House Price Predication

—— Zining Wang, Wenxuan Wang and Wenda Zheng

Motivation and objective

Everyone has a dream house in mind!

- Tradeoffs between certain criteria and house prices
- Avoid over-estimated house
- How to predict the actual value of the house
- Convenient without field investigation

Goal:

Predicting House price according to different conditions.

Data Description

- 79 house attributes

- include structural attributes (e.g. the area of first floor), location attributes (e.g. whether the house is adjacent to railroads), and some overall quality ratings.

- Training data & Testing data

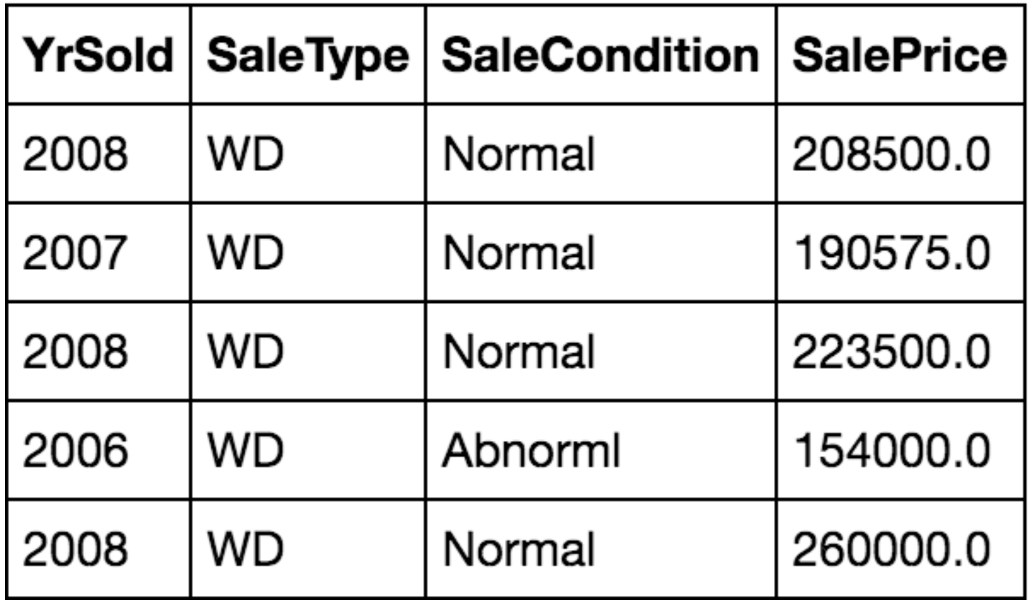
Data preprocessing

- Discard attributes that have lots of missing values
- Adjust house price based on CPI index, choose Dec.2010 as base year

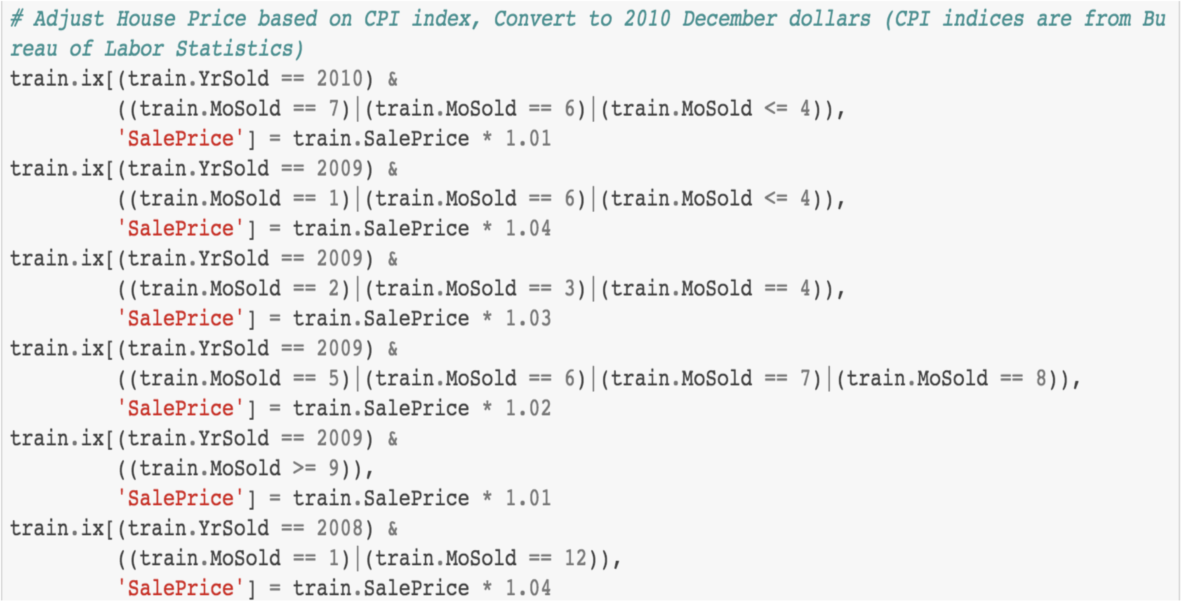
- Consumer price index (CPI)
  - Measure of the overall level of prices
  - Measure of the overall cost of goods and services
  - Bought by a typical consumer

