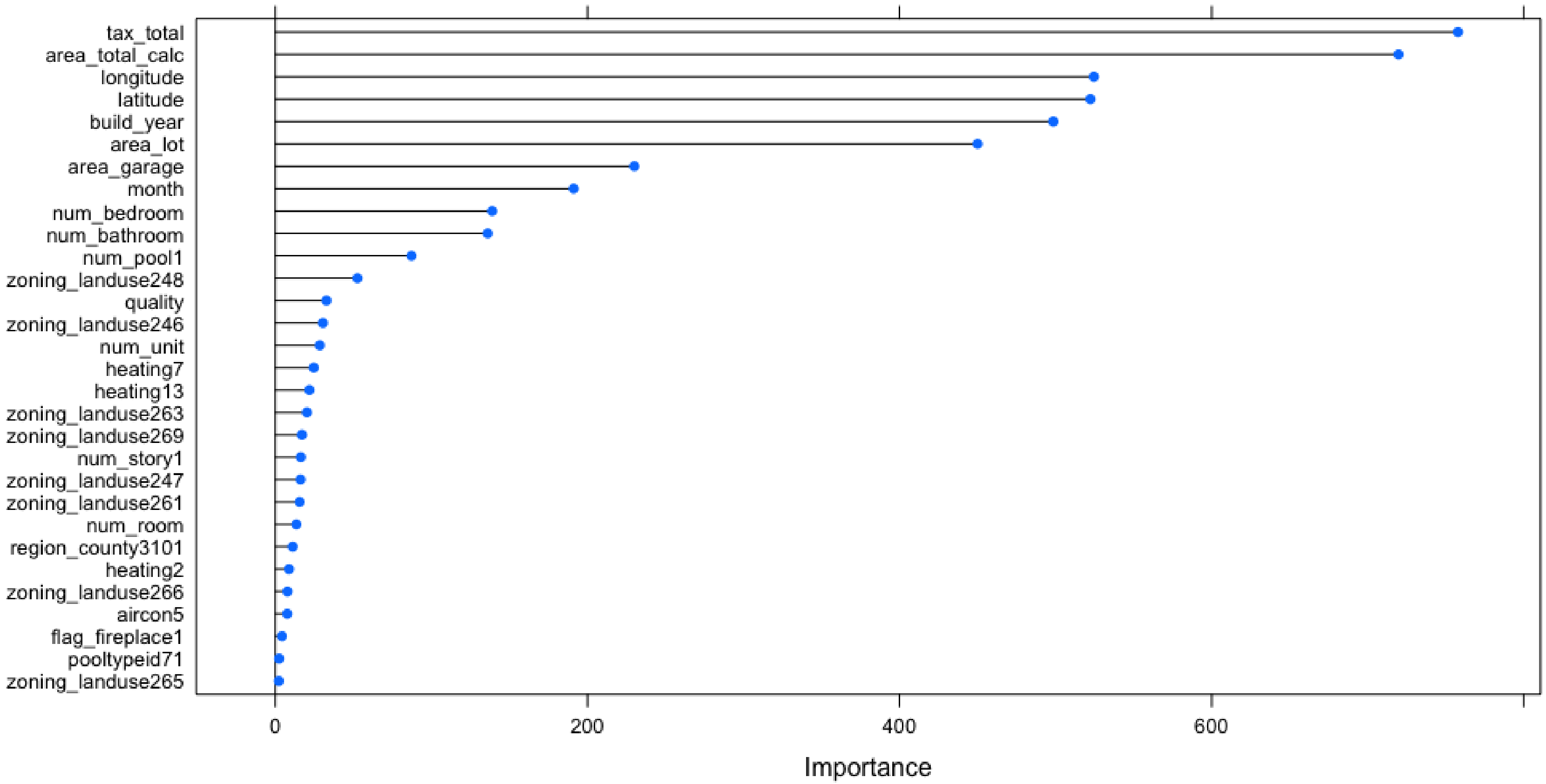


n.trees	2500
depth	20

shrinkage	0.003
minobsinnode	30

