03.21.2018

Housing Price Prediction

Group 4

Members:

Yuexuan Huang

Lin Du

Xlaoyi Li

Keran Li

1. Executive Summary:

In this project we focus on predicting the house price because housing market affects economy mainly and housing price leads to significant changes in retail price inflation (Strobel and Vavra 2014). There are three business goals for our project. First, selecting variables that have greater impact on housing price. Simultaneously, building machines learning's models to predict reasonable price for people who want to sell houses. Eventually, finding potential relationship between housing price and inflation rate. After determining our business goals, we then went to search the data online. Compared with different set of data we chose a data with 79 explanatory variables describing every aspect of residential homes. The advantages of this data set are comprehensive and convinced.

In the process of processing data, we first explored the dependent variable---"SalePrice". We did histogram and made some simple conclusion. Then we analyzed the correlation between dependent variable and each independent variable. For these important features we also did some visualizations such as histogram and heatmap. After pre-processing the data, we went to next part---modeling.

To make our project more comprehensive we decided to run 4 different models which are ordinary least square regression, lasso regression, decision tree and random forest. Using cross-validation we calculated the Mean Square Error and R-squared for each model and chose the best model with lowest Mean Square Error and highest R^2. OLS is the best models after compared with others.

Our project uses data analysis techniques to process and visualization data, uses machine learning techniques to fit different models with different parameters, uses business methods to relate model to marketing. This project achieves our business goals perfectly and can be used as a business model in the marketing of housing in the future.

2. Business Understanding:

Although there is a lot of prediction about the trend of the whole housing market, a model which can assist a house seller to decide a reasonable price can be really helpful. Currently, models predicting house price usually based on the supply-demand relationship, which is not accurate enough. What we want to achieve through this project is to build a model involving more house features and increase the accuracy in deciding house price. Besides, there are many models using linear regression to predict price, but we want to try more advanced machine learning models such as lasso regression and decision tree. Also, these models can show more detailed relationship between housing price and GDP/inflation.

Also, we want to find other significant features effect on the housing price. Beyond the ordinary idea about the house area and location, there are many other features are worthy to be detected. We want to know other potential features, such as the building class and linear feet of street connected to property, and whether they can significantly influence the price or not. Besides, another important variable is time. We want to analyze whether or not the selling time in a year could affect the price.

Finally, using the model which contains many valuable features should be considered in, we can give home sellers an assistance to predict the reasonable price. Instead of comparing to some limited features with houses in the neighborhood, an individual home seller can easy get the interval of the price of his house with all the important features. This model also can help home buyers to have an evaluation of the houses which they want to buy. Therefore, this model can help to build a more efficient and more reasonable housing market.

Here is some potential problem with our model. First, the dataset is not big, only having 1400 rows about the housing data in one state in five years. Maybe there would have some bias and randomness in the process of predicting the housing. When people use this model to predict their houses, they should consider the specific condition in their area and the whole market situation. Plus, if we want to get a model with more accuracy, we should get rid of the influence of subprime crisis. After all, it affected housing price significantly. And if we could get data in a long period of time, a clearer trend of housing price change would be analyzed.

3. Data Understanding:

3.1 General information about the dataset:
Data Source:
https://www.kaggle.com/c/house-prices-advanced-regression-techniques

In this dataset, we have 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, and we have predictors, such as "GrLivArea", "GrLivArea" and etc., and predicted variable in the dataset called "SalePrice".

3.2 Explore the predicted variable: "SalePrice"

3.2.1 The distribution of sale price

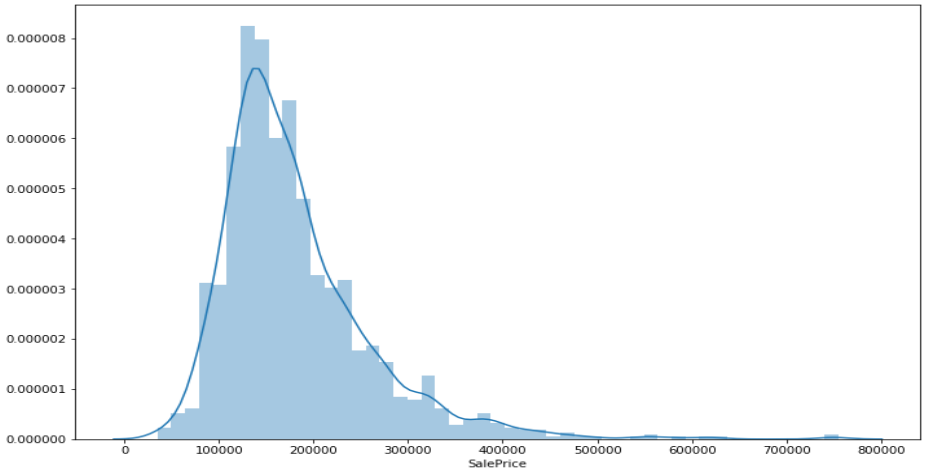

Figure 3.1

As we can see from the Figure 3.1, the sale prices are right skewed. This was expected as few people can afford very expensive houses.

