Best Model for Predicting House Prices in Ames, Iowa

Team: J-A-G
Jennifer Thuy Nguyen
Aurora Yucong Hu
Garrett Hastings

01

Introduction

Dataset
Problem
Concerns

02

Data Cleaning

Missing data
Variable Distribution assessment
Dummy variables transformation

03

Data Mining Techniques/Algorithms

Variable filtering
Linear regression Techniques
Non-linear Regression

04

Conclusion

Model comparison
Interpretation
Takeaways - Application

01

Introduction

Dataset
Problem of Interest
Concerns

02

Data Cleaning

Missing data
Variable Distribution assessment
Dummy variables transformation

03

Data Mining Techniques/Algorithms

Variable filtering
Linear regression Techniques
Non-linear Regression

04

Conclusion

Model comparison
Interpretation
Takeaways - Application

Introduction

Our data is a collection of variables about houses in Ames, Iowa (80 independent variables and house sale price) (www.kaggle.com).

Goal: To find the best model to predict the Selling Price from the given housing features

Possible predictors: neighborhood, square footage of the lot, number of bedrooms, year built, etc.

Possible concerns: missing values, highly skewed variables (high number of zero's), categorical variable handling, and computational speed

01

Introduction

Dataset
Problem
Concerns

02

Data Cleaning

Missing data
Variable distribution assessment
Dummy variables transformation

03

Data Mining Techniques/Algorithms

Variable filtering
Linear regression Techniques
Non-linear Regression

04

Conclusion

Model comparison
Interpretation
Takeaways - Application

Data Cleaning

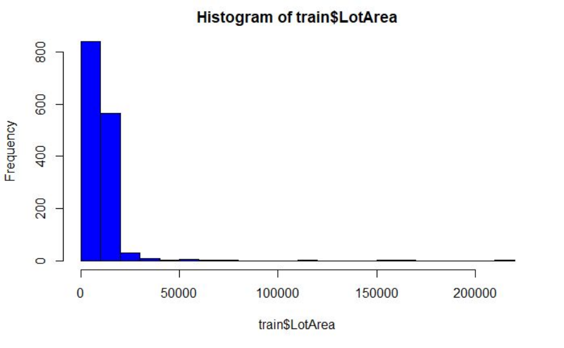
- Missing categorical entries
  - Add a new level called "Missing" to store all of the NA's
- Missing numerical variables
  - Replace with median values

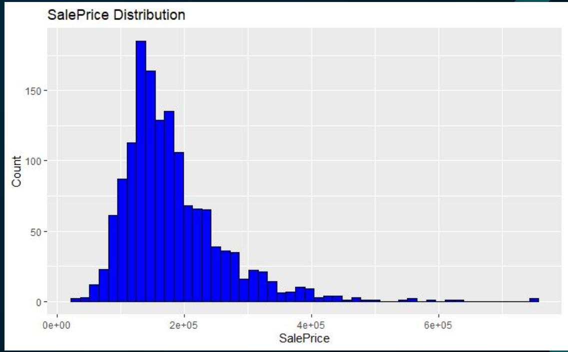
```
summary(train$Alley)

Grvl Pave NA's
  50   41 1369
```

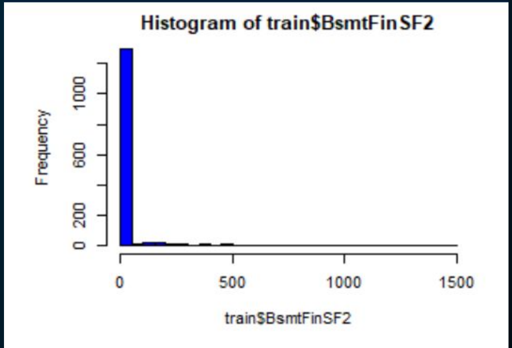
```
summary(train$Alley)

   Grvl    Pave Missing
     50      41    1369
```

Data Cleaning

- Variable Distribution Assessment - Skewness
  - Log transformation on “LotArea” and “Sale Price” (response variable)
  - Categorization of several numerical variables into “0” or “More than 0” (or “1” or “More than 1”)







Data Cleaning

- Dummy variables transformation
  - For each categorical variable, we turned it into multiple dummy variables (each dummy represents one sub-category)
  - Number of independent variables increases from 80 to 314

```r
library(dummies)
train_off <- dummy.data.frame(train_off, sep = ".")
```