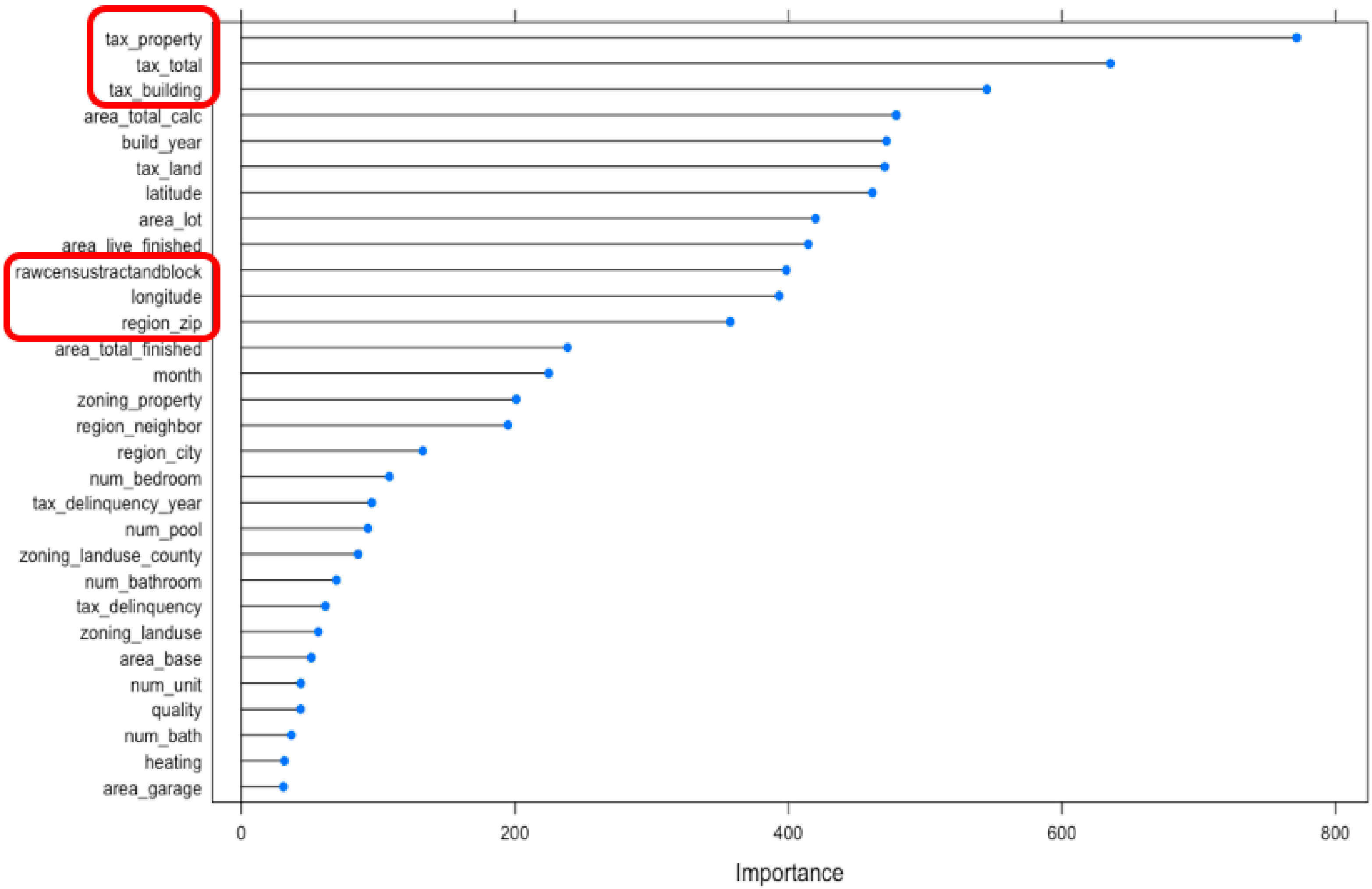
4 Machine Learning Models & Results

- Tree based models (GBM)



