Kaggle Challenge: Predicting Housing Prices

BEN BRUNSON, NICHOLAS MALOOF, AARON OWEN, JOSH YOON

Introduction

Challenge: Predicting Housing Prices in Ames, Iowa using various machine learning techniques

Data:

- Train Data Set**:** 1460 Observations x 80 Variables (Including Response Variable: Sale Price)
- Test Data Set: 1459 Observations x 79 Variables

Useful Links:

- **Kaggle Homepage:** https://www.kaggle.com/c/house-prices-advanced-regression-techniques
- **Data Description:** https://storage.googleapis.com/kaggle-competitions-data/kaggle/5407/data_description.txt

Understanding the Data

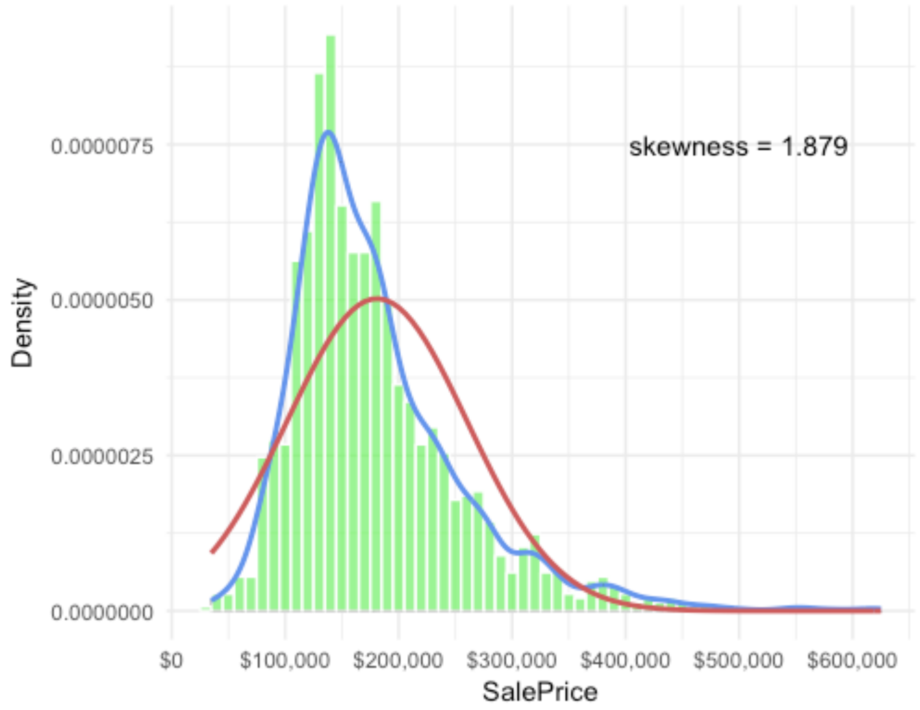
Total Predictor Variables Provided: 79

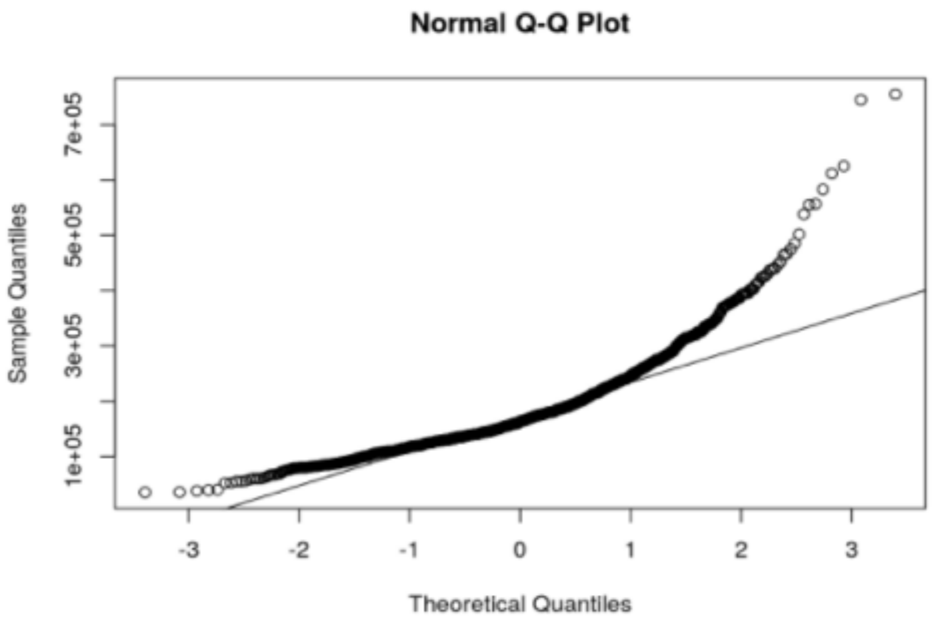
- Continuous Variables: 28
- Categorical Variables: 51

Combined test and train data sets to get a holistic view of each variable

- (i.e., total missing values, total categories in categorical variable)