01

Introduction

Dataset
Problem of Interest
Concerns

02

Data Cleaning

Missing data
Variable Distribution assessment
Dummy variables transformation

03

Data Mining Techniques/Algorithms

Variable filtering
Linear regression Techniques
Non-linear Regression

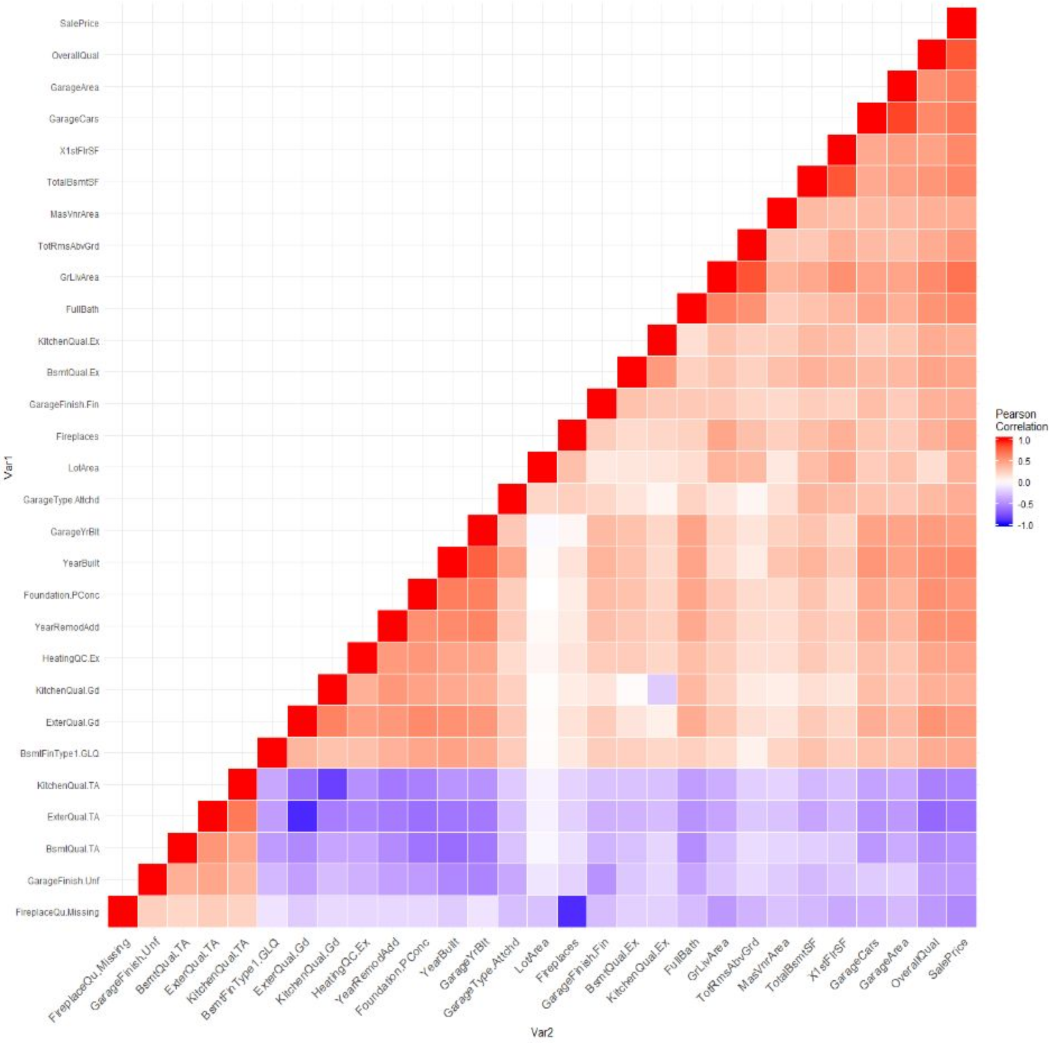
04

Conclusion

Model comparison
Interpretation
Takeaways - Application

Variable Filtering

- Only select predictors with moderately high correlation with response y = SalePrice
  - Corr(X,Y) > 0.4
  - From 314 predictors down to 28 predictors



Linear Regression Techniques

Subset Selection: Best Subset Selection

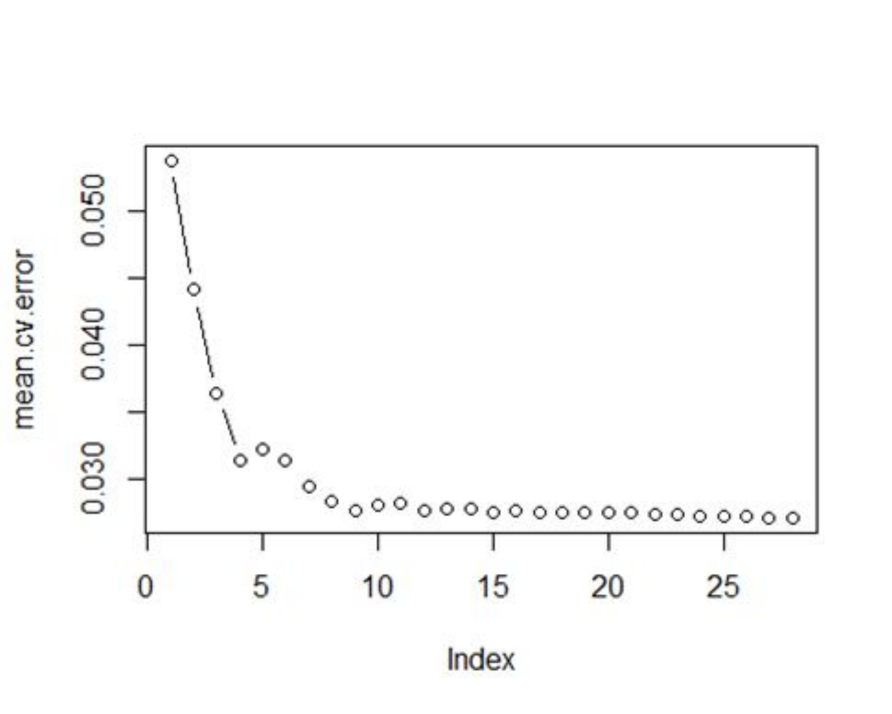
Regularization: Lasso & Ridge

Dimension Reduction: PCR and PLS

Best Subset Selection

Test MSE: 0.0188, 9 predictors

- **Pros:**
  - Has simple fitting procedure
  - Gives sparse model (feature selection)
  - Assesses all possible subset of variables
  - Presents the best candidate for a least-squared model with q variables
- **Cons:**
  - Takes a long time to process large models; computationally expensive