4 Machine Learning Models & Results

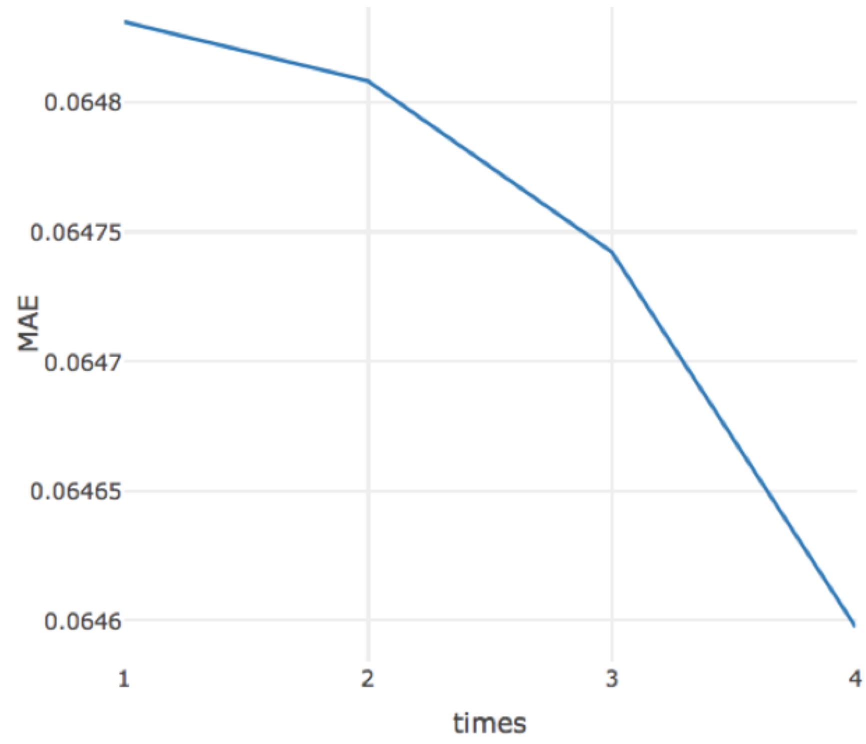
- Tree based models (GBM)



4 Machine Learning Models & Results

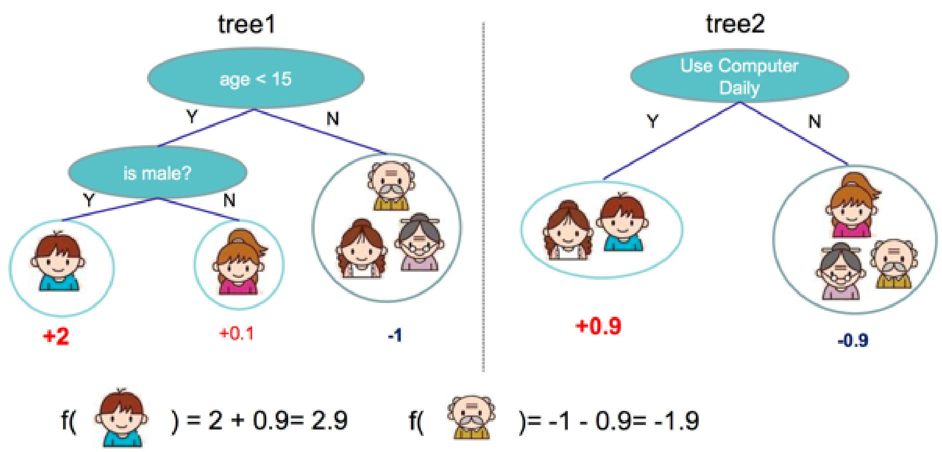
- Tree based models (GBM)

n.trees	1500	2000
depth	14	20
shrinkage	0.001	0.003
minobsinnode	35	30



4 Machine Learning Models & Results

- Tree based models (Extreme Gradient Boosting)



$$Obj = \underbrace{\sum_{i=1}^{n} l(y_i, \hat{y}_i)}_{\text{Training loss}} + \underbrace{\sum_{k=1}^{K} \Omega(f_k)}_{\text{Complexity of the Trees}}$$

- Additive Boosting

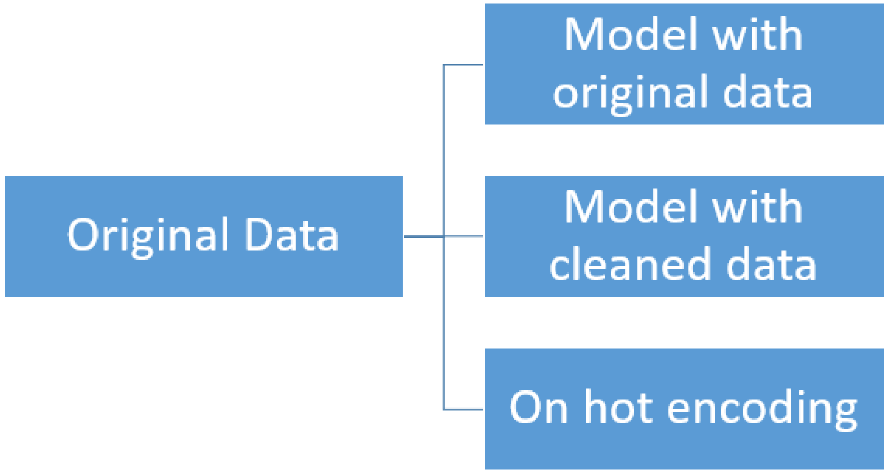
$$\begin{aligned} \hat{y}_i^{(0)} &= 0 \\ \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ &\dots \end{aligned}$$