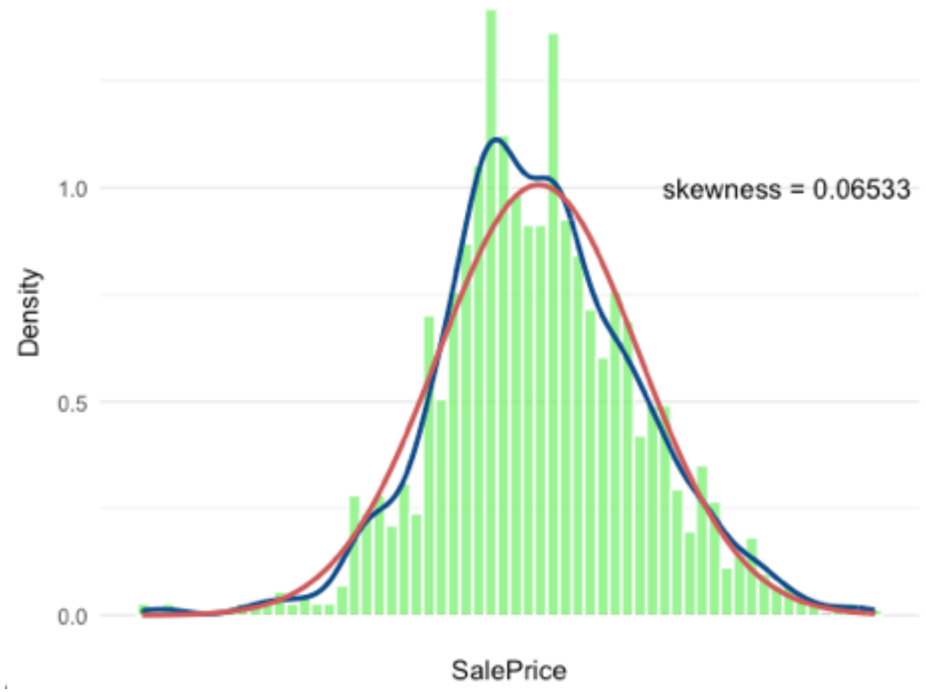
Processing the Data: Response Variable

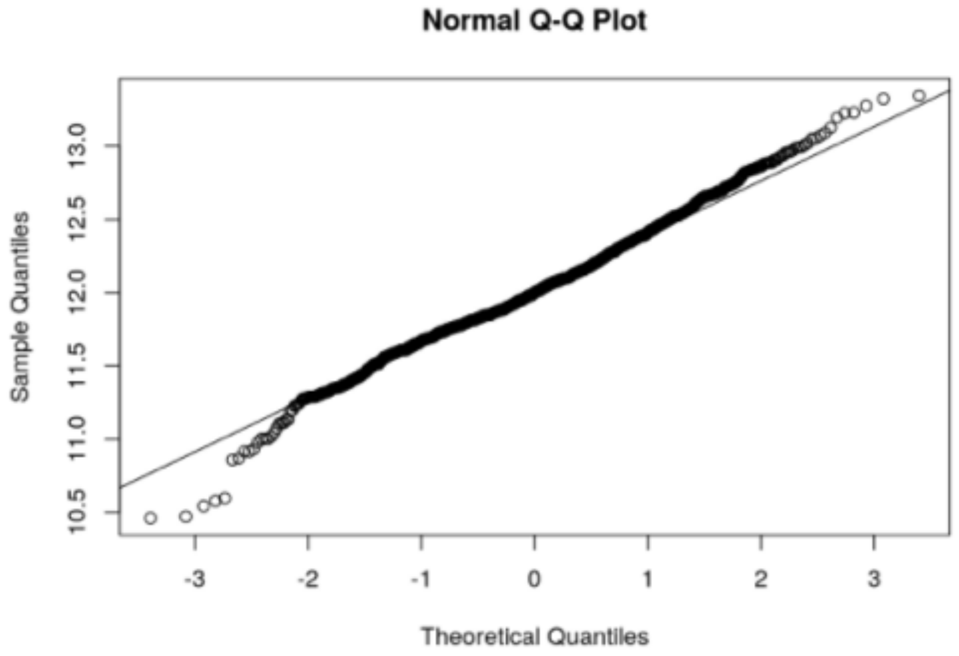




Processing the Data: Response Variable

Treat response variable with log + 1 transformation





Remember to inverse log before submitting to Kaggle

Processing the Data: Overview of Missingness

34 predictors with missing values

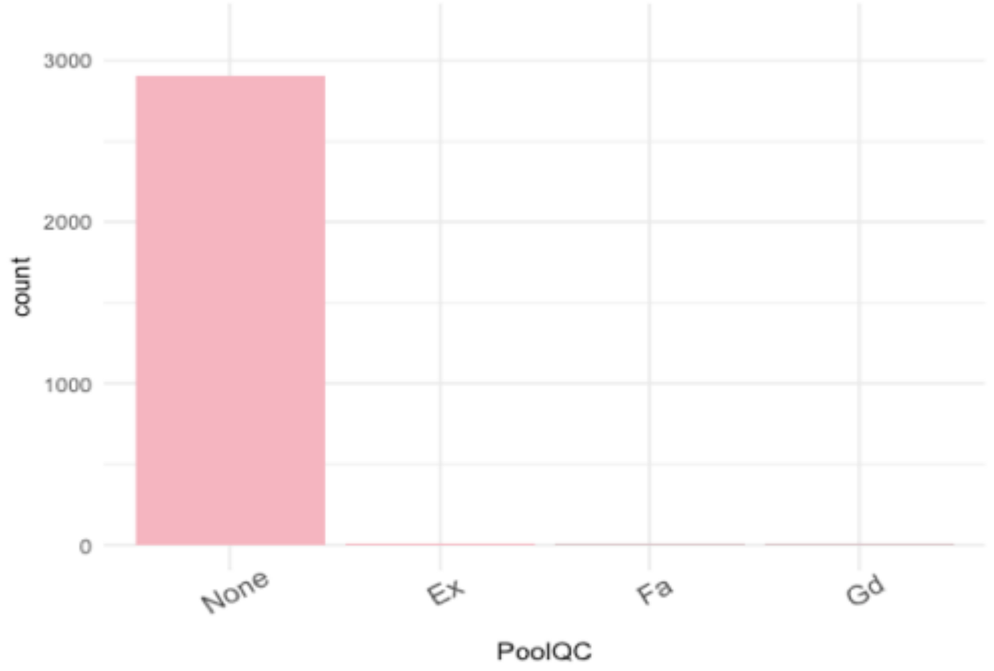
```
##       PoolQC  MiscFeature        Alley        Fence  FireplaceQu
##         2909         2814         2721         2348         1420
##  LotFrontage  GarageYrBlt GarageFinish   GarageQual   GarageCond
##          486          159          159          159          159
##   GarageType     BsmtCond BsmtExposure     BsmtQual BsmtFinType2
##          157           82           82           81           80
## BsmtFinType1   MasVnrType   MasVnrArea     MSZoning    Utilities
##           79           24           23            4            2
## BsmtFullBath BsmtHalfBath   Functional  Exterior1st  Exterior2nd
##            2            2            2            1            1
##   BsmtFinSF1   BsmtFinSF2    BsmtUnfSF  TotalBsmtSF   Electrical
##            1            1            1            1            1
##  KitchenQual   GarageCars   GarageArea     SaleType
##            1            1            1            1
```

Processing the Data: Handling Missing Data

1) Are data really missing?

Ex. Pool Quality

- 2909 out of 2919 observations have ”NA” values
  - Most NAs are due to houses not having pools

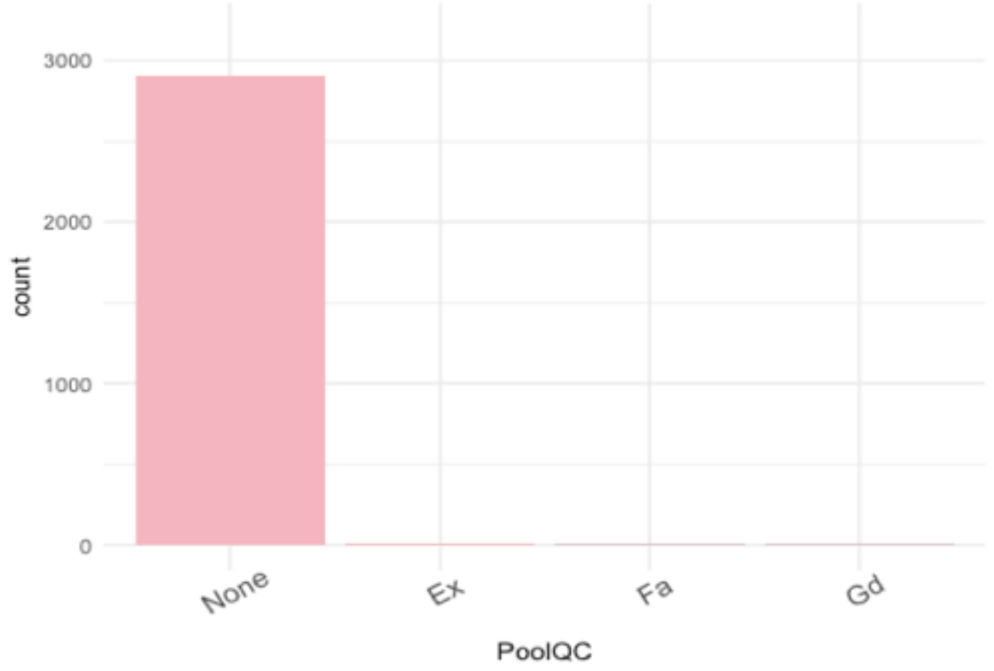


Processing the Data: Handling Missing Data

1) Are data really missing?

Ex. Pool Quality

- 2909 out of 2919 observations have ”NA” values
  - Most NAs are due to houses not having pools
- Solution:
  - Replace (most) NAs with new category: “None”



Processing the Data: Handling Missing Data

2) Not all NA values indicate a missing feature

Ex. Pool Quality

```
##         PoolQC PoolArea
## 2421    <NA>        368
## 2504    <NA>        444
## 2600    <NA>        561
```

```
##    PoolQC        mean counts
##     <chr>       <dbl>  <int>
## 1      Ex 359.7500000      4
## 2      Fa 583.5000000      2
## 3      Gd 648.5000000      4
## 4    <NA>   0.4719835   2909
```

Processing the Data: Handling Missing Data

2) Not all NA values indicate a missing feature

Ex. Pool Quality

```
##        PoolQC PoolArea
## 2421     <NA>      368
## 2504     <NA>      444
## 2600     <NA>      561
```

```
##    PoolQC         mean counts
##     <chr>        <dbl>  <int>
## 1      Ex 359.7500000      4
## 2      Fa 583.5000000      2
## 3      Gd 648.5000000      4
## 4    <NA>   0.4719835   2909
```

- <u>Solution</u>: Use related numerical variable to impute categorical variable
  - Calculate average area of each pool class within Pool Quality and fill for NAs