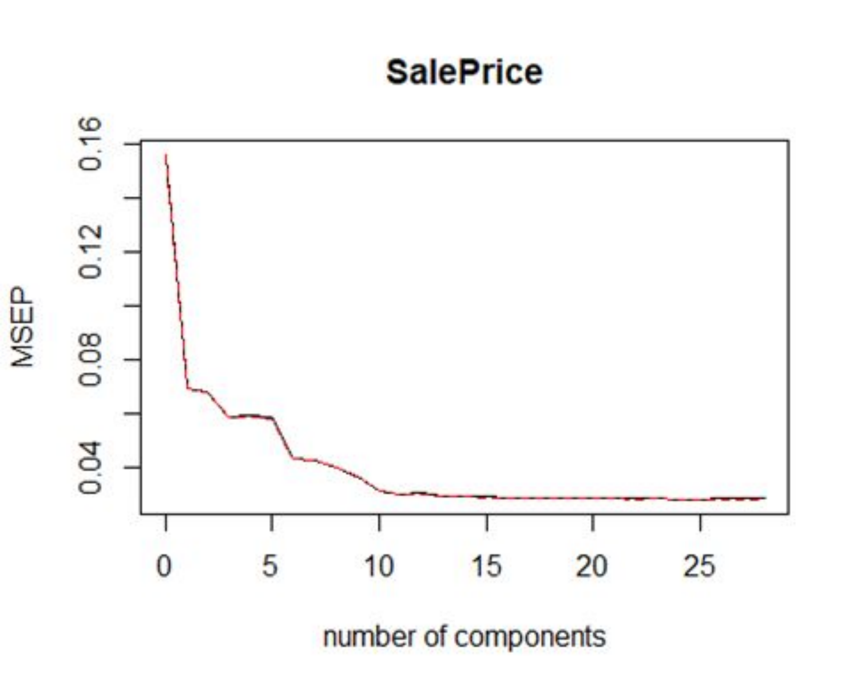


Principal Component Regression

Creates new components from linear combinations of original variables such that they capture as much variability in the predictors as possible

- Pros:
  - Reduces data dimension
  - When the number of components is small, overfitting can be avoided
- Cons:
  - Does not yield feature selection
  - The first M principal components, though may best explain the predictors, are not necessarily predictive of the response

Test MSE: 0.0216
28 Variables (10 PCs)



Partial Least Squares

A *supervised* alternative to PCR - PLS approach attempts to find directions that help explain both the predictors AND the response.

Pros:

- All the pros of PCR
- The supervised dimension reduction can reduce bias

Cons:

- Does not yield feature selection
- The supervised dimension reduction can increase variance => will not perform that much better than PCR

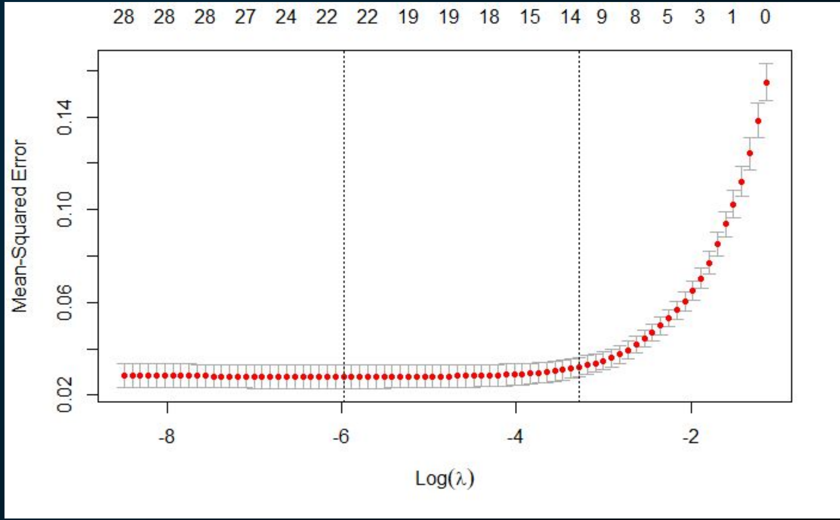
Test MSE: 0.0201
28 Variables (9 components)

Lasso

Test MSE: 0.0184, 23 predictors

Pros

- Eliminates many variables in its model (sparse)
- Can create flexible models that do not rely on hierarchies, unlike forward and backward subset
- Gives better predictions than Variable filtering and fwd/bwd stepwise



Cons

- Interpretability - why does it select certain variables and not others?
- Complicated model-fitting procedure (hard to do without statistical software)

The best Log(Lambda) = -5.978623
23 predictors in best model

Ridge

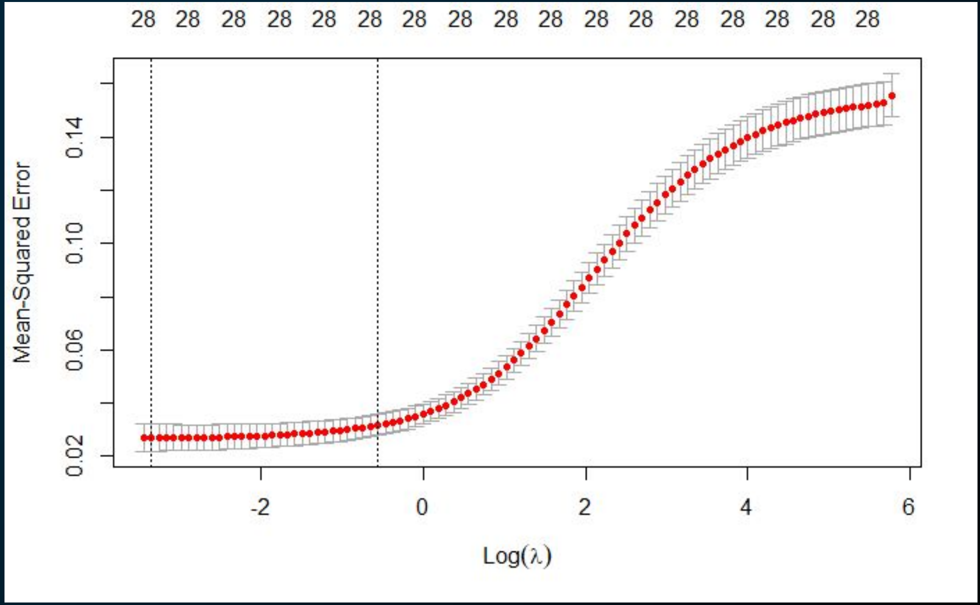
Test MSE: 0.0191, 28 predictors

Pros

- Can create flexible models that do not rely on hierarchies, as opposed to forward and backward subset selection
- Gives better performance than Lasso if all variables are significant