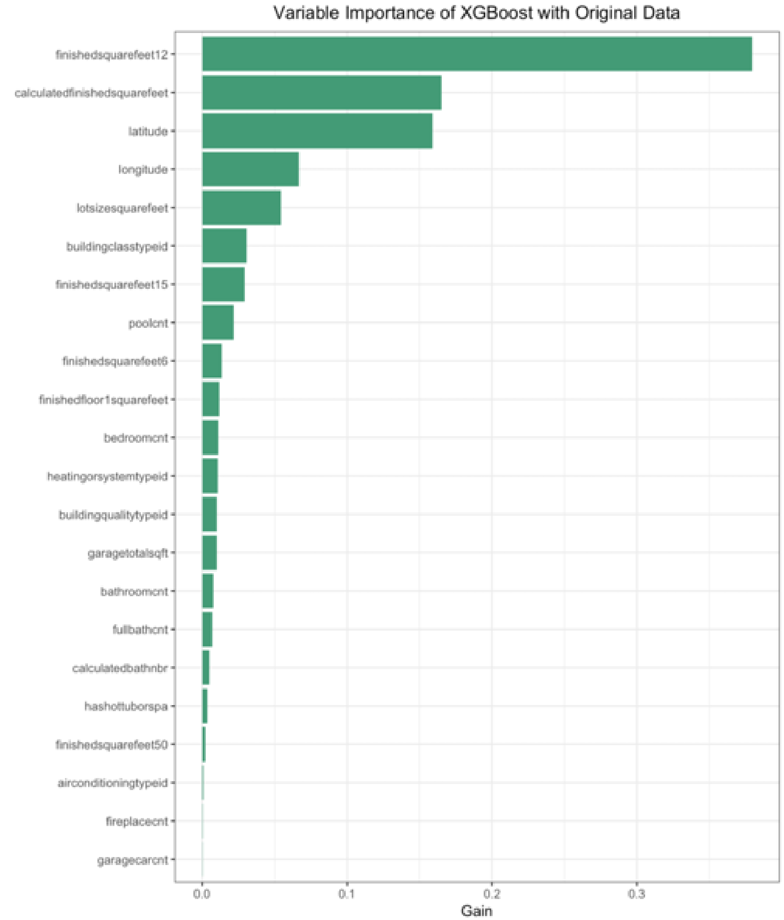
- Greedy Learning of the Tree
  - Max depth
  - Eta
  - Min_child_weight
  - Subsample
  - Colsample_bytree
  - Nround(early stop round)

$$Gain = \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda} - \gamma$$

4 Machine Learning Models & Results

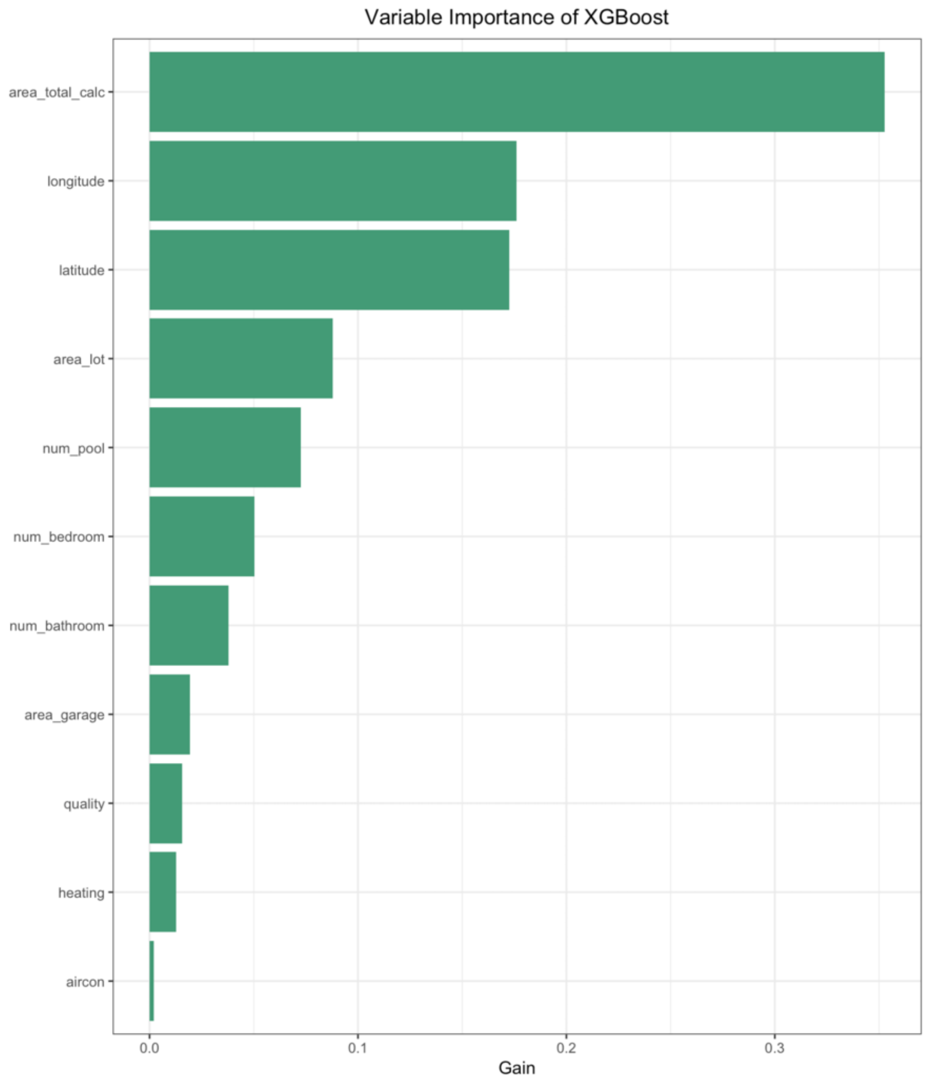
- Tree based models (Extreme Gradient Boosting)