$$\text{Inflation rate in year 2} = \frac{\text{CPI in year 2 - CPI in year 1}}{\text{CPI in year 1}} \times 100$$

Adjusting price based on CPI Index



YrSold	SaleType	SaleCondition	SalePrice
2008	WD	Normal	208500.0
2007	WD	Normal	190575.0
2008	WD	Normal	223500.0
2006	WD	Abnorml	154000.0
2008	WD	Normal	260000.0

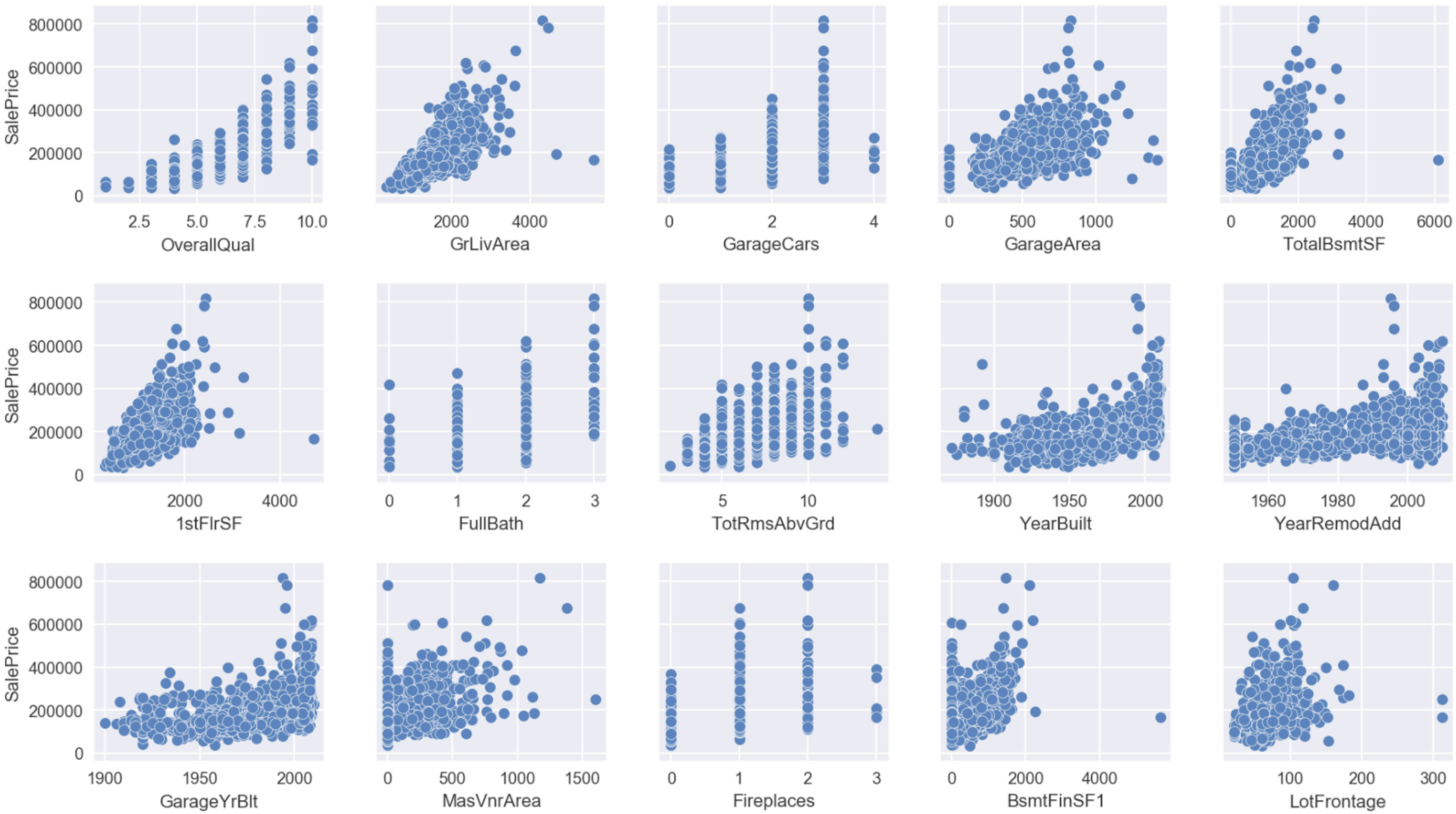


```python
# Adjust House Price based on CPI index, Convert to 2010 December dollars (CPI indices are from Bureau of Labor Statistics)
train.ix[(train.YrSold == 2010) &
         ((train.MoSold == 7)|(train.MoSold == 6)|(train.MoSold <= 4)),
         'SalePrice'] = train.SalePrice * 1.01
train.ix[(train.YrSold == 2009) &
         ((train.MoSold == 1)|(train.MoSold == 6)|(train.MoSold <= 4)),
         'SalePrice'] = train.SalePrice * 1.04
train.ix[(train.YrSold == 2009) &
         ((train.MoSold == 2)|(train.MoSold == 3)|(train.MoSold == 4)),
         'SalePrice'] = train.SalePrice * 1.03
train.ix[(train.YrSold == 2009) &
         ((train.MoSold == 5)|(train.MoSold == 6)|(train.MoSold == 7)|(train.MoSold == 8)),
         'SalePrice'] = train.SalePrice * 1.02
train.ix[(train.YrSold == 2009) &
         ((train.MoSold >= 9)),
         'SalePrice'] = train.SalePrice * 1.01
train.ix[(train.YrSold == 2008) &
         ((train.MoSold == 1)|(train.MoSold == 12)),
         'SalePrice'] = train.SalePrice * 1.04
```