Cons

- Does not eliminate any variables (as opposed to Lasso)
- Can also lead to high variance due to no variable reduction (high flexibility)



Best log(lambda) = -3.35042

Non-linear Regression Techniques

K Nearest Neighbors

Regression Tree

Bagging & Random Forest

Boosting

K-nearest neighbors

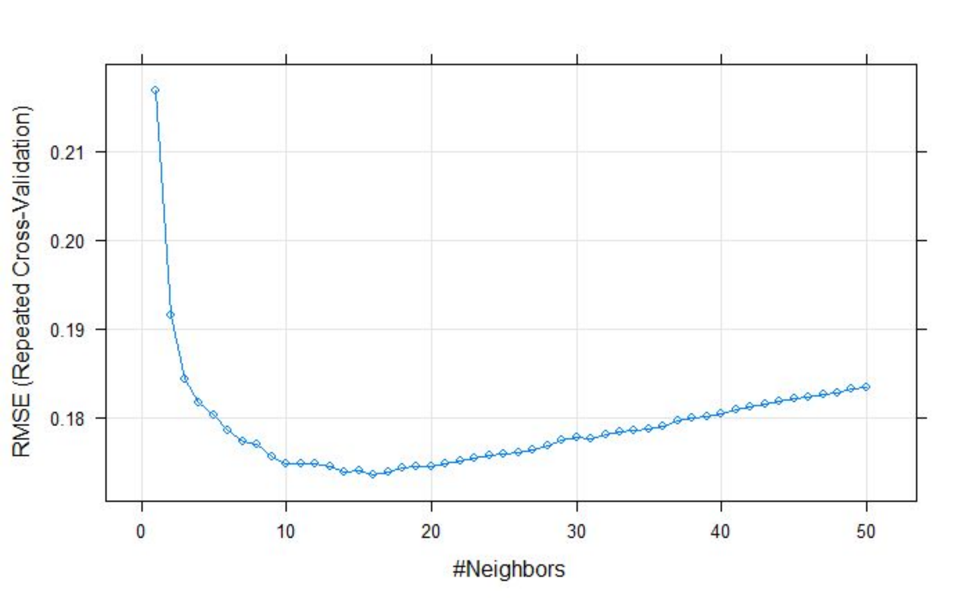
Test MSE: 0.0264, k=16

Pros

- Non-parametric, more flexible
- Offers a more accurate model if the true shape is non-linear
- Simple fitting process