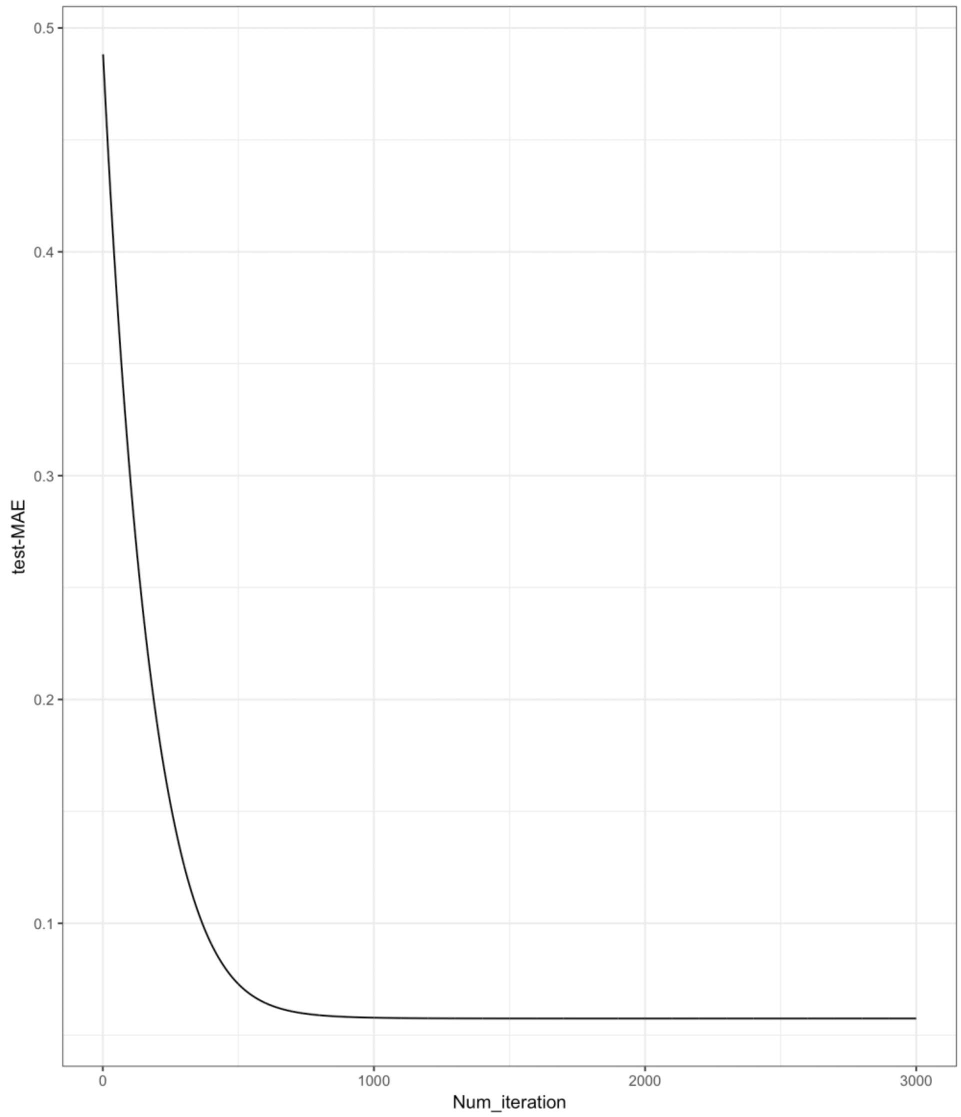
4 Machine Learning Models & Results

- Tree based models (Extreme Gradient Boosting)