Data preprocessing (Other Techniques)

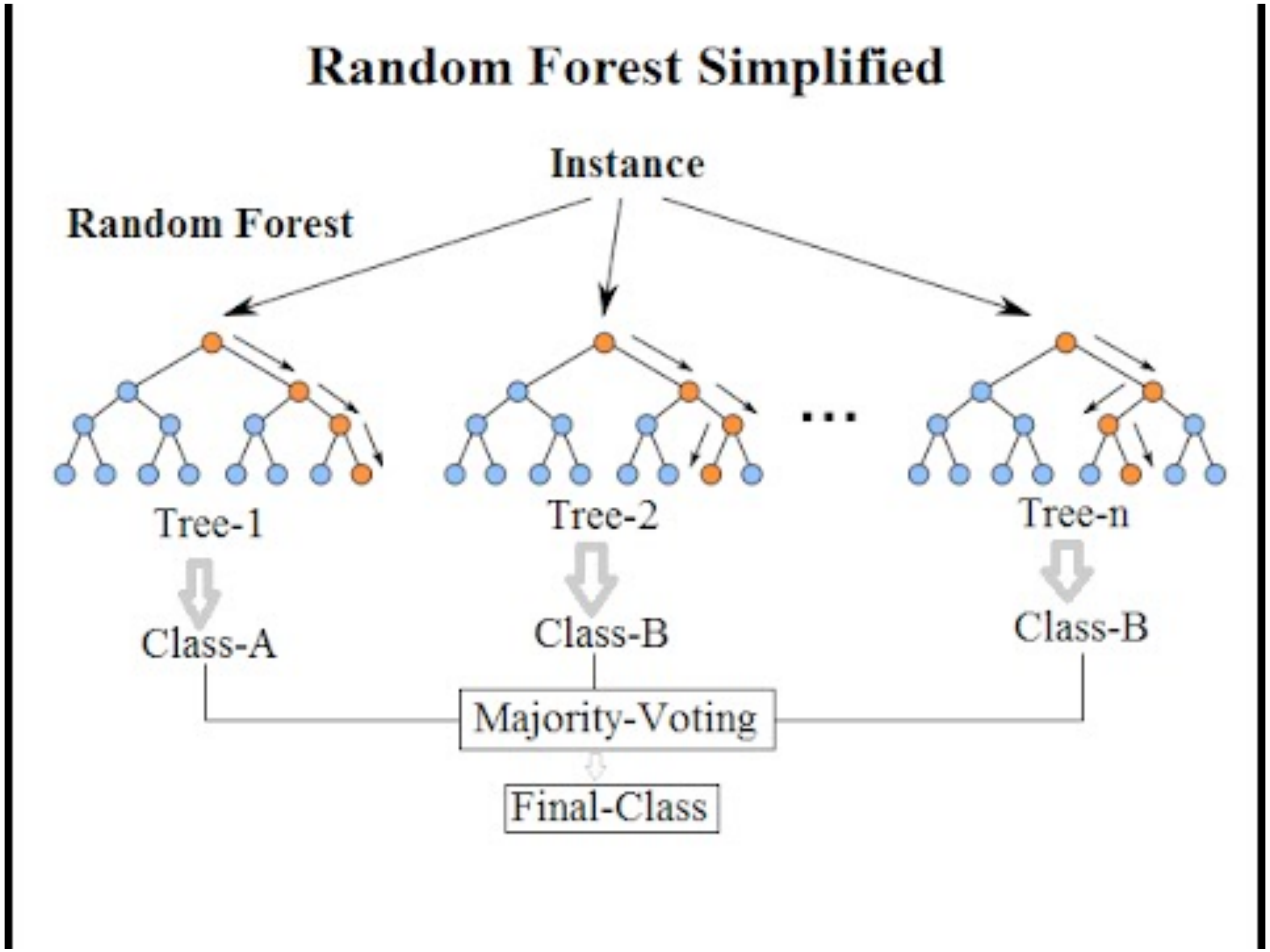
- int to string
- NA to another kind
- NA to 0.0
- Mean/Mode imputation
- Drop
- ...