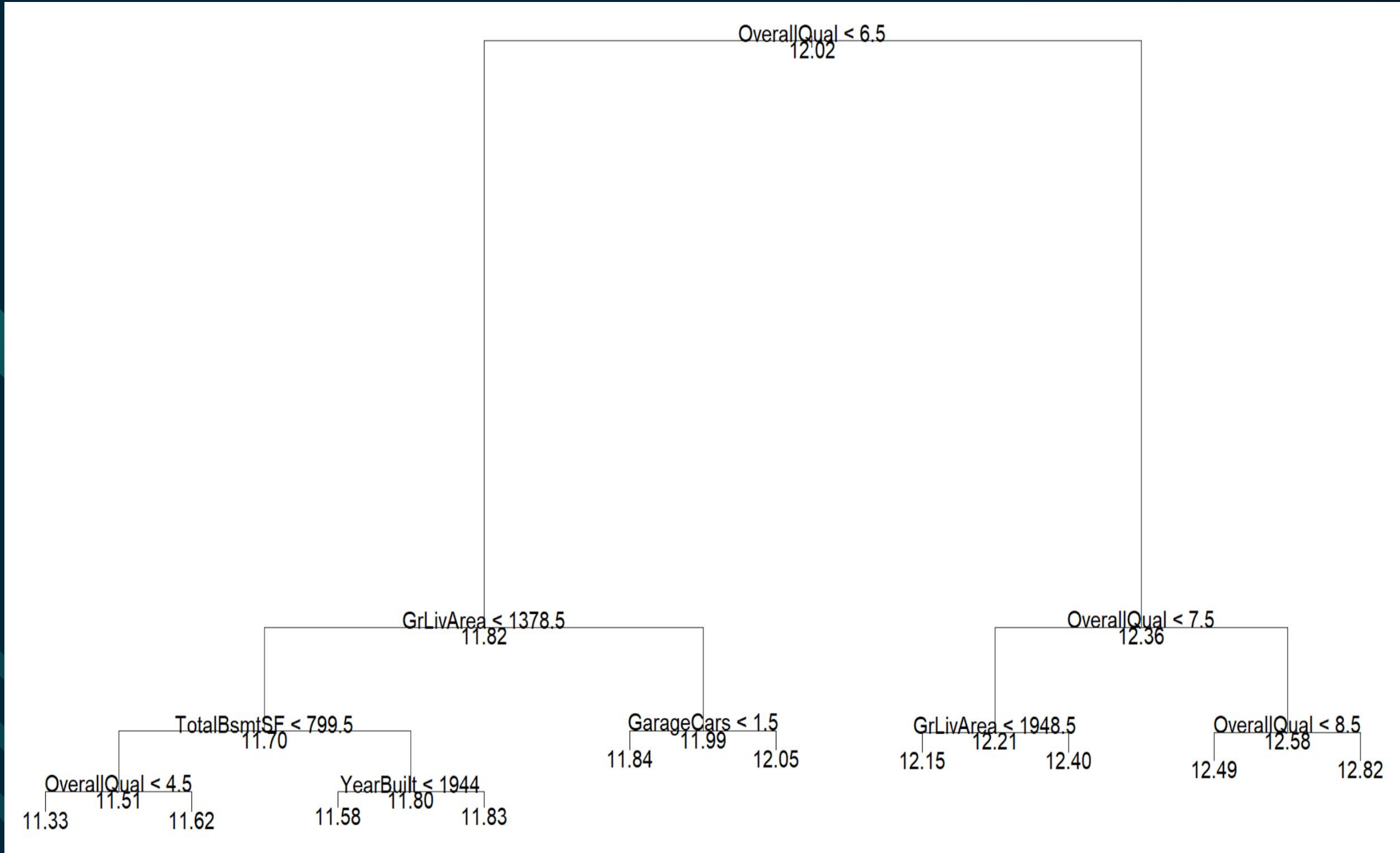
Cons

- Rarely outclass parametric approaches
- Does not work well with high dimensions
- Difficult to identify importance of variables
- Sensitive to noisy data, missing values and outliers



Regression Tree

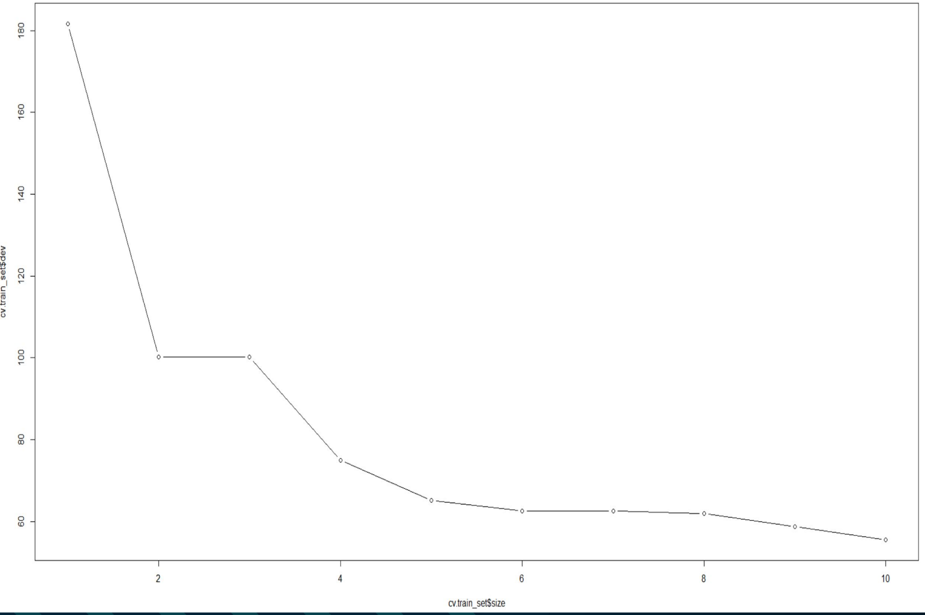
Test MSE: 0.0443

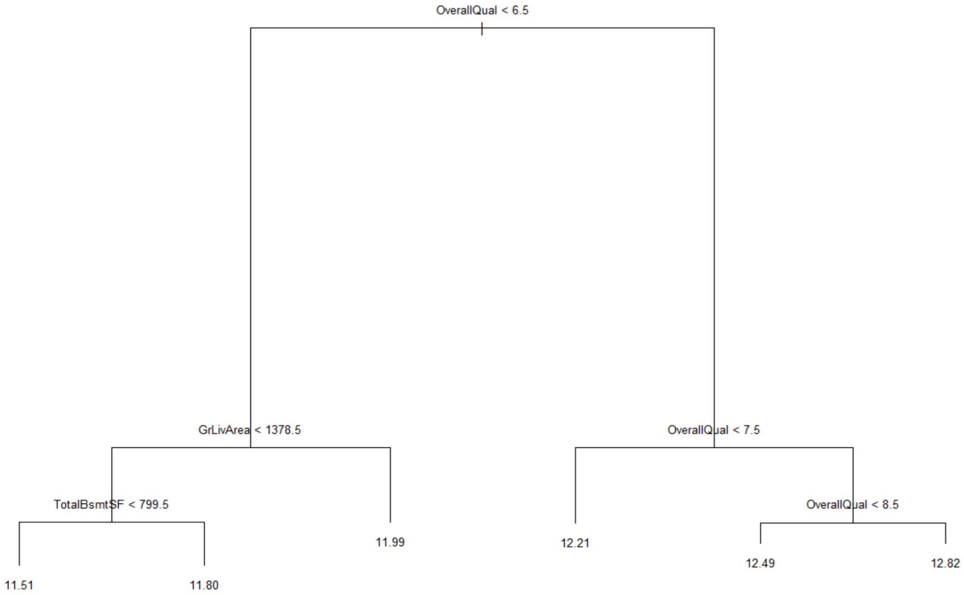


Regression Tree

```
# for more interpretability and overfiting concerns do some pruning within 6-9 terminal nodes
prune.train_set = prune.tree(tree.train_set, best = 6)
```