3.3 The most significant numeric predictors from correlation matrix

- "SalePrice" correlation matrix

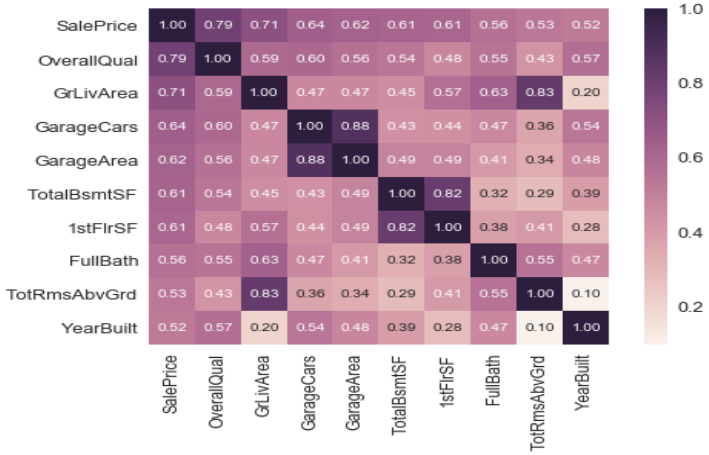

Figure 3.2

These are the variables most correlated with 'SalePrice'. According to the heatmap of correlation matrix, the darker the color, the higher the correlation between two variables. We have the following finding： 'OverallQual', 'GrLivArea' and 'TotalBsmtSF' are strongly correlated with 'SalePrice'. We should keep these variables in our model.'GarageCars' and 'GarageArea' are also some of the most strongly correlated variables. Therefore, we just need one of these variables in our analysis (we can keep 'GarageCars' since its correlation with 'SalePrice' is higher).'TotalBsmtSF' and '1stFloor', 'TotRmsAbvGrd' and 'GrLivArea' are highly correlated. We decided to keep only 'TotalBsmtSF' and 'GrLivArea' according their correlation.

- Scatter plots between “SalePrice” and correlated variables

Figure 3.3

In the Figure 3.3, here we should pay attention to the relationship between 'TotalBsmtSF' and 'GrLiveArea', we can see the dots drawing a linear line, which almost acts like a border, indicating that the majority of the dots stay below that line.

3.4 Relationship with categorical features

- Overall Quality

From Figure 3.4, as the overall quality increasing, the average sale price increases. But we may also find that the range of price is wider and wider as overall quality becoming larger and larger. In addition, from our correlation matrix, 'YearBuilt' is not highly correlated with 'SalePrice'. So we will not make it as time series in our model.