Processing the Data: Handling Missing Data

2) Not all NA values indicate a missing feature

Ex. Sale Type (1 Missing observation, but we know Sale Condition)

```
##
##          COD  Con ConLD ConLI ConLw  CWD  New  Oth   WD
##  Abnorml   46    0     3     2     0    1    0    5  133
##  AdjLand    0    0     0     0     0    0    0    0   12
##  Alloca     0    0     0     0     0    0    0    0   24
##  Family     2    0     1     2     1    1    0    1   38
##  Normal    39    4    21     5     7   10    0    1 2314
##  Partial    0    1     1     0     0    0  239    0    4
```

Processing the Data: Handling Missing Data

2) Not all NA values indicate a missing feature

Ex. Sale Type (1 Missing observation, but we know Sale Condition)

- Solution: Use related categorical variables to impute
  - For Sale Condition that is "Normal" we see by far most common Sale Type value is "WD" and we can impute.

```
##
##             COD  Con ConLD ConLI ConLw  CWD  New  Oth   WD
##   Abnorml    46    0     3     2     0    1    0    5  133
##   AdjLand     0    0     0     0     0    0    0    0   12
##   Alloca      0    0     0     0     0    0    0    0   24
##   Family      2    0     1     2     1    1    0    1   38
##   Normal     39    4    21     5     7   10    0    1 2314
##   Partial     0    1     1     0     0    0  239    0    4
```

Processing the Data: Handling Missing Data

3) Use domain knowledge

Ex. Lot Frontage (486 NAs)

- Houses in close proximity likely have similar lot areas

```
##    Neighborhood median
##           <chr>  <dbl>
##  1      Blmngtn   43.0
##  2      Blueste   24.0
##  3       BrDale   21.0
##  4      BrkSide   51.0
##  5      ClearCr   80.5
##  6      CollgCr   70.0
##  7      Crawfor   70.0
##  8      Edwards   65.0
##  9      Gilbert   64.0
## 10       IDOTRR   60.0
## # ... with 15 more rows
```

Processing the Data: Handling Missing Data

3) Use domain knowledge

Ex. Lot Frontage (486 NAs)

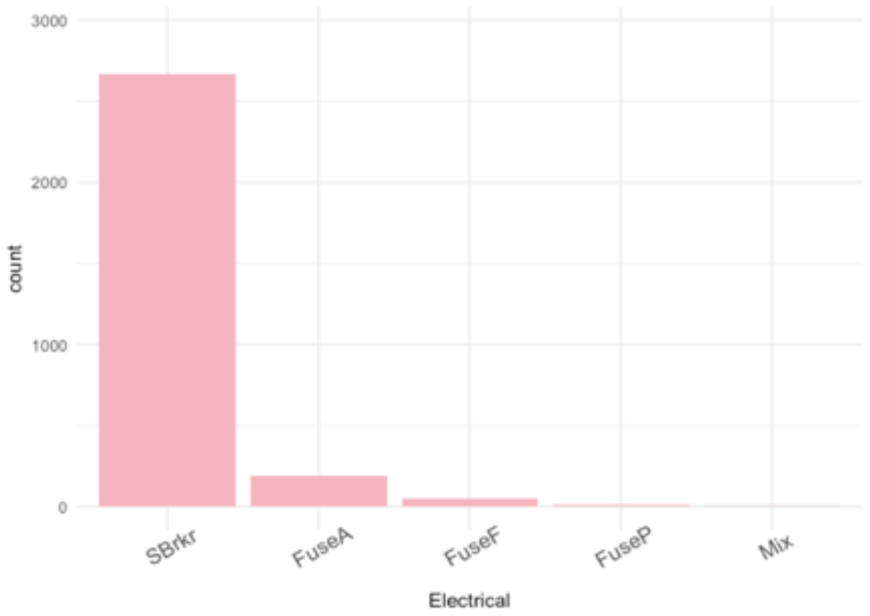
- Houses in close proximity likely have similar lot areas
- Solution: use categorical variable to impute numerical
  - Use median Lot Area by neighborhood to impute missing value

```
##     Neighborhood median
##            <chr>  <dbl>
##  1       Blmngtn   43.0
##  2       Blueste   24.0
##  3        BrDale   21.0
##  4       BrkSide   51.0
##  5       ClearCr   80.5
##  6       CollgCr   70.0
##  7       Crawfor   70.0
##  8       Edwards   65.0
##  9       Gilbert   64.0
## 10        IDOTRR   60.0
## # ... with 15 more rows
```

Processing the Data: Handling Missing Data

4) Variables with little to no relation to other variables

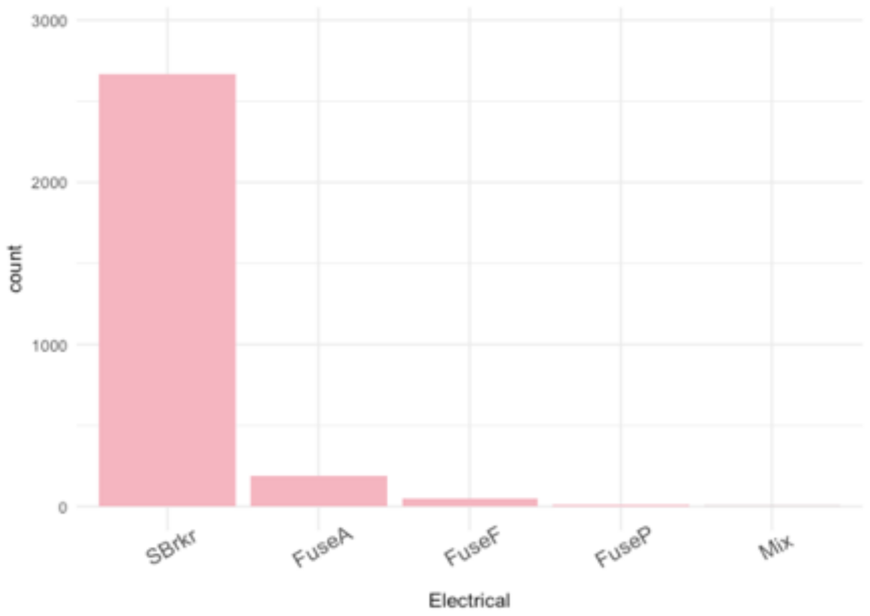
Ex. Electrical



Processing the Data: Handling Missing Data

4) Variables with little to no relation to other variables

Ex. Electrical

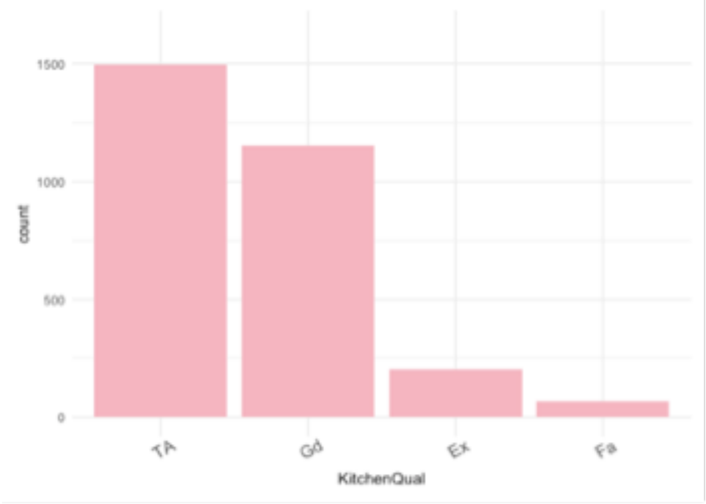


- Solution: Impute by most commonly occurring class within variable

Processing the Data: Categorical Variables (Ordinal)