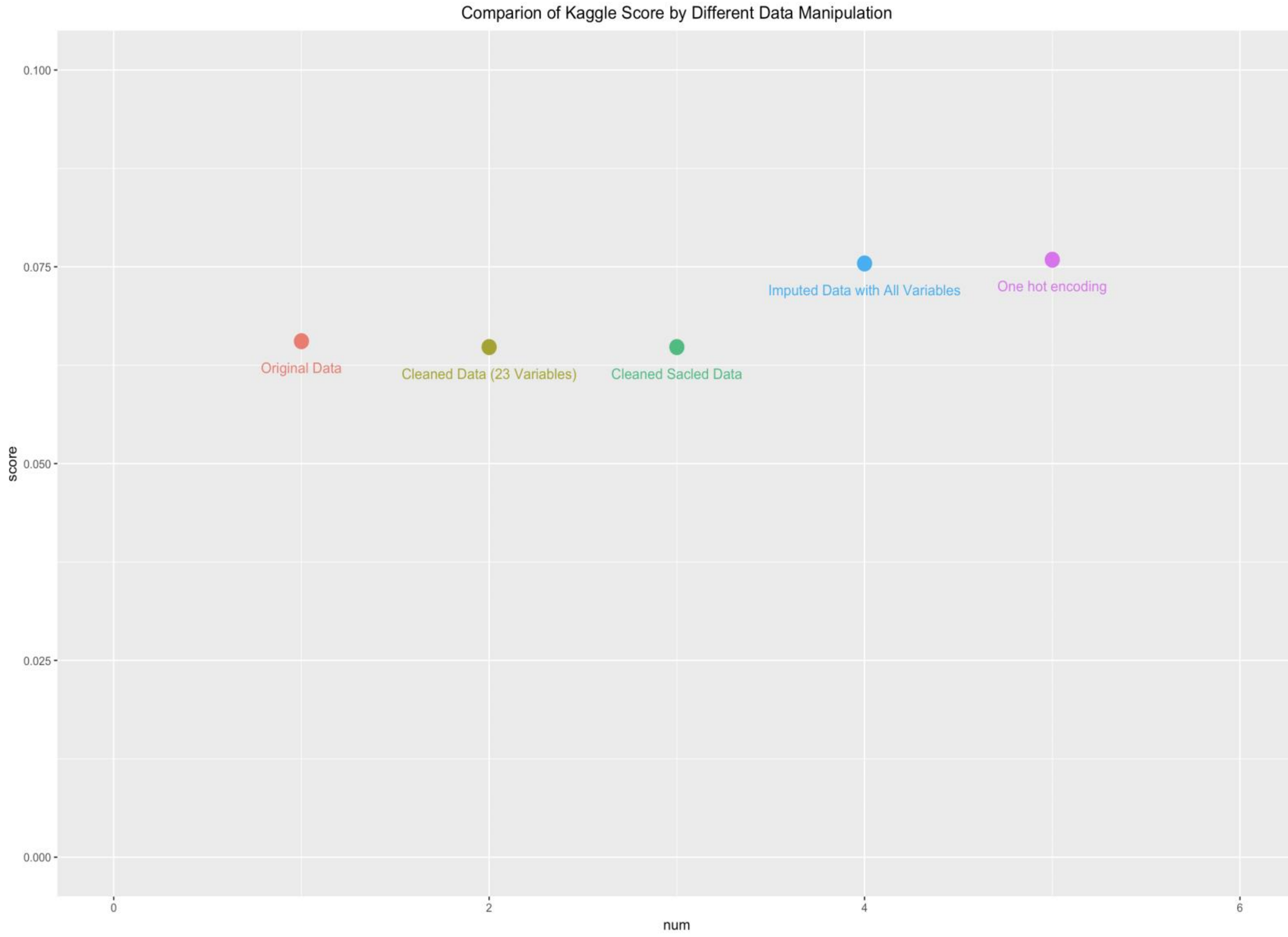
4 Machine Learning Models & Results

- Tree based models (Extreme Gradient Boosting)