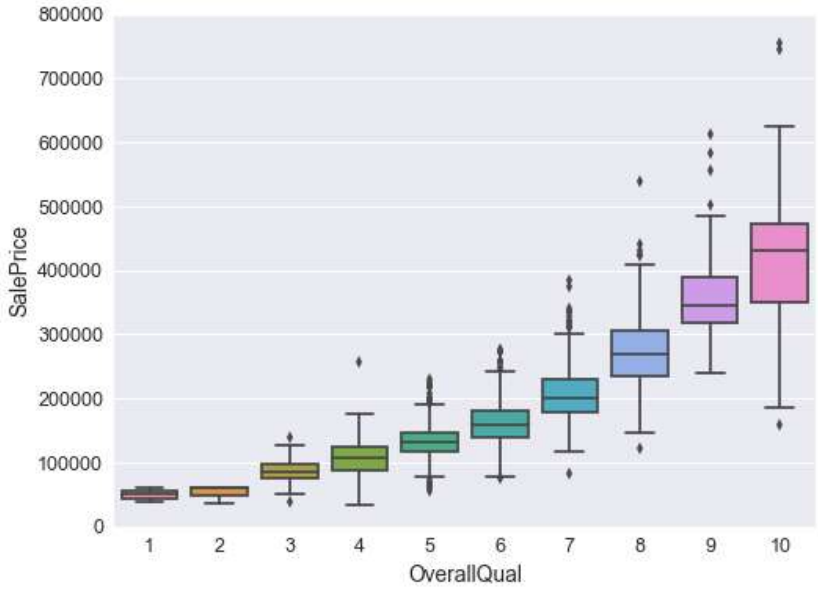

Figure 3.4

- Year Built

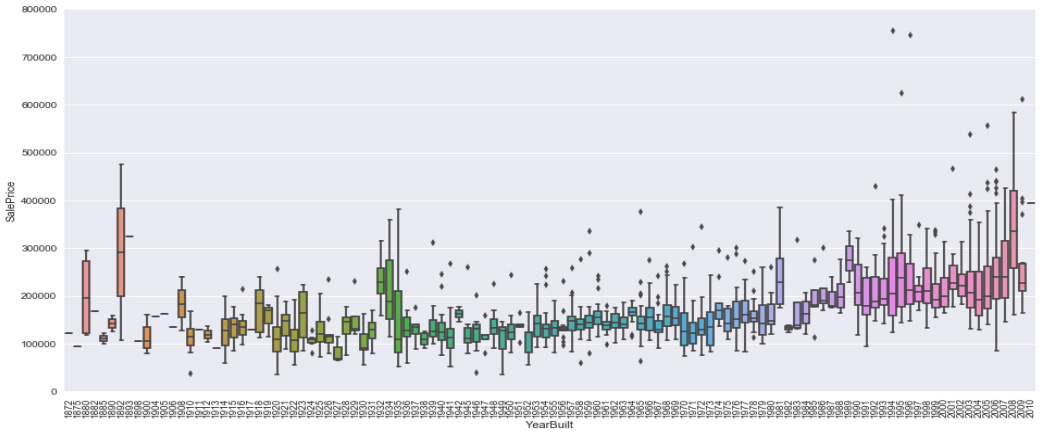

Figure 3.5

From Figure 3.5, although it's not a strong tendency, 'SalePrice' is still more prone to spend more money in new stuff than in old relics.

4. Data Preparation:

4.1 Missing Data:

Firstly, we made a table to check out the missing radio of each variable. And then we input each feature with missing values, after analyzing the meaning of their missing values. We used three methods to fill the missing values. For features which are categorial ones, NA means there is not have that furniture or device. For example, for PoolQC, data description says NA means "No Pool". Therefore, the huge ratio of missing value (+99%) makes sense, because majority of houses have no Pool at all in general. So, we fill the missing values with None. The same method applied to MiscFeature, Alley, Fence, FireplaceQu, LotFrontage, GarageType, GarageFinish, GarageQual and GarageCond, etc. For numerical features which have missing values, we set 0 into them. For those only have a few missing values, we filled them with the most common string or value. The last step is setting dummy categorical features. Finally, we got 221 variables.

4.2 Train & Test Split:

We use 80% of our dataset as training dataset and 20% as test data set.

5. Data Modeling:

5.1 OLS:

- Critical model selection
  In the data understanding we find lots of variables follow linear trend so linear regression model can shows optimal results. The aim of this model is to minimize the sum of the squared residuals.

- Model design & Implementation

1) First, In the part of data understanding we run the correlation matrix between dependent variable(price) and all independent variables. In this matrix we select top 9 variables that have higher correlation. Then, I run first OLS model only by these 9 variables and then using test date to get the Mean Error table
2) Next, we use forward selection method(AIC) to do all variable selection. In this process we lock 36 features and minimize the AIC to -5332.44. After choosing which variables should be used we then run second OLS model using training set. We test our second OLS model by test set and calculate the error.
3) Eventually we select top 10 important features among the 36 variables that we locked by AIC method last step.

- Model output & Strength

First OLS Model	ME	MSE	MAE	MPE	MAPE	R^2
Test Set	0.01456	0.19440	0.12626	-0.14419	1.05885	0.8236

(Error table of First OLS model)

Second OLS Model	ME	MSE	MAE	MPE	MAPE	R^2
Test set	-0.00685	0.17073	0.09949	-0.06762	0.83293	0.9474

(Error table of Second OLS model)

Figure 5.1

Compared with the error table between two regression models we think second linear regression model is better because all kinds of error are smaller than first one. Also second linear regression model is more comprehensive and accurate because it has a higher R^2.

We also select the top 10 important features which are , 'GrLivArea', 'TotalBsmtSF', 'OverallQual','YearBuilt' , 'LotArea','BsmtFinSF1','Fireplace', 'GarageArea', and 'GarageCars' and 'YearRemodAdd'