After pruning (Test MSE): 0.0553





Regression Tree

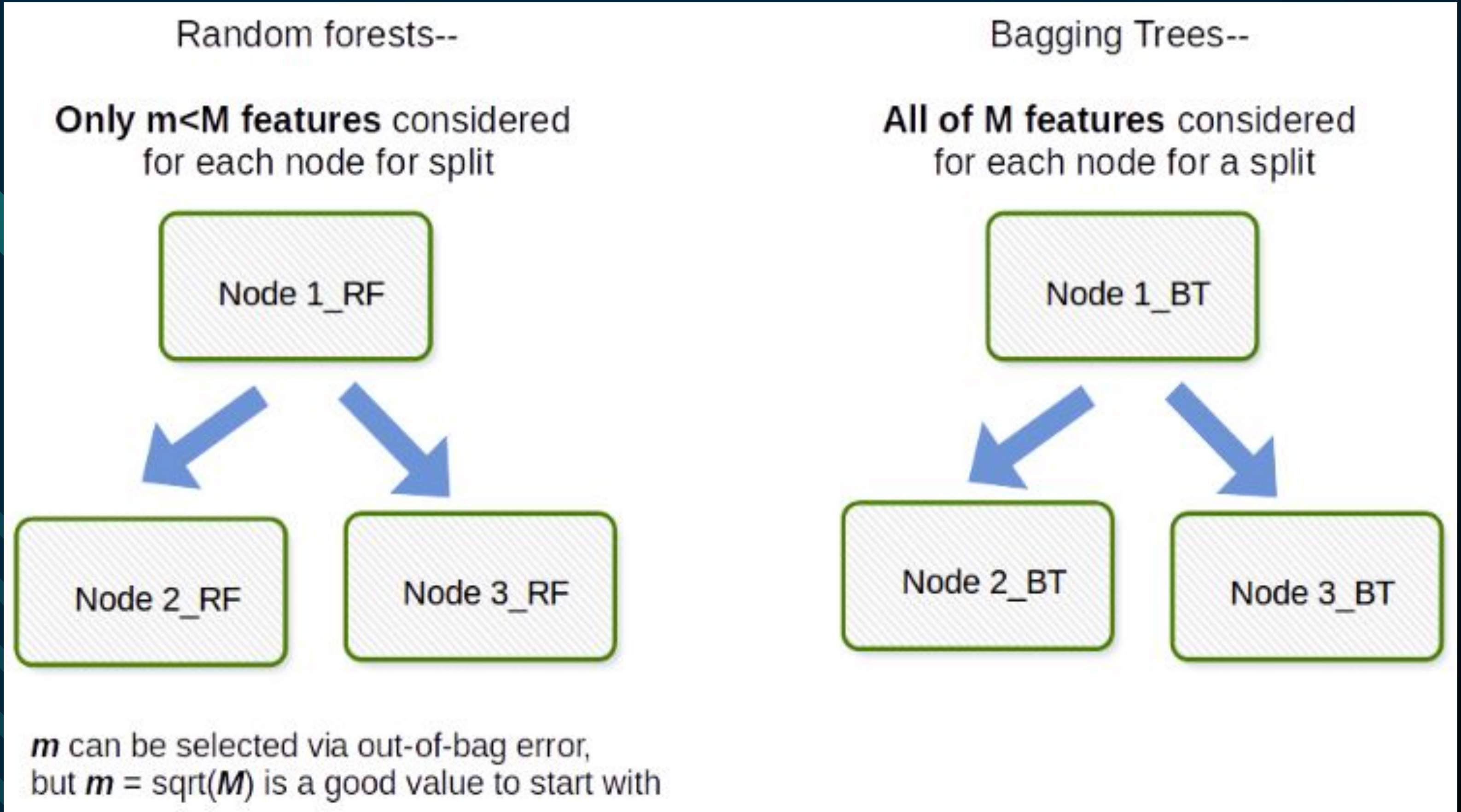
Pros

- Interpretability & visual representation
- Numerical and categorical features accommodation
- Little data preprocessing
- Feature selection happens automatically

