# Data Mining Project

Xudong ZHANG
24/11/2017

## Understanding the problem:

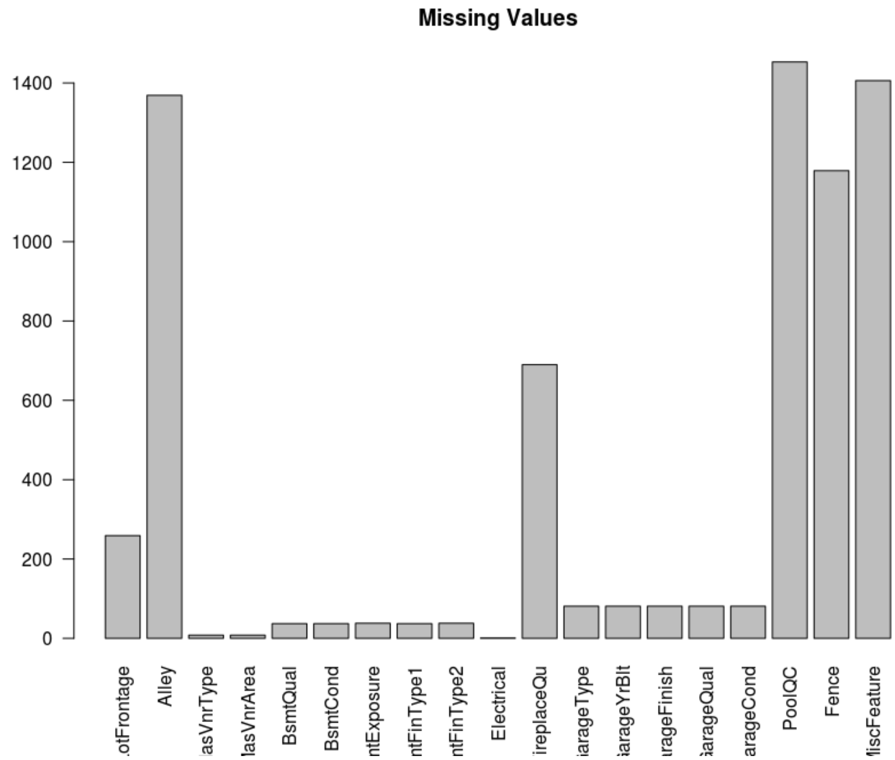
The goal of this problem to expect the prices for the houses. With missing data and a big number of features. The problem from “Kaggle” called “House Prices: Advanced Regression Techniques”.
81 features and 1460 sample with many missing values. So during this project I am trying to deal with the big number of features and find the best model to predict the house prices correctly.

The source code in github with the dataset:
https://github.com/a511260195/house-price-predicting