Pairwise Correlation



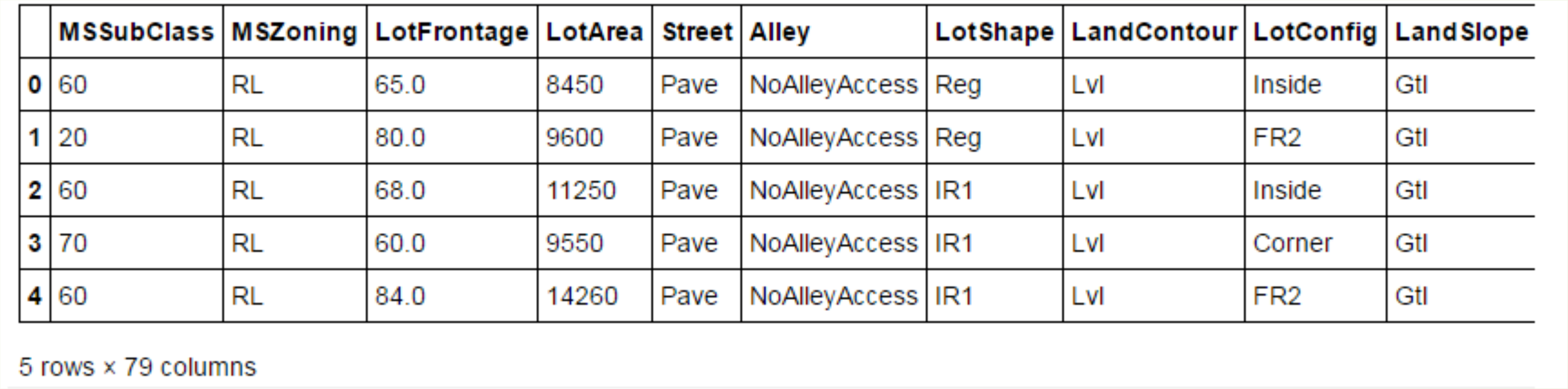
XGBoost (Extreme Gradient Boosting)

- based on the boosting tree model
- Uses gradient descent and boosting method to overcome incorrectly classified subsets over each iteration, until some stopping criterion is met
- Existing Python packages for XGBoost


Random Forest Simplified
Instance
Random Forest
...
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B
Majority-Voting
Final-Class

