*This 600-word essay answers the Kaggle challenge to predict house prices in Ames, Iowa (*[*https://www.kaggle.com/c/fi-ames-housing-price-competition*](https://www.kaggle.com/c/fi-ames-housing-price-competition)*). Code was written in R.*

**Dataset:** There are 73 variables: 20 continuous, 13 discrete, 23 categorical nominal, 17 categorical ordinal. All variables could theoretically be related to the Sale\_Price, but I expect some to be more predictive than others, e.g. “Gr\_Liv\_Area,” “Neighborhood,” and “Overall\_Cond”.

Some pairs of variables are highly correlated. If I were running a regression I would consider getting rid of the “Bathrooms” variable to avoid multicollinearity.

Plotting “Gr\_Liv\_Area'' against price reveals three outliers. Two of them are purchases of large properties for a low price, likely family transactions. Random Forests are robust to outliers, but I remove them anyway because they affect measurement error. I keep all other variables and observations.

There are no missing values.

The test set, “amestest.csv,” contains the same variables as the “amestrain.csv” set but, at 1000 observations, it is smaller and it has the outcome variable (Sale\_Price) removed.

**Methods:** I used the Bagging and Random Forest methods to solve this prediction problem. I considered different values of the “mtry” tuning parameter, which indicates the number of nodes considered at each iteration, with mtry equal to 73 for bagging. I used two methods to choose the mtry value that yields the best predictions:

OOB error: I considered the Out-Of-Bag (OOB) error for RandomForests with different mtry values. I looked at mtry values of 73 (bagging), 60, 50, 40, 30, 20, 10, and 5. The OOB error was substantially lowest when with mtry tuning parameters equal to 30.

To further find the best mtry value, I compared the OOB error of RandomForests with mtry values of 26 through 39. As the OOB error differs slightly with each iteration, I averaged the OOB error for each mtry value over five runs. The lowest OOB MSE was for mtry equal to 28. However, in general, all mtry values between 27 and 37 gave good OOB MSE below 622,000,000 (see Table 1).

Five-fold cross-validation: For mtry values of 73 (bagging), 36, 18, 9, 5, 2, and 1 I used the rfcv function from the RandomForest library and the lowest MSE was for mtry equal to 36, for both 5-fold and 10-fold cross-validation (see Table 2). For mtry values of 73 (bagging) and 36 through 27 I used custom code and the five mtry values with lowest MSE were 27, 29, 32, 33, and 34 (see Table 3).

**Findings:** For the final model I used an mtry value equal to 32, as this value was indicated by both tuning methods.

I used two measures of variable importance for my best Random Forest model. First is the mean decrease of accuracy in predictions on the out of bag samples when a given variable is excluded from the model (‘%IncMSE’). The second is the average total reduction in node impurity that results from splits over that variable (‘IncNodePurity’). There were six variables that were among the five most important according to either of these measures (see Table 4).

Four of them describe property size. Although, as expected, those variables are highly correlated, each adds new information to the model, allowing it to make better predictions. The two other important variables capture the quality of the property beyond size, through neighborhood and year of construction. Surprisingly, the year of sale was not important, despite the dataset spanning the crisis of 2008.

Lastly, I checked if the model can be improved by running a Random Forest using only the six most important variables. I found that the OOB error of a model using six variables is significantly higher, and therefore decided against the smaller model.

**TABLES**

**Table 1**

The OOB MSE were as follows:

mtry OOB MSE

73 636377653

60 636393301

50 629051874

40 626738284

38 623610129

37 625342378

36 **621708118**

35 624426895

34 **621510666**

33 623808157

32 **619919494**

31 627608523

30 624237514

29 **621470078**

28 **617064574**

27 **621081778**

26 **621242449**

20 623738168

10 647396496

5 707335692

**Table 2**

mtry 5-fold CV MSE 10-fold CV MSE

73 691174431 665472643

36 **649590244 638208181**

18 678533376 677350011

9 818704489 770283551

5 938008497 941774487

2 1355960296 1342472058

1 3637339348 3599472336

**Table 3**

mtry 5-fold CV MSE

73 687340379

36 657710120

35 664362735

34 **653986322**

33 **650585903**

32 **655630618**

31 661346890

30 656072335

29 **654610927**

28 659141030

27 **655329010**

**Table 4**

%IncMSE IncNodePurity

Gr\_Liv\_Area 44.3361017 1.431532e+12

Neighborhood 39.2909168 2.801469e+12

First\_Flr\_SF 30.0686564 5.899208e+11

Total\_Bsmt\_SF 26.6953572 7.268592e+11

Garage\_Cars 22.0018466 1.747014e+12

Year\_Built 19.0147570 1.252380e+12

Gr\_Liv\_Area: total living area excluding the basement and garage. This is a continuous variable.

Neighborhood: nominal variable describing property’s physical location within city limits.

First\_Flr\_SF: living area on the first (i.e. ground) floor. This is a continuous variable.

Total\_Bsmt\_SF: total area of the basement. This is a continuous variable.

Garage\_Cars: this discrete variable describes how many cars fit in the garage.

Year\_Built: this discrete variable describes the original construction date.