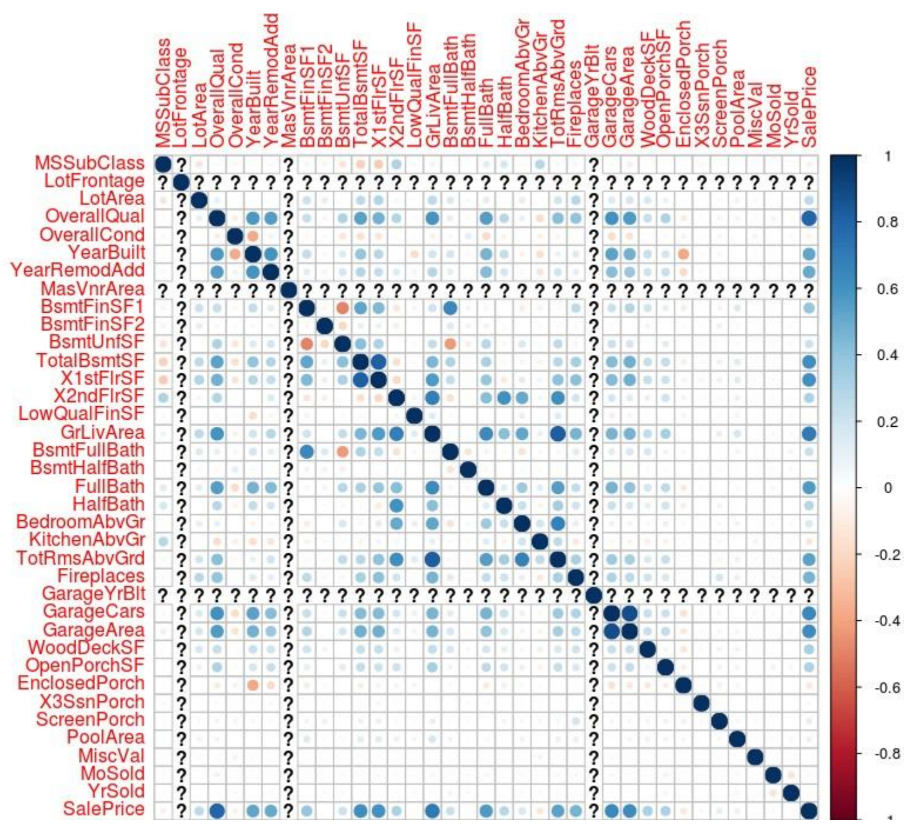
**SalePrice** - the property's sale price in dollars. This is the target variable that you're trying to predict.
**MSSubClass**: The building class
**MSZoning**: The general zoning classification
**LotFrontage**: Linear feet of street connected to property
**LotArea**: Lot size in square feet
**Street**: Type of road access
**Alley**: Type of alley access
**LotShape**: General shape of property
**LandContour**: Flatness of the property
**Utilities**: Type of utilities available
**LotConfig**: Lot configuration
**LandSlope**: Slope of property
**Neighborhood**: Physical locations within Ames city limits
**Condition1**: Proximity to main road or railroad
**Condition2**: Proximity to main road or railroad (if a second is present)
**BldgType**: Type of dwelling
**HouseStyle**: Style of dwelling
**OverallQual**: Overall material and finish quality
**OverallCond**: Overall condition rating
**YearBuilt**: Original construction date
**YearRemodAdd**: Remodel date
**RoofStyle**: Type of roof
**RoofMatl**: Roof material
**Exterior1st**: Exterior covering on house
**Exterior2nd**: Exterior covering on house (if more than one material)
**MasVnrType**: Masonry veneer type
**MasVnrArea**: Masonry veneer area in square feet
**ExterQual**: Exterior material quality
**ExterCond**: Present condition of the material on the exterior
**Foundation**: Type of foundation
**BsmtQual**: Height of the basement
**BsmtCond**: General condition of the basement
**BsmtExposure**: Walkout or garden level basement walls
**BsmtFinType1**: Quality of basement finished area