Result of every single step

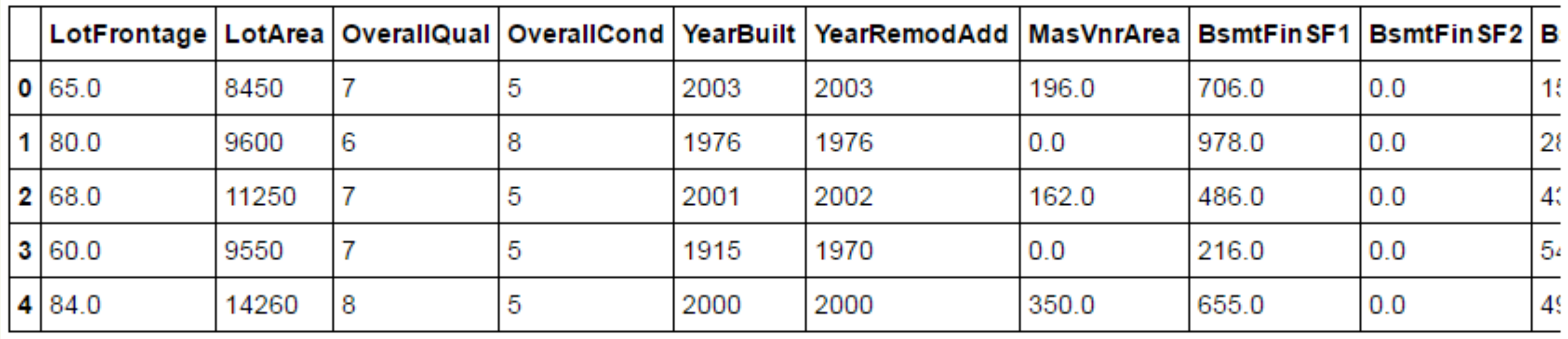
First 5 rows of input after cleaning



	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	LotConfig	LandSlope
0	60	RL	65.0	8450	Pave	NoAlleyAccess	Reg	Lvl	Inside	Gtl
1	20	RL	80.0	9600	Pave	NoAlleyAccess	Reg	Lvl	FR2	Gtl
2	60	RL	68.0	11250	Pave	NoAlleyAccess	IR1	Lvl	Inside	Gtl
3	70	RL	60.0	9550	Pave	NoAlleyAccess	IR1	Lvl	Corner	Gtl
4	60	RL	84.0	14260	Pave	NoAlleyAccess	IR1	Lvl	FR2	Gtl