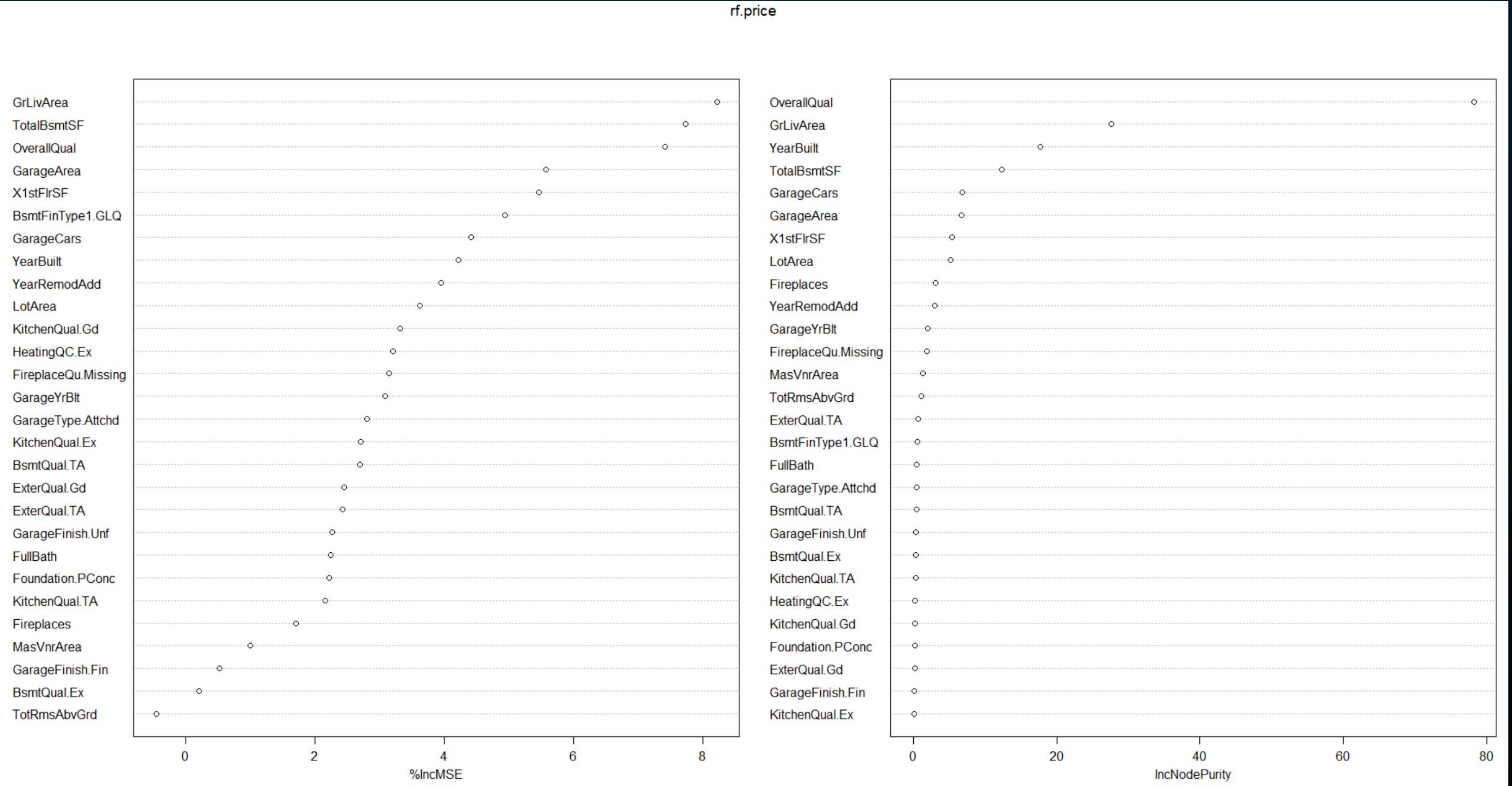
Cons

- Inflexible: dynamic model adjustment
- Unstable
- Overfitting, which can be mitigated by:
  - Limiting tree depth
  - Minimal # of objects in leaves
  - Tree pruning

Bagging and Random Forest



Random Forest



Bagging and Random Forest

Test MSE: 0.0194 -- ntree=500, mtry=28
Test MSE: 0.0207 -- ntree=25, mtry=28
Test MSE: 0.0200 -- ntree=25, mtry=20 (RF)

Pros

- Impressive in versatility
- Parallelizable
- Robust to outliers and nonlinear data
- Low bias, moderate variance

Cons

- Complexity
- High computational resources requirement
- Overfit --- solved by tuning hyperparameters

Boosting

Gradient Boosting

- Fits a new predictor in the residuals committed by the preceding predictor
- By combining one weak learner to the next learner, the error is reduced significantly over time

Tuning Parameters

- N.minobsinnode = c(10,15)
- Interaction.depth = c(1,3)
- N.tree = c(1000, 1500)
- Shrinkage = c(0.05,0,1)

Boosting

```
```{r, include = FALSE}
# hyper-parameter tuning of gradient boosting
set.seed(7)
grid <- expand.grid(n.trees = 900, interaction.depth=2, shrinkage=c(0.05,0.1),n.minobsinnode=c(10,15))
ctrl <- trainControl(method  = "cv",number  = 10)
```

Tuning Parameters

- N.minobsinnode = c(10,15)
- Interaction.depth = c(1,3)
- N.tree = c(1000, 1500)
- Shrinkage = c(0.05,0,1)

Test MSE: 0.0173, all predictors

- N.minobsinnode = 10
- Interaction.depth = 2
- N.tree = 900
- Shrinkage = 0.05
```

Boosting

Pros

- Easy to read and interpret
- Resilient method that curbs over-fitting easily

Cons

- Sensitive to outliers
- Almost impossible to scale up

01

Introduction

Dataset
Problem of Interest
Concerns

02

Data Cleaning

Missing data
Variable Distribution assessment
Dummy variables transformation

03

Data Mining Techniques/Algorithms

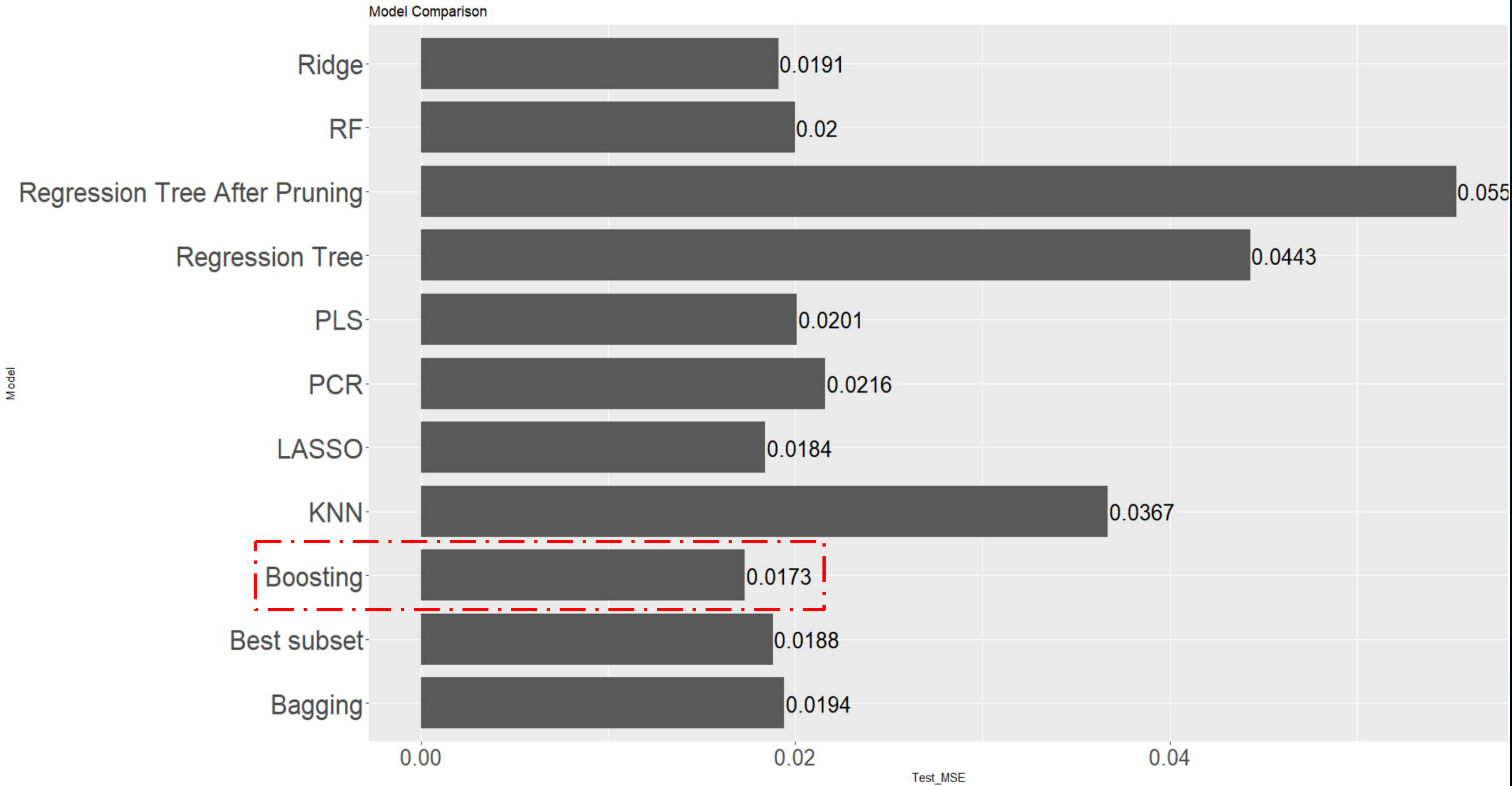
Variable filtering
Linear regression Techniques
Non-linear Regression