**BsmtFinSF1**: Type 1 finished square feet
**BsmtFinType2**: Quality of second finished area (if present)
**BsmtFinSF2**: Type 2 finished square feet
**BsmtUnfSF**: Unfinished square feet of basement area
**TotalBsmtSF**: Total square feet of basement area
**Heating**: Type of heating
**HeatingQC**: Heating quality and condition
**CentralAir**: Central air conditioning
**Electrical**: Electrical system
**1stFlrSF**: First Floor square feet
**2ndFlrSF**: Second floor square feet
**LowQualFinSF**: Low quality finished square feet (all floors)
**GrLivArea**: Above grade (ground) living area square feet
**BsmtFullBath**: Basement full bathrooms
**BsmtHalfBath**: Basement half bathrooms
**FullBath**: Full bathrooms above grade
**HalfBath**: Half baths above grade
**Bedroom**: Number of bedrooms above basement level
**Kitchen**: Number of kitchens
**KitchenQual**: Kitchen quality
**TotRmsAbvGrd**: Total rooms above grade (does not include bathrooms)
**Functional**: Home functionality rating
**Fireplaces**: Number of fireplaces
**FireplaceQu**: Fireplace quality
**GarageType**: Garage location
**GarageYrBlt**: Year garage was built
**GarageFinish**: Interior finish of the garage
**GarageCars**: Size of garage in car capacity
**GarageArea**: Size of garage in square feet
**GarageQual**: Garage quality
**GarageCond**: Garage condition
**PavedDrive**: Paved driveway
**WoodDeckSF**: Wood deck area in square feet
**OpenPorchSF**: Open porch area in square feet
**EnclosedPorch**: Enclosed porch area in square feet
**3SsnPorch**: Three season porch area in square feet
**ScreenPorch**: Screen porch area in square feet
**PoolArea**: Pool area in square feet
**PoolQC**: Pool quality
**Fence**: Fence quality
**MiscFeature**: Miscellaneous feature not covered in other categories
**MiscVal**: $Value of miscellaneous feature
**MoSold**: Month Sold
**YrSold**: Year Sold
**SaleType**: Type of sale
**SaleCondition**: Condition of sale

The Features with Missing Values:




The correlation between the Features:



We can notice a big similar between the Garage Area and the number of cars in the garage
and between Overall material and finish quality and the price.

## Data Prepration:

So to deal with the missing value.
We can notice for example about Garage features the features is null in the same time. when the GarageCars=0 and GarageArea=0. so that's meaning there are no information to the garage for this house.

In this way I will give “None” instead of null for this value in the data.

| | GarageType | GarageYrBlt | GarageFinish | GarageCars | GarageArea | GarageQual | GarageCond |
|---|---|---|---|---|---|---|---|
| 1338 | <NA> | NA | <NA> | 0 | 0 | <NA> | <NA> |
| 1339 | BuiltIn | 2002 | RFn | 2 | 492 | TA | TA |
| 1340 | Attchd | 1972 | RFn | 1 | 288 | TA | TA |
| 1341 | Detchd | 1974 | Unf | 4 | 480 | TA | TA |
| 1342 | Detchd | 2004 | Unf | 2 | 576 | TA | TA |
| 1343 | Attchd | 2002 | RFn | 2 | 647 | TA | TA |
| 1344 | Detchd | 1929 | Unf | 2 | 342 | Fa | Fa |
| 1345 | Attchd | 2006 | Fin | 2 | 440 | TA | TA |
| 1346 | Detchd | 1997 | Unf | 1 | 308 | TA | TA |
| 1347 | Attchd | 1968 | RFn | 2 | 508 | Gd | TA |
| 1348 | Attchd | 2006 | Fin | 3 | 712 | TA | TA |
| 1349 | Attchd | 1998 | RFn | 2 | 514 | TA | TA |
| 1350 | <NA> | NA | <NA> | 0 | 0 | <NA> | <NA> |