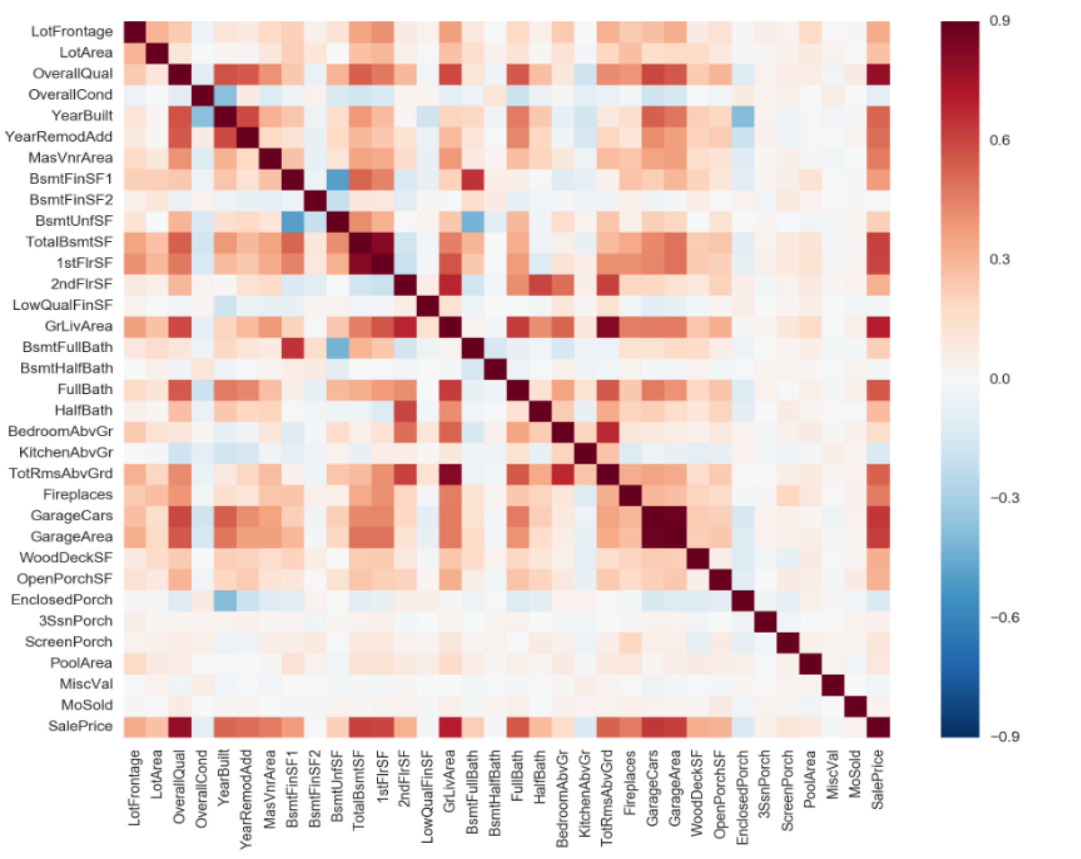
5 rows × 79 columns

Replacing Dummy Variables

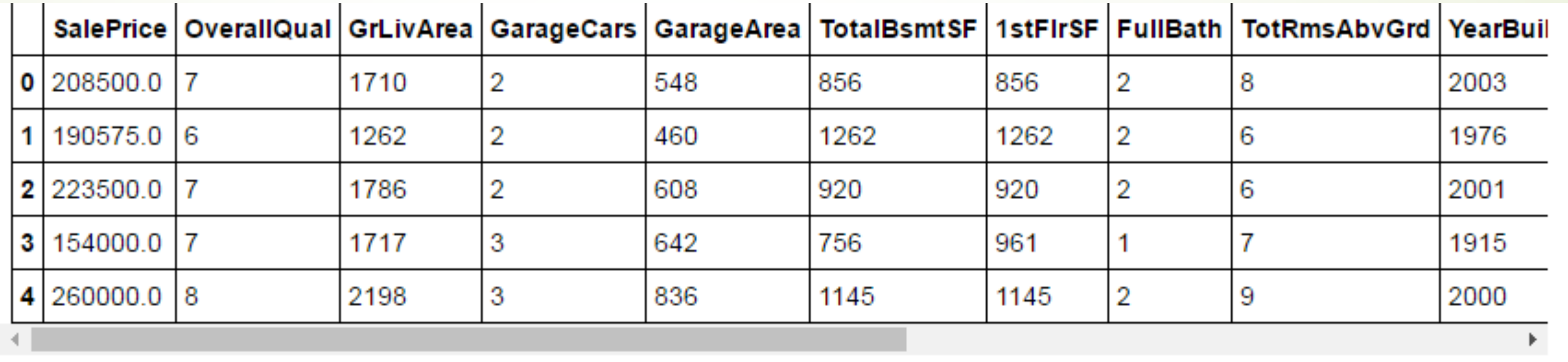


	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	B
0	65.0	8450	7	5	2003	2003	196.0	706.0	0.0	15
1	80.0	9600	6	8	1976	1976	0.0	978.0	0.0	28
2	68.0	11250	7	5	2001	2002	162.0	486.0	0.0	43
3	60.0	9550	7	5	1915	1970	0.0	216.0	0.0	54
4	84.0	14260	8	5	2000	2000	350.0	655.0	0.0	49

Correlation Heatmap



Fifteen most important attributes



	SalePrice	OverallQual	GrLivArea	GarageCars	GarageArea	TotalBsmtSF	1stFlrSF	FullBath	TotRmsAbvGrd	YearBuil
0	208500.0	7	1710	2	548	856	856	2	8	2003
1	190575.0	6	1262	2	460	1262	1262	2	6	1976
2	223500.0	7	1786	2	608	920	920	2	6	2001
3	154000.0	7	1717	3	642	756	961	1	7	1915
4	260000.0	8	2198	3	836	1145	1145	2	9	2000

Related work and method

- Regularized linear regression

- XGBoost (Extreme Gradient Boosting)

Regularized Linear Regression

- In our dataset, 79 house attributes are used as independent variables, and house sales price is the dependent variable.

- Served as the baseline model for our project

- Adding a regularized term to control the complexity of the regression model. E.g. Ridge or Lasso regularization

Future work

Thank you!!