4 Machine Learning Models & Results

- Tree based models (Extreme Gradient Boosting)



4 Machine Learning Models & Results

- Automatic Machine Learning (h2o)

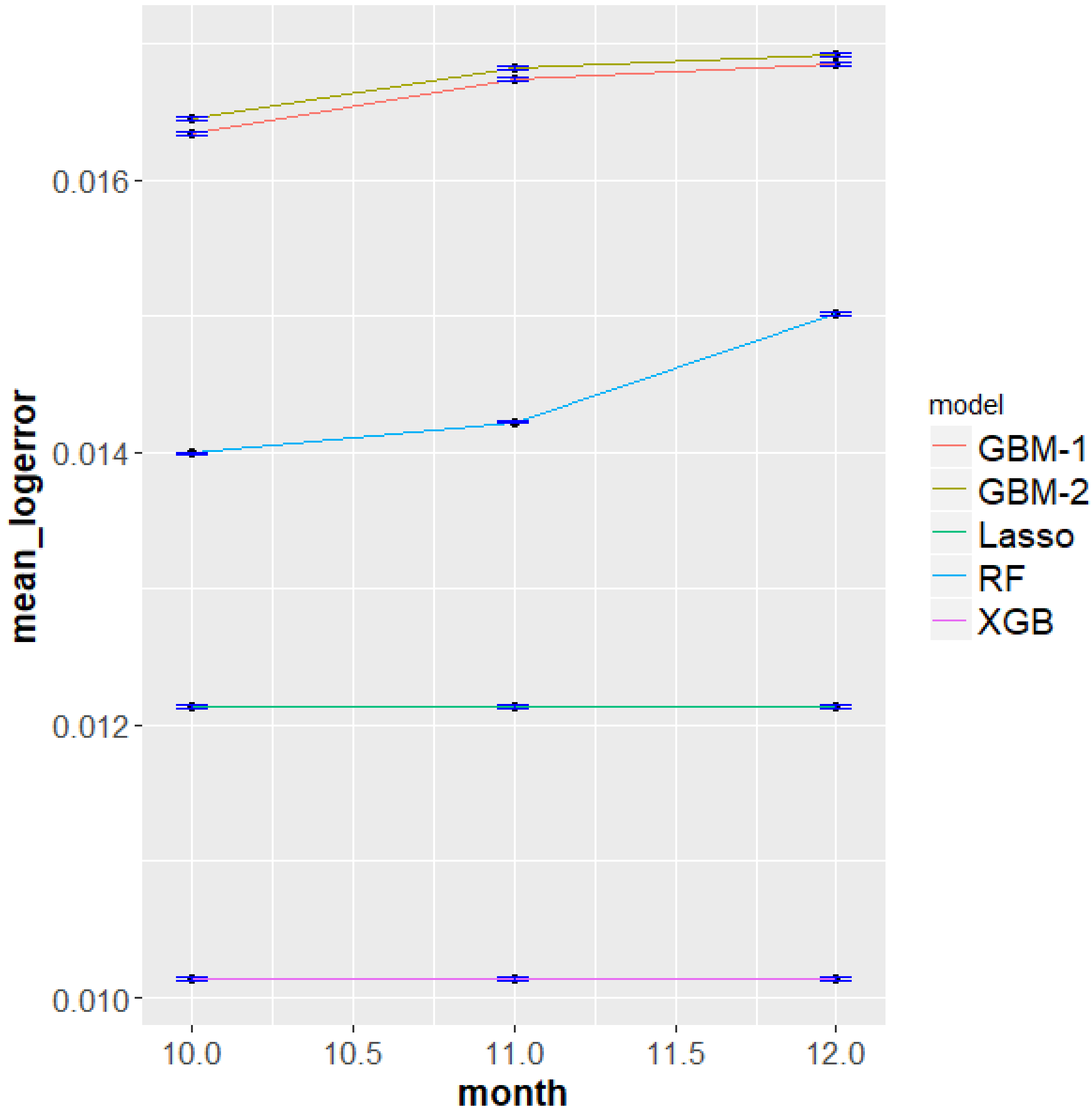
No	model_id	rmse	mae
1	DRF_0_AutoML_20170817_214820	0.137509	0.063328
2	XRT_0_AutoML_20170817_214820	0.137736	0.063418
3	StackedEnsemble_0_AutoML_20170817_214820	0.154234	0.067019
4	GBM_grid_0_AutoML_20170817_214820_model_3	0.154812	0.067172
5	GBM_grid_0_AutoML_20170817_214820_model_0	0.155727	0.067522
6	GBM_grid_1_AutoML_20170817_214820_model_0	0.155933	0.069786
7	GBM_grid_0_AutoML_20170817_214820_model_4	0.156817	0.067545
8	GBM_grid_1_AutoML_20170817_214820_model_1	0.156877	0.081081
9	GBM_grid_0_AutoML_20170817_214820_model_2	0.156933	0.067585
10	GBM_grid_1_AutoML_20170817_214820_model_5	0.157338	0.06747
11	GBM_grid_0_AutoML_20170817_214820_model_1	0.157673	0.067739
12	DL_grid_0_AutoML_20170817_214820_model_8	0.159624	0.068881
13	DL_grid_0_AutoML_20170817_214820_model_2	0.160036	0.069747
14	DL_grid_0_AutoML_20170817_214820_model_9	0.160038	0.069528
15	DL_grid_0_AutoML_20170817_214820_model_4	0.160161	0.068861
16	GBM_grid_1_AutoML_20170817_214820_model_6	0.160207	0.068089
17	GBM_grid_1_AutoML_20170817_214820_model_2	0.16028	0.068072
18	DL_grid_0_AutoML_20170817_214820_model_1	0.160551	0.068734
19	DL_grid_0_AutoML_20170817_214820_model_7	0.160553	0.067986
20	DL_grid_1_AutoML_20170817_214820_model_0	0.160577	0.06791