Some machine learning algorithms cannot handle non-numerical values

Ex. Kitchen Quality



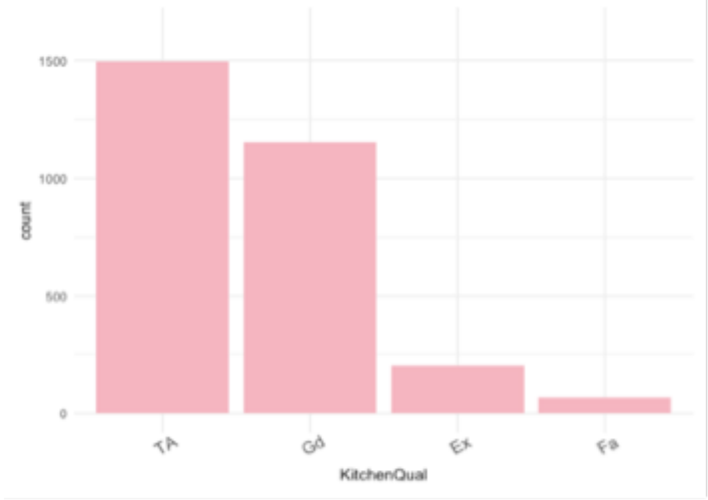
##	KitchenQual	mean.Quality	mean.Price	n
## 1	Fa	4.49	105565.2	39
## 2	TA	5.34	139962.5	735
## 3	Gd	6.79	212116.0	586
## 4	Ex	8.27	328554.7	100

('None' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)

Processing the Data: Categorical Variables (Ordinal)

Some machine learning algorithms cannot handle non-numerical values

Ex. Kitchen Quality




```
##   KitchenQual mean.Quality mean.Price   n
## 1          Fa         4.49   105565.2  39
## 2          TA         5.34   139962.5 735
## 3          Gd         6.79   212116.0 586
## 4          Ex         8.27   328554.7 100
```

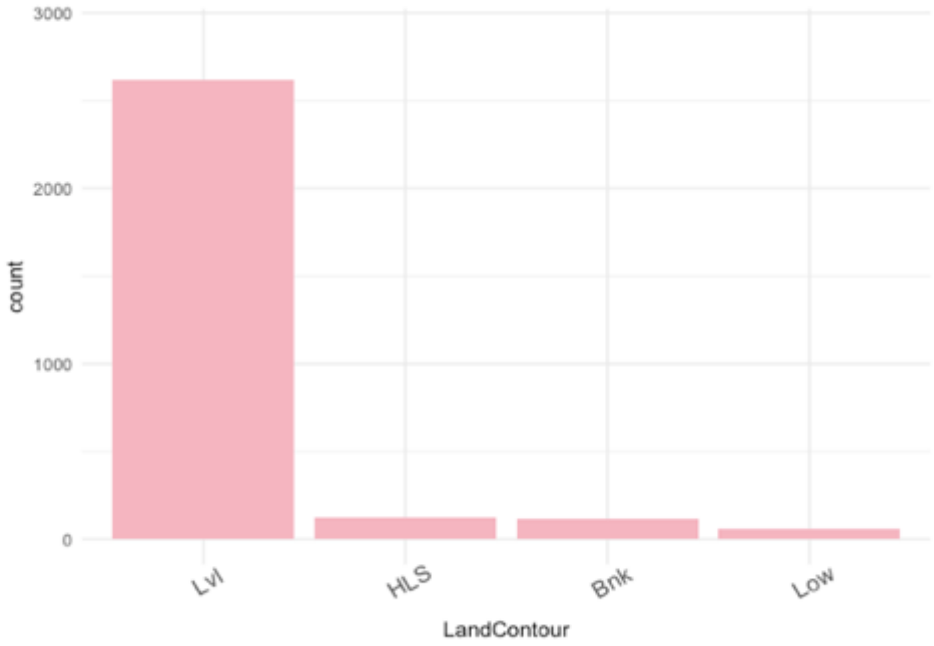
('None' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)

- Solution: Use average Sale Price to assign ordered numerical values to categories

Processing the Data: Categorical Variables (Nominal)

Some machine learning algorithms cannot handle non-numerical values

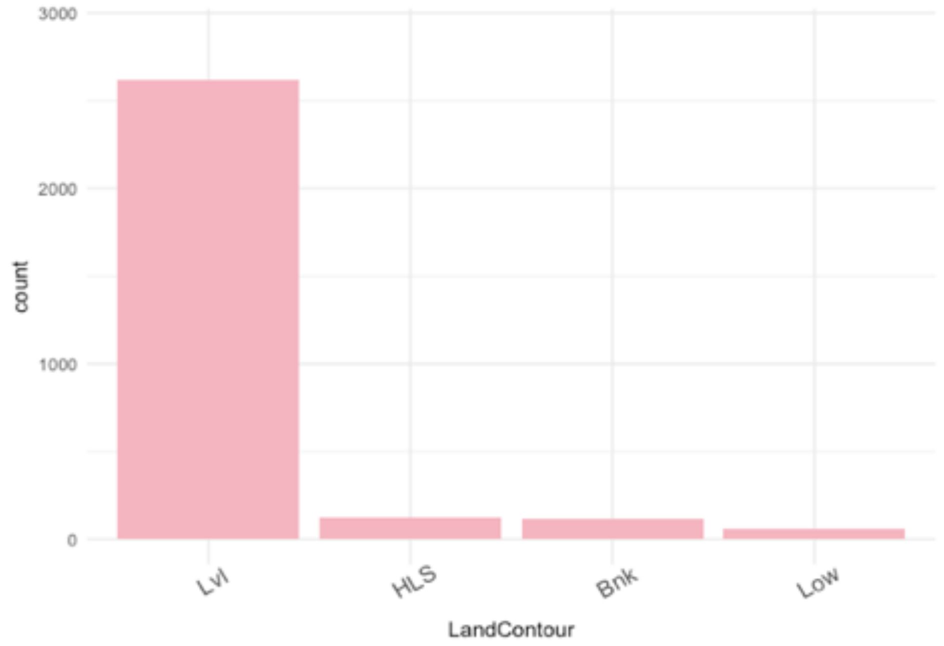
Ex. Land Contour



Processing the Data: Categorical Variables (Nominal)

Some machine learning algorithms cannot handle non-numerical values

Ex. Land Contour



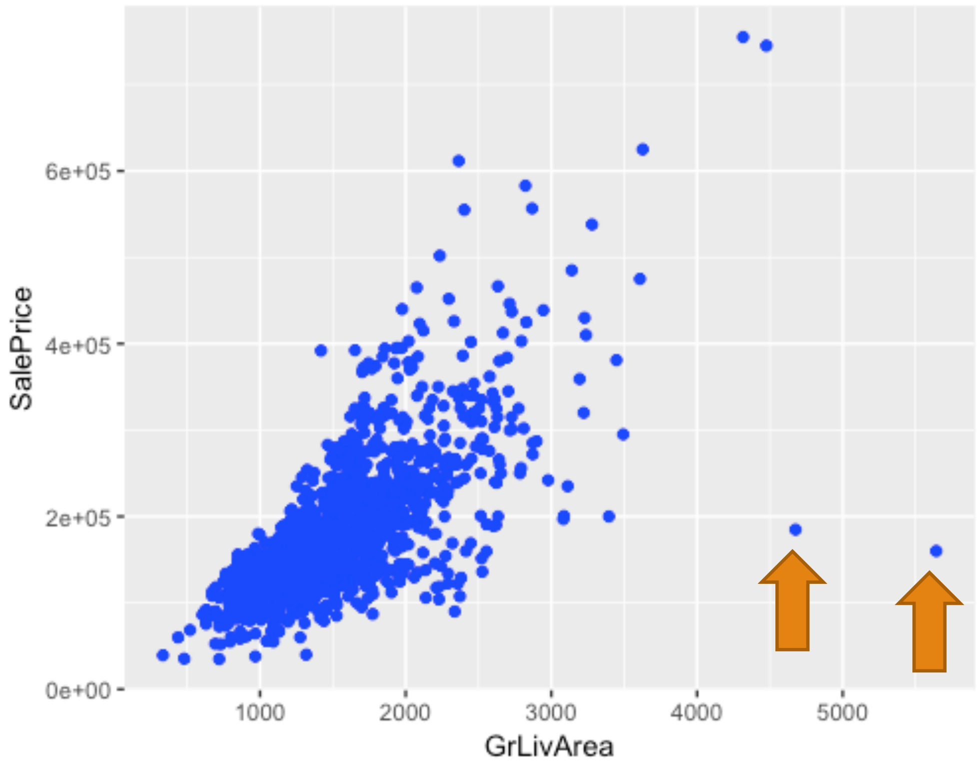
- Solution: One-hot encoding technique: binarizing classes of each variable