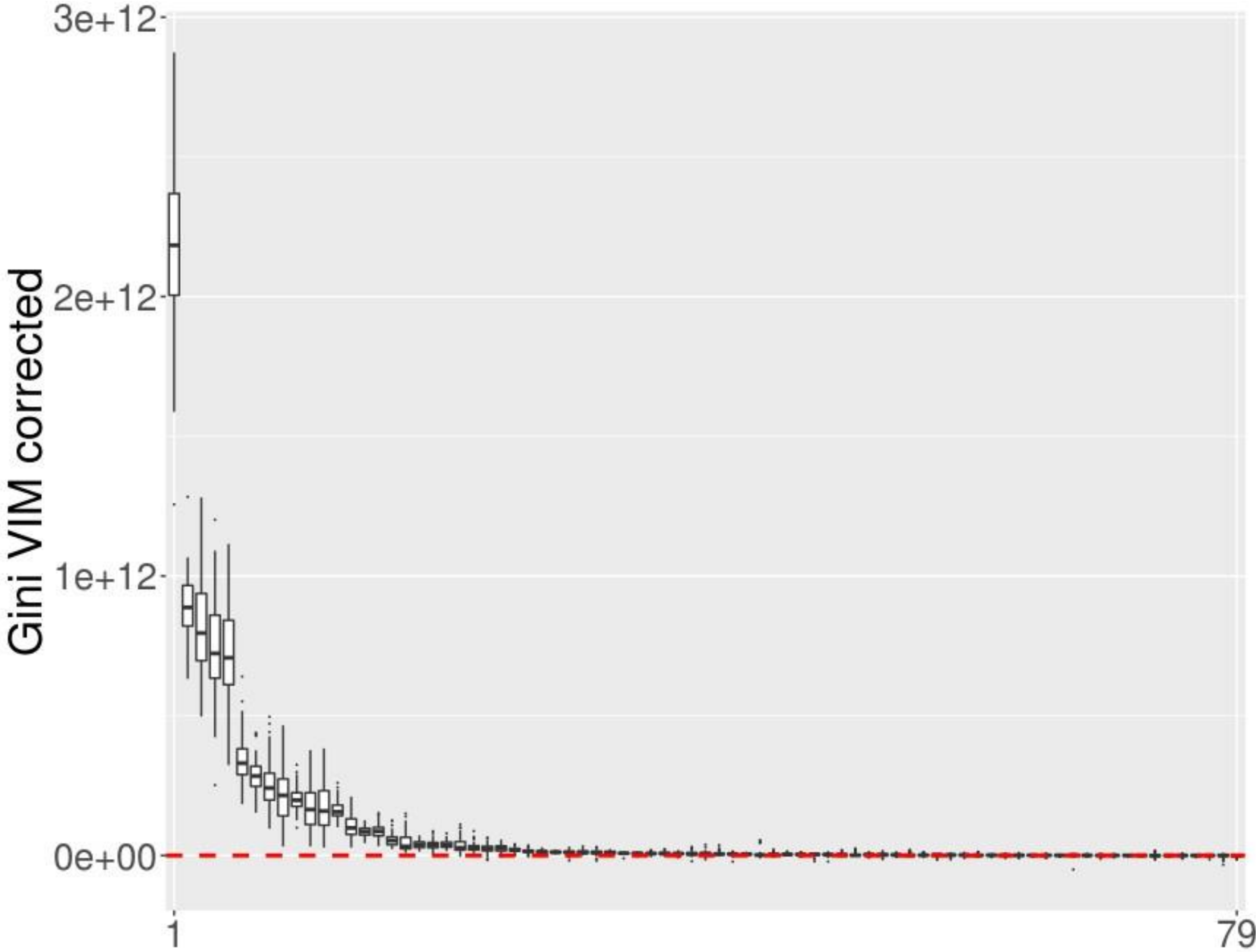
and for example for “LotFrontage” it is integer so I will replace the null value with the median.

And that processing for all the features.

## Feature selection:

I used Decision tree in parallel and choose the best 15 feature. After that I saved the features because the code take long time. So the feature after the selection.

```
names_f <- c("OverallQual", "GrLivArea", "Neighborhood", "GarageCars", "ExterQual",
           "TotalBsmtSF", "X1stFlrSF", "GarageArea", "KitchenQual", "X2ndFlrSF",
           "BsmtQual", "YearBuilt", "BsmtFinSF1")
```