04

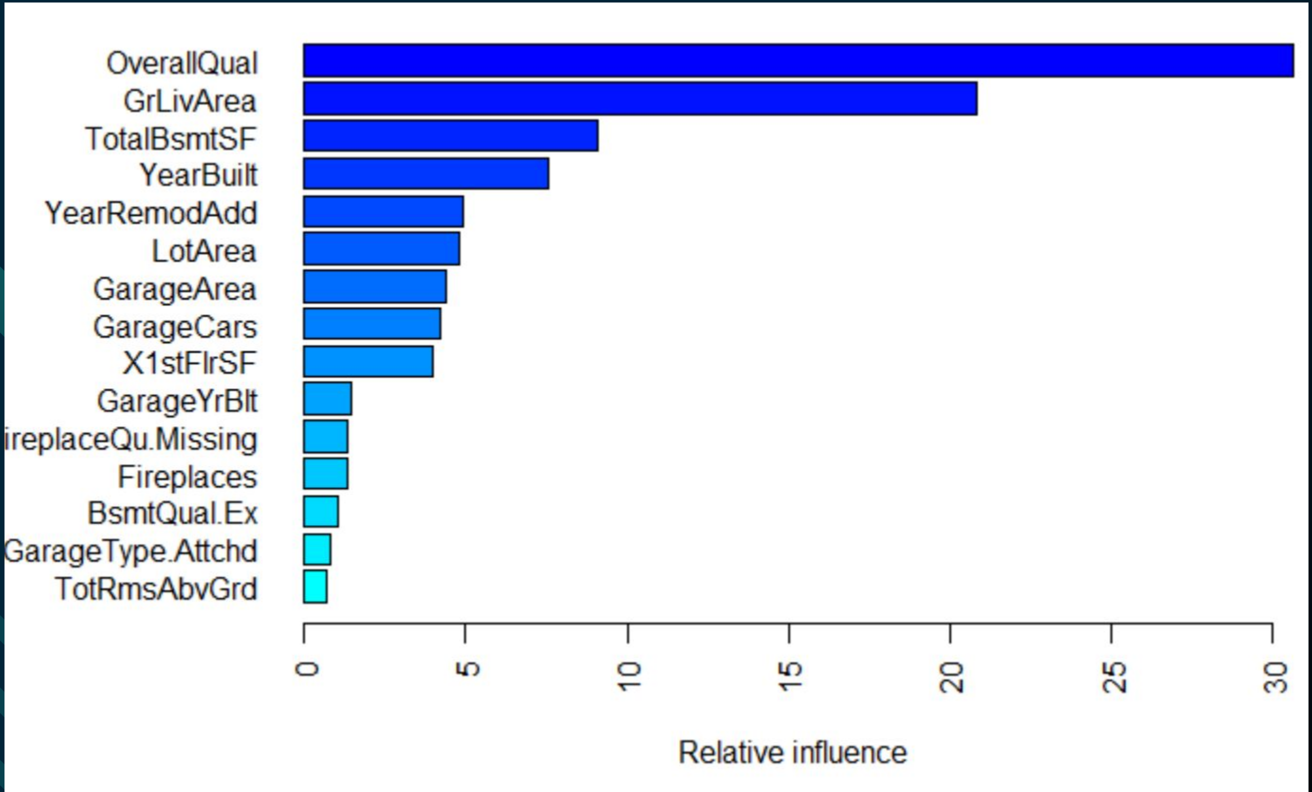
Conclusion

Model comparison
Interpretation
Takeaways - Application

Model Comparison - Test MSE



Most important variables from Boosting model



Conclusions

- Best method: Gradient Boosting
- Performance Accuracy: 86% on average
- Most important variables
    - OverallQual - Overall material and finish quality
    - GrLivArea: Above grade (ground) living area square feet
    - TotalBsmtSF: Total square feet of basement area
    - YearBuilt: Original construction date
- Surprises: No location indicator; Garage-related features importance
- Improvement: Better handling of high dimension next time without variable filtering

Thanks!

Do you have any questions?