Processing the Data: Outliers

Some observations may be abnormally far from other values

Ex. Ground Living Area vs Sale Price

- Two points with very large area but very low sale price

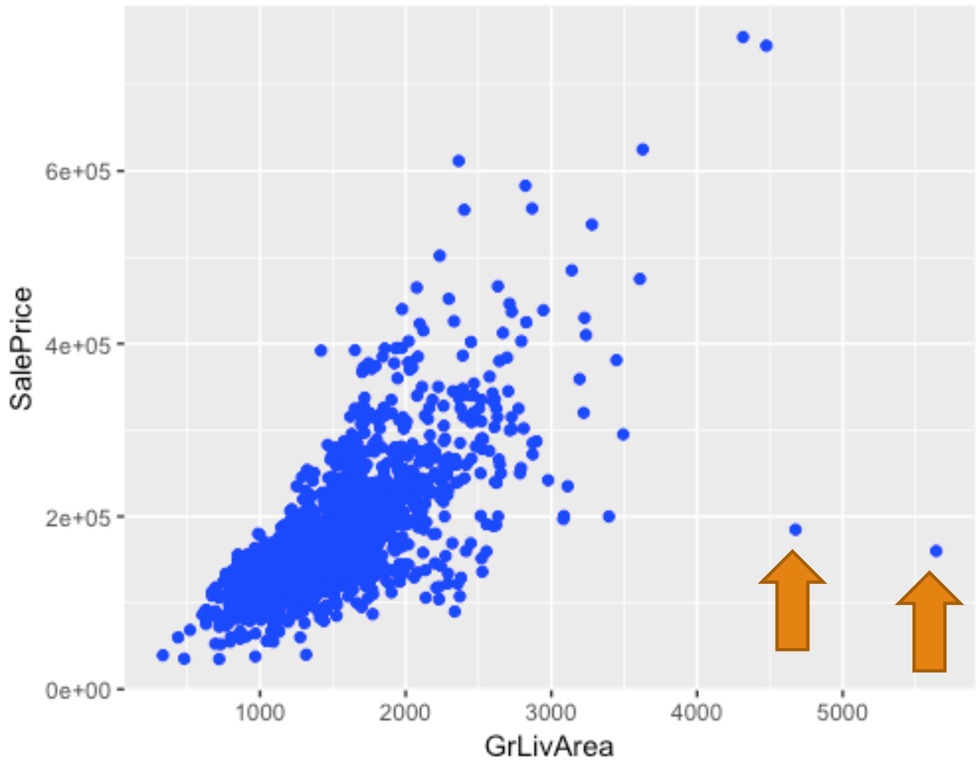


Processing the Data: Outliers

Some observations may be abnormally far from other values

Ex. Ground Living Area vs Sale Price

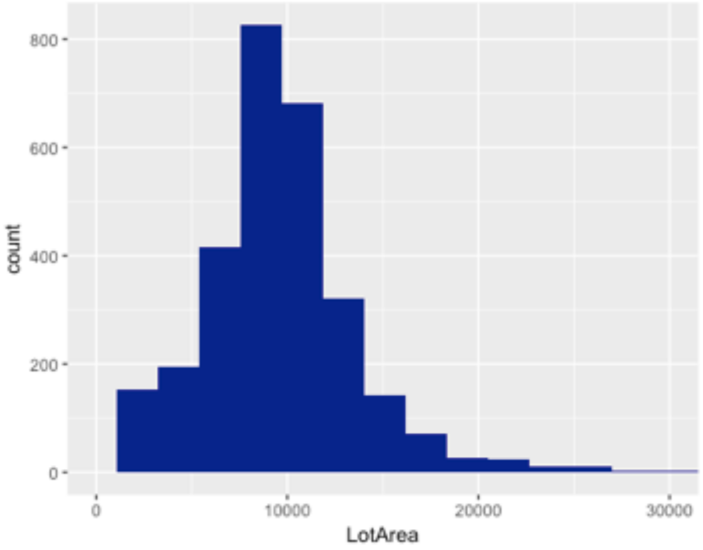
- Two points with very large area but very low sale price
- <u>Solution:</u> Remove outliers



Processing the Data: Skewness and Scaling

Distributions of some variables may be highly skewed

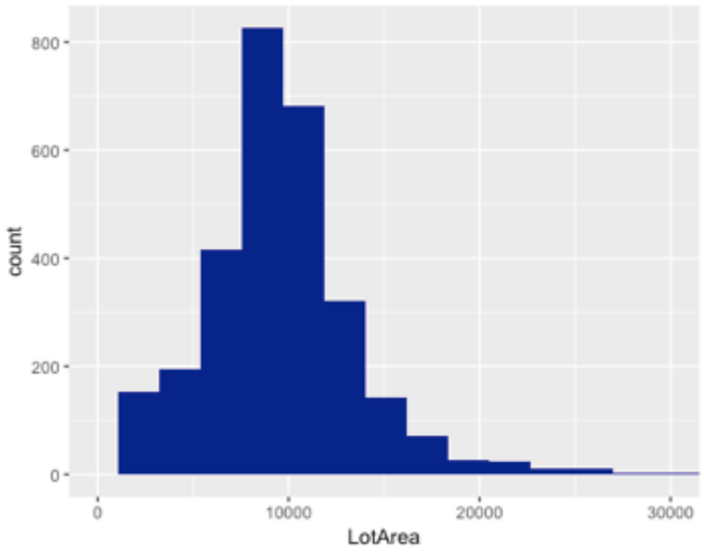
Ex. Lot Area

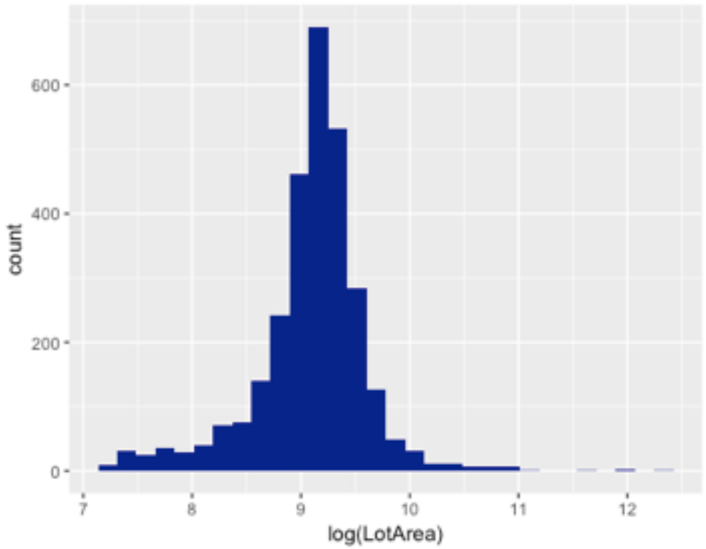


Processing the Data: Skewness and Scaling

Distributions of some variables may be highly skewed

Ex. Lot Area





- Solution: Log + 1 Transformation

Processing the Data: Near Zero Variance Predictors

Low variance predictors add little value to models

- Calculate ratio of most frequent vs. second most frequent value
- Ratios >> 1 suggest very low variance

	freqRatio	percentUnique	zeroVar	nzv
Id	1.000000	100.0000000	FALSE	FALSE
MSSubClass	1.792642	1.0273973	FALSE	FALSE
MSZoning	5.279817	0.3424658	FALSE	FALSE
LotFrontage	2.042857	7.5342466	FALSE	FALSE
LotArea	1.041667	73.4931507	FALSE	FALSE
Street	242.333333	0.1369863	FALSE	TRUE
Alley	1.219512	0.1369863	FALSE	FALSE
LotShape	1.911157	0.2739726	FALSE	FALSE
LandContour	20.809524	0.2739726	FALSE	TRUE
Utilities	1459.000000	0.1369863	FALSE	TRUE
LotConfig	4.000000	0.3424658	FALSE	FALSE
LandSlope	21.261538	0.2054795	FALSE	TRUE
Neighborhood	1.500000	1.7123288	FALSE	FALSE
Condition1	15.555556	0.6164384	FALSE	FALSE
Condition2	240.833333	0.5479452	FALSE	TRUE
BldgType	10.701754	0.3424658	FALSE	FALSE
HouseStyle	1.631461	0.5479452	FALSE	FALSE
OverallQual	1.061497	0.6849315	FALSE	FALSE
OverallCond	3.257937	0.6164384	FALSE	FALSE
YearBuilt	1.046875	7.6712329	FALSE	FALSE
YearRemodAdd	1.835052	4.1780822	FALSE	FALSE
RoofStyle	3.989510	0.4109589	FALSE	FALSE