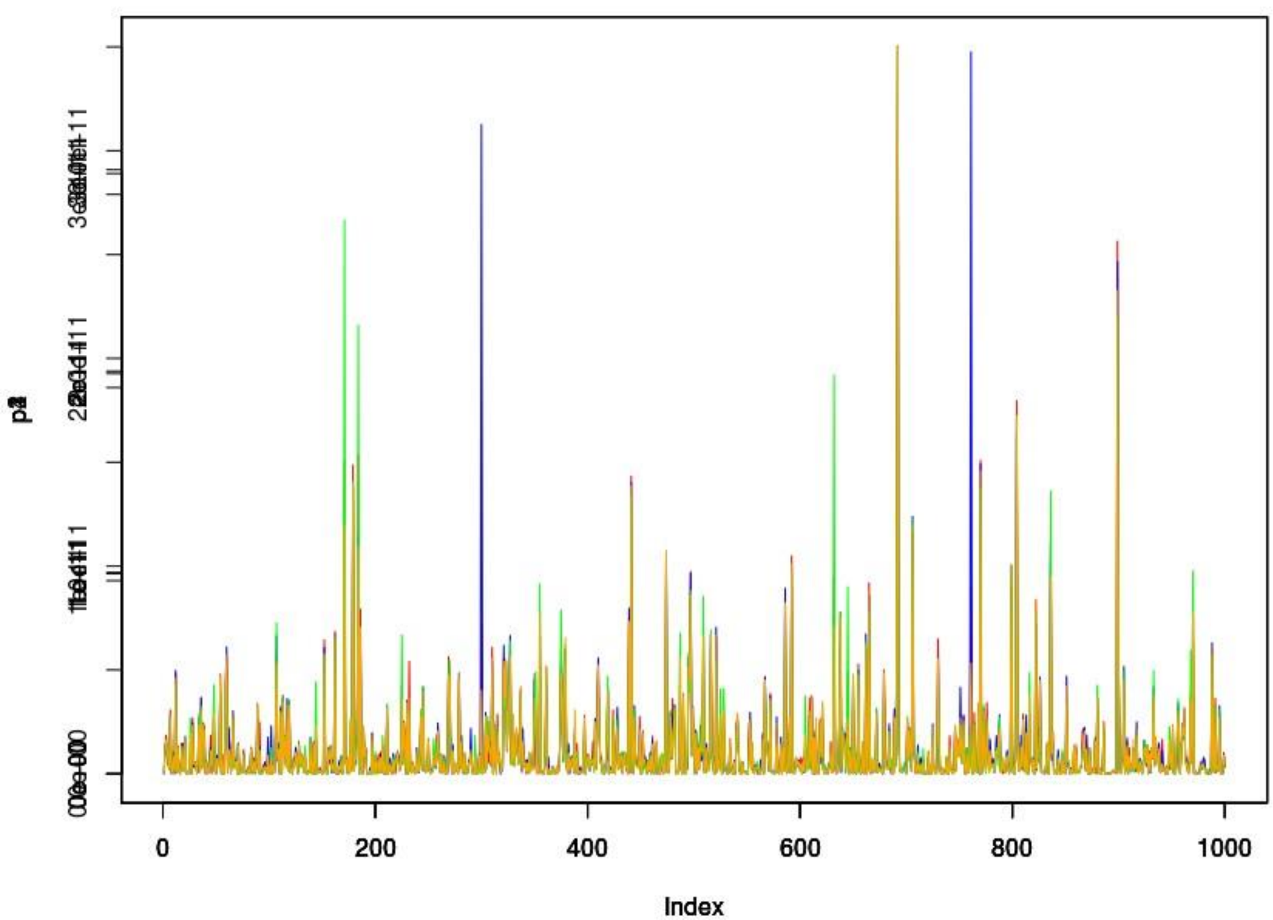


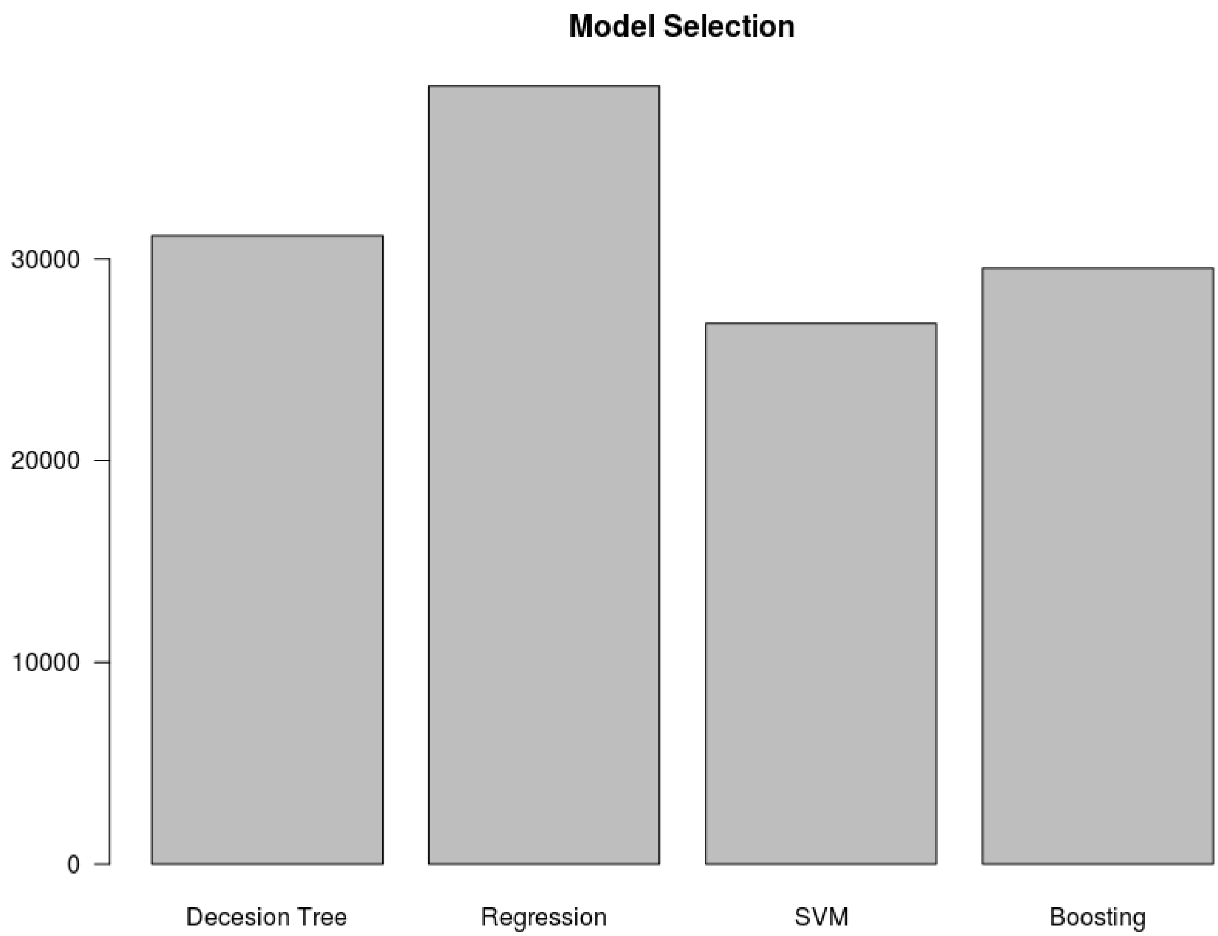

Modeling:

I tried 4 algorithms SVM, Bossting with laplace, Random forest and Regression.

Model Selection:
so at the end I found the SVM is the best model for this problem.

```
#The differnt between the predict value and the real value for all the algorithms
p1 <- (predict-train_label)*(predict-train_label)
p2 <- (predict2-train_label)*(predict2-train_label)
p3 <- (predict3-train_label)*(predict3-train_label)
p4 <- (predict4-train_label)*(predict4-train_label)
plot(p1,type="l",col="red")
par(new=TRUE)
plot(p2,type="l",col="blue")
par(new=TRUE)
plot(p3,type="l",col="green")
par(new=TRUE)
plot(p4,type="l",col="orange")
```






So SVM has the lowest error. And now we can predict the test file for predict the answers.