4 Machine Learning Models & Results

- Automatic Machine Learning (h2o) [Best Kaggle score: 0.0649128]

No	variable	relative_importance	scaled_importance	percentage	percentage
1	area_live_finished	1126.242	1	0.063897	6.3897
2	tax_total	1099.532	0.976283	0.062382	6.2382
3	build_year	1097.818	0.974762	0.062285	6.2285
4	tax_building	1094.175	0.971527	0.062078	6.2078
5	area_total_calc	1091.283	0.96896	0.061914	6.1914
6	month	1059.947	0.941136	0.060136	6.0136
7	tax_property	992.8304	0.881542	0.056328	5.6328
8	tax_land	984.9302	0.874528	0.05588	5.588
9	latitude	930.8225	0.826485	0.05281	5.281
10	longitude	886.1227	0.786796	0.050274	5.0274
11	region_neighbor	835.937	0.742235	0.047427	4.7427
12	area_lot	707.0311	0.627779	0.040113	4.0113
13	region_zip	510.8607	0.453598	0.028984	2.8984
14	num_bedroom	492.4936	0.437289	0.027942	2.7942
15	region_city	439.7923	0.390495	0.024952	2.4952
16	id_parcel	411.8048	0.365645	0.023364	2.3364
17	area_total_finished	352.1405	0.312669	0.019979	1.9979
18	quality	325.0064	0.288576	0.018439	1.8439
19	num_bathroom_calc	289.6146	0.257151	0.016431	1.6431
20	num_bathroom	280.7419	0.249273	0.015928	1.5928

4 Conclusions



4 Conclusions

- Two different imputation strategies were implemented:

    a) Variable based imputation: In this approach, every variable was individually studied and a best imputation strategy was determined by looking at the type, missingness percentage and common sense.

    b) Strategic imputation: In this approach, numerical NAs were imputed with -999 and the categorical variables were imputed with -1 or -999.

- Four different models were trained (simple to more advanced): Lasso, Random Forest, Gradient Boosting Machine, XGBoost and H2O. The best result for each model are:

Model	Lasso	rf	gbm	XGBootst	H2O
Local CV MAE	0.05723179	0.05738716	0.05283183	0.05746833	0.051023
Kaggle score	0.0649128	0.0646149	0.0645974	0.064802	0.0654966

- Our best score was obtained using gbm with the all the features from the training dataset, ranking at ~725th on Kaggle. The next best model was Random Forest which ranked 753rd at the time of submission.

- Following variables were important in these models—
  tax_property (taxamount), tax_building (structuretaxvaluedollarcnt), area_total_calc (calculatedfinishedsquarefeet), build_year (yearbuilt), tax_land (landtaxvaluedollarcnt), latitude, area_lot (lotsizesquarefeet), area_live_finished (finishedsquarefeet12), longitude

- Future work:
  - Develop a better strategy to handle categorical features
  - Feature engineering
  - Stacking: Choose best models for stacking, Models for predicting outliers, Models for different counties

THANK YOU