Processing the Data: Near Zero Variance Predictors

Low variance predictors add little value to models

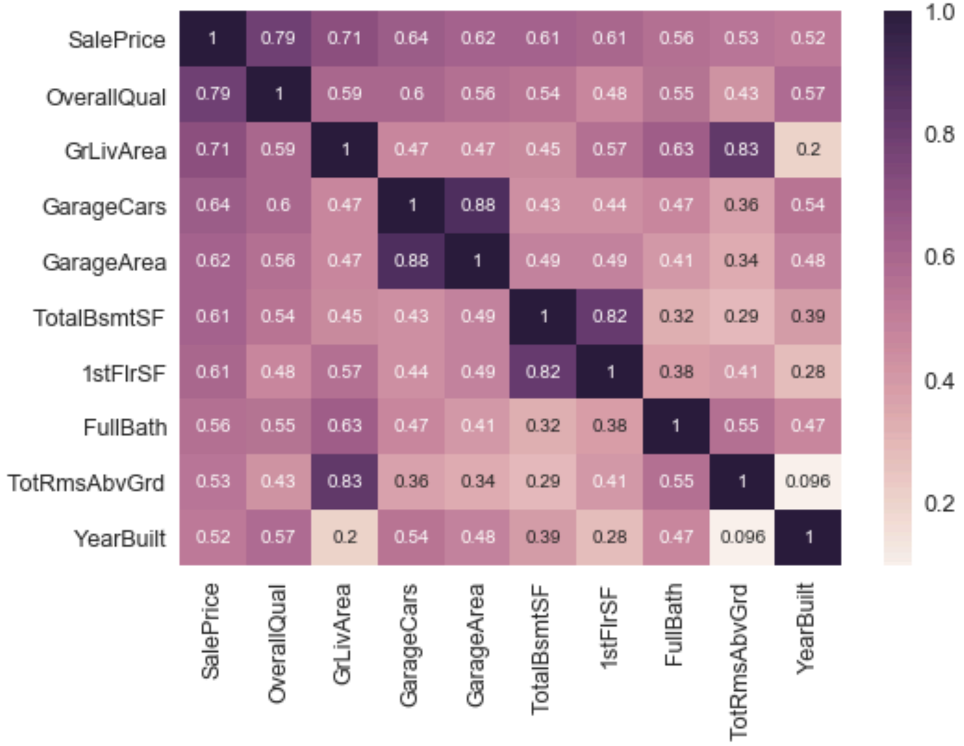
- Calculate ratio of most frequent vs. second most frequent value
- Ratios >> 1 suggest very low variance

- Solution: Remove near zero predictors with cutoffs of 95:5

	freqRatio	percentUnique	zeroVar	nzv
Id	1.000000	100.0000000	FALSE	FALSE
MSSubClass	1.792642	1.0273973	FALSE	FALSE
MSZoning	5.279817	0.3424658	FALSE	FALSE
LotFrontage	2.042857	7.5342466	FALSE	FALSE
LotArea	1.041667	73.4931507	FALSE	FALSE
Street	242.333333	0.1369863	FALSE	TRUE
Alley	1.219512	0.1369863	FALSE	FALSE
LotShape	1.911157	0.2739726	FALSE	FALSE
LandContour	20.809524	0.2739726	FALSE	TRUE
Utilities	1459.000000	0.1369863	FALSE	TRUE
LotConfig	4.000000	0.3424658	FALSE	FALSE
LandSlope	21.261538	0.2054795	FALSE	TRUE
Neighborhood	1.500000	1.7123288	FALSE	FALSE
Condition1	15.555556	0.6164384	FALSE	FALSE
Condition2	240.833333	0.5479452	FALSE	TRUE
BldgType	10.701754	0.3424658	FALSE	FALSE
HouseStyle	1.631461	0.5479452	FALSE	FALSE
OverallQual	1.061497	0.6849315	FALSE	FALSE
OverallCond	3.257937	0.6164384	FALSE	FALSE
YearBuilt	1.046875	7.6712329	FALSE	FALSE
YearRemodAdd	1.835052	4.1780822	FALSE	FALSE
RoofStyle	3.989510	0.4109589	FALSE	FALSE

Processing the Data: Numerical Variables

Top10 Numerical Variables With Greatest Covariance vs. SalePrice



As expected, important quantitative factors to consider are space/size, date, overall quality.

Feature Engineering

Ideas for new features:

- Remodeled – Year Built not equal to Year Additional Remodeling
- Seasonality – Combine Month Sold and Year Sold
- New House – Year Built same as Year Sold
- Total Area – sum all variables denoting square footage
- Inside Area – sum all variables denoting square footage referring to space inside the house

Feature Engineering

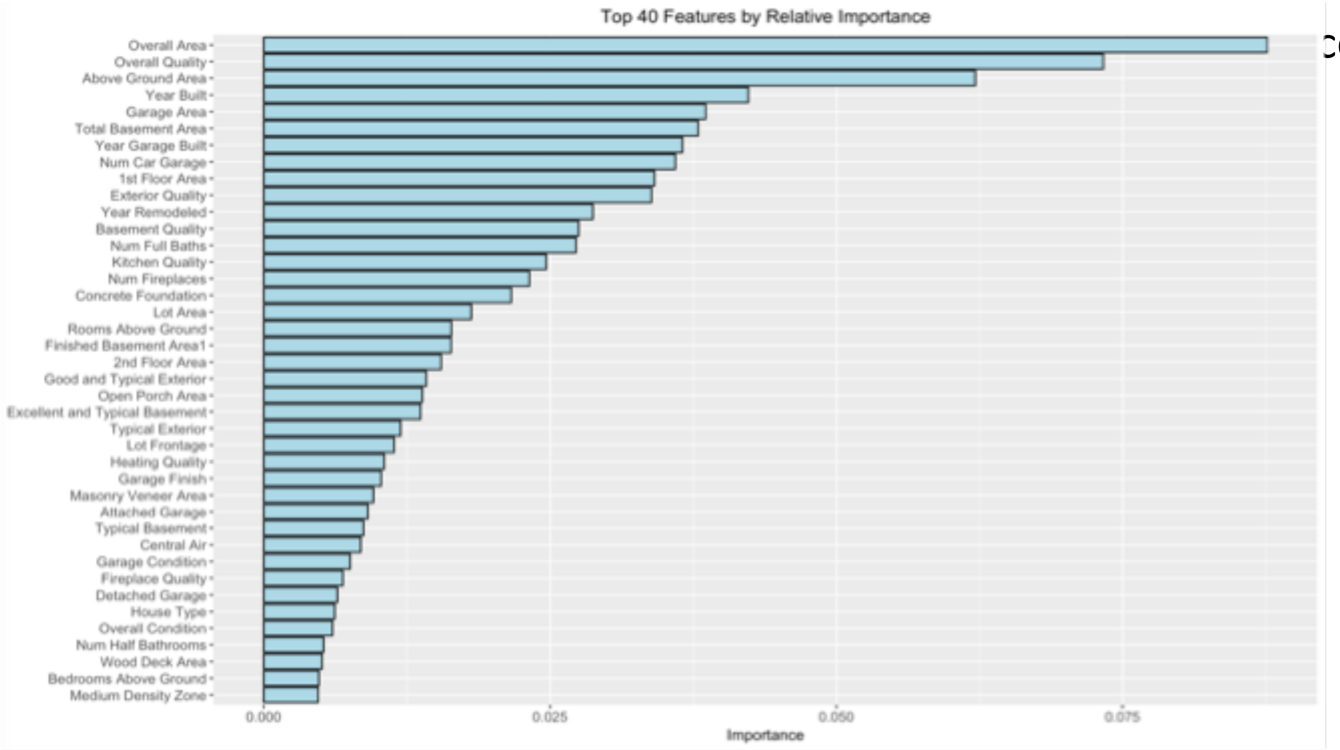
Ideas for new features:

- Remodeled – Year Built not equal to Year Additional Remodeling
- Seasonality – Combine Month Sold and Year Sold
- New House – Year Built same as Year Sold
- Total Area – sum all variables denoting square footage
- Inside Area – sum all variables denoting square footage referring to space inside the house
- Overall Basement – Basement Quality and Basement Condition
- Overall Condition – Condition 1 and Condition 2
- Overall Quality – External Quality and External Condition
- Overall Sale – Sale Type and Sale Condition
- Sale and Condtion – Sale Type and Overall Condition

Models

	Pros	Cons	Hyperparameters	Cross-Validated RSME Score	Kaggle Score
Random Forest	Lower variance, Decorrelates data, Scale invariant	High bias, Overfitting, Difficult to interpret	Num features = 48, Num trees = 10000	0.14997	0.14758

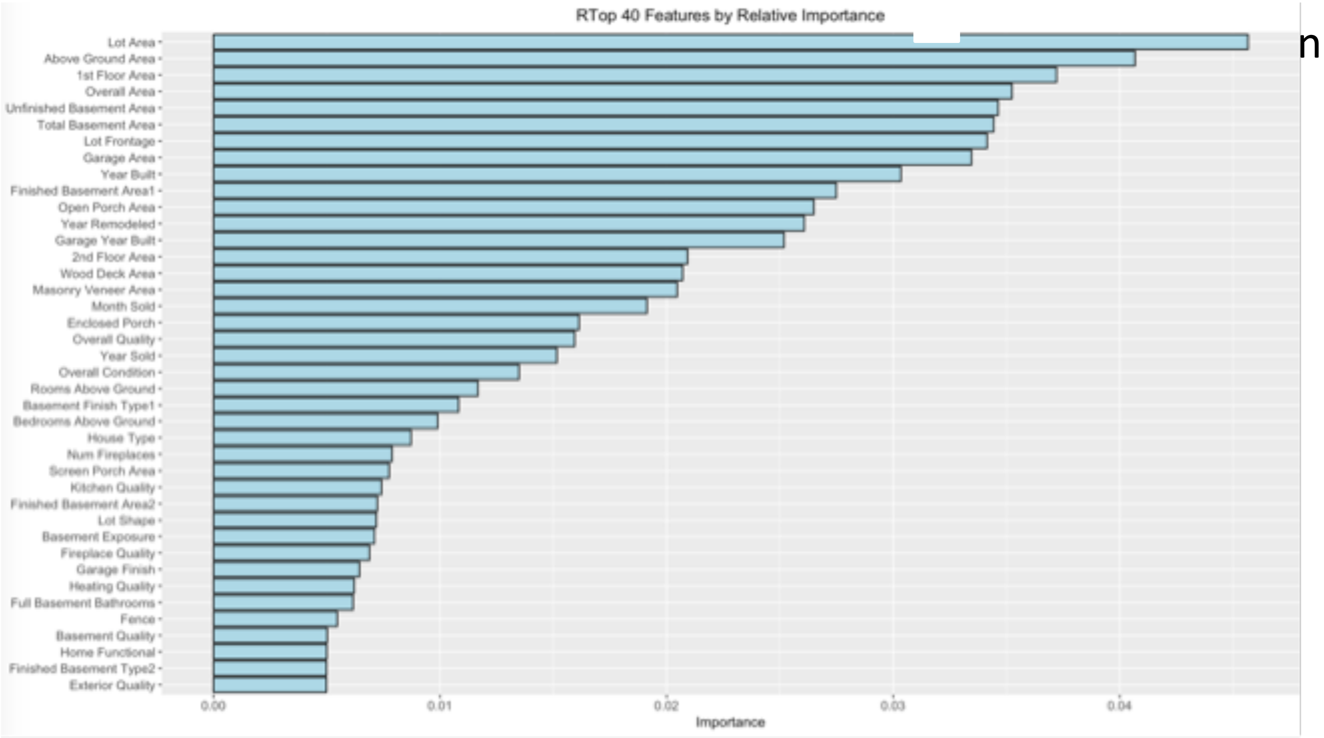
ce Random Forest



Models

	Pros	Cons	Hyperparameters	Cross-Validated RSME Score	Kaggle Score
Random Forest	Lower variance, Decorrelates data, Scale invariant	High bias, Overfitting, Difficult to interpret	Num features = 48, Num trees = 10000	0.14997	0.14758
Gradient Boost	**Feature scaling not needed, High accuracy**	**Computationally expensive, Overfitting**	**Num trees = 1000, Depth = 2, Num Features = sqrt, Samples/leaf = 15, Learning rate = 0.05**	**0.1128**	**0.12421**

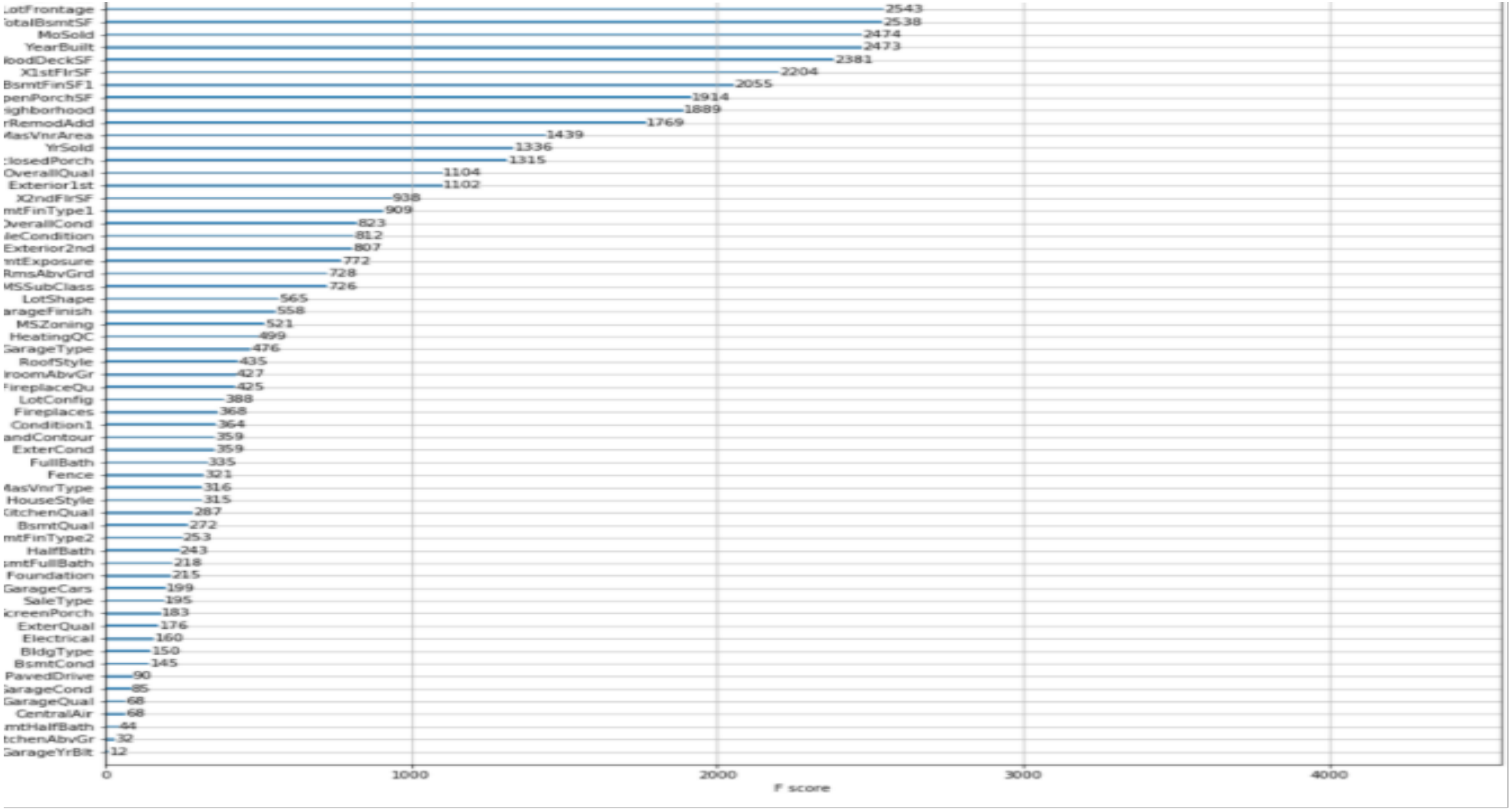
nce Gradient Boost



Models

	Pros	Cons	Hyperparameters	Cross-Validated RSME Score	Kaggle Score
Random Forest	Lower variance, Decorrelates data, Scale invariant	High bias, Overfitting, Difficult to interpret	Num features = 48, Num trees = 10000	0.14997	0.14758
Gradient Boost	Feature scaling not needed, High accuracy	Computationally expensive, Overfitting	Num trees = 1000, Depth = 2, Num Features = sqrt, Samples/leaf = 15, Learning rate = 0.05	0.1128	0.12421
XGBoost	**Extremely fast, Allows parallel computing**	**Very fast, Difficult to interpret**	**Num trees = 2724, Max depth = 30, Gamma = 0.0, Minimum child weight = 4**	**0.13642**	**0.13082**

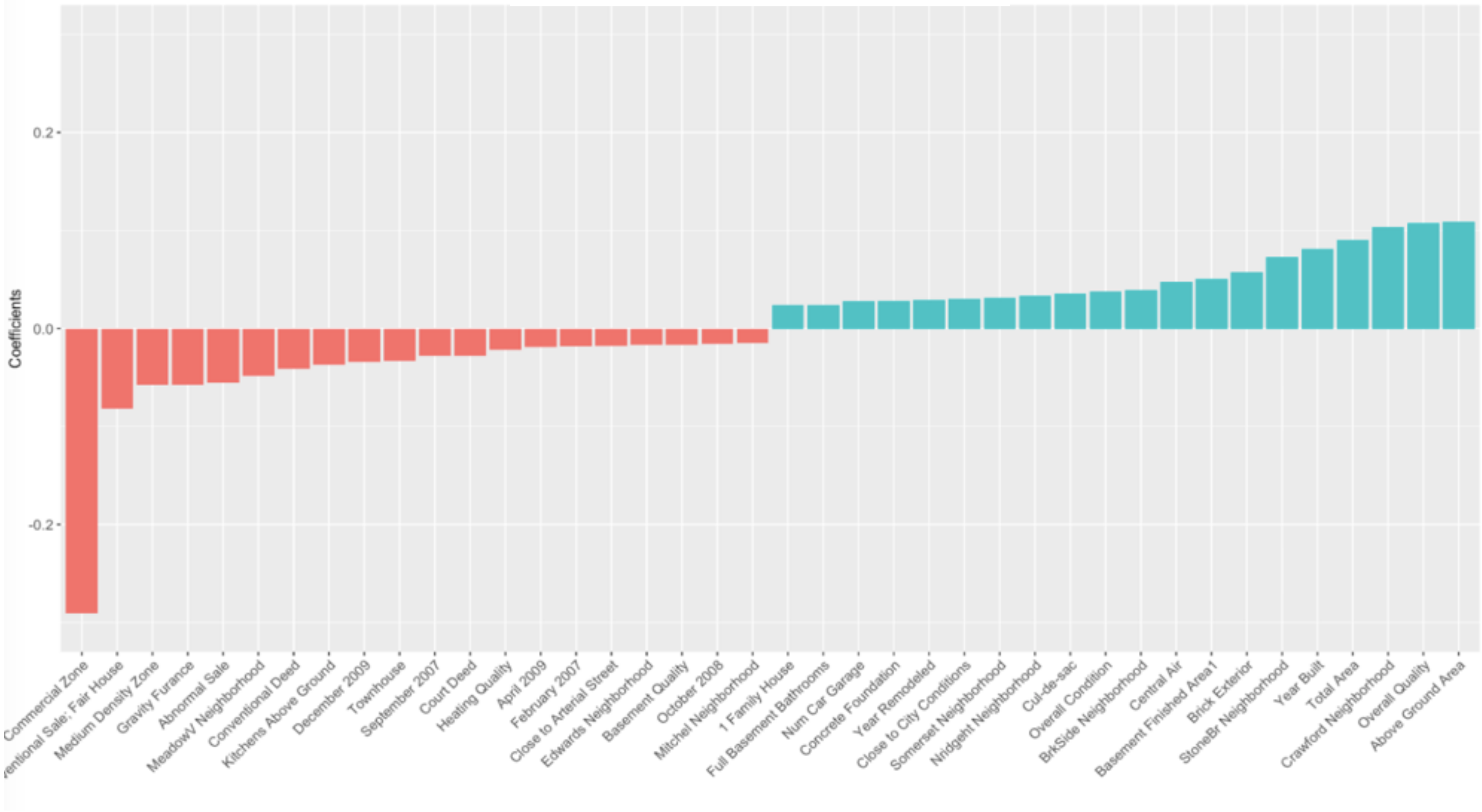
Top 40 Features by Relative Importance XGBoost



Models

	Pros	Cons	Hyperparameters	Cross-Validated RSME Score	Kaggle Score
Random Forest	Lower variance, Decorrelates data, Scale invariant	High bias, Overfitting, Difficult to interpret	Num features = 48, Num trees = 10000	0.14997	0.14758
Gradient Boost	Feature scaling not needed, High accuracy	Computationally expensive, Overfitting	Num trees = 1000, Depth = 2, Num Features = sqrt, Samples/leaf = 15, Learning rate = 0.05	0.1128	0.12421
XGBoost	Extremely fast, Allows parallel computing	Very fast, Difficult to interpret	Num trees = 2724, Max depth = 30, Gamma = 0.0, Minimum child weight = 4	0.13642	0.13082
Regularize Linear Regression	**Easily interpretable, Computationally inexpensive, Less prone to overfitting**	**Requires scaled variables, Requires numerical variables**	**Lambda = 0.0005, Alpha = 0.9**	**0.1111**	**0.11922**

Coefficients of Top 40 Predictors



Models

	Pros	Cons	Hyperparameters	Cross-Validated RSME Score	Kaggle Score
Random Forest	Lower variance, Decorrelates data, Scale invariant	High bias, Overfitting, Difficult to interpret	Num features = 48, Num trees = 10000	0.14997	0.14758
Gradient Boost	Feature scaling not needed, High accuracy	Computationally expensive, Overfitting	Num trees = 1000, Depth = 2, Num Features = sqrt, Samples/leaf = 15, Learning rate = 0.05	0.1128	0.12421
XGBoost	Extremely fast, Allows parallel computing	Very fast, Difficult to interpret	Num trees = 2724, Max depth = 30, Gamma = 0.0, Minimum child weight = 4	0.13642	0.13082
Regularize Linear Regression	Easily interpretable, Computationally inexpensive, Less prone to overfitting	Requires scaled variables, Requires numerical variables	Lambda = 0.0005, Alpha = 0.9	0.1111	0.11922
Ensembling	**Can improve accuracy**	**Lose interpretability**	**Lasso, Enet, Gradient Boost, Gradient Boost Lite**	**0.1071**	**0.11751**

Conclusions

Conclusions

Lowest RMSE yields an error of $\pm$ \$8500

- (RMSE * SD of Sale Price)

Median of \$163000, error $\approx$ 5%

- $\approx$ 24% lowest priced house
- $\approx$ 1% highest priced house

Conclusions

Lowest RMSE yields an error of $\pm$ $8500

- (RMSE * SD of Sale Price)

Median of $163000, error ≈ 5%

- ≈ 24% lowest priced house
- ≈ 1% highest priced house

<u>What drives sale price?</u>

Size, Age

Overall Quality/Condition

Neighborhood (both good and bad)

Commercial Zone

Year sold (housing crash)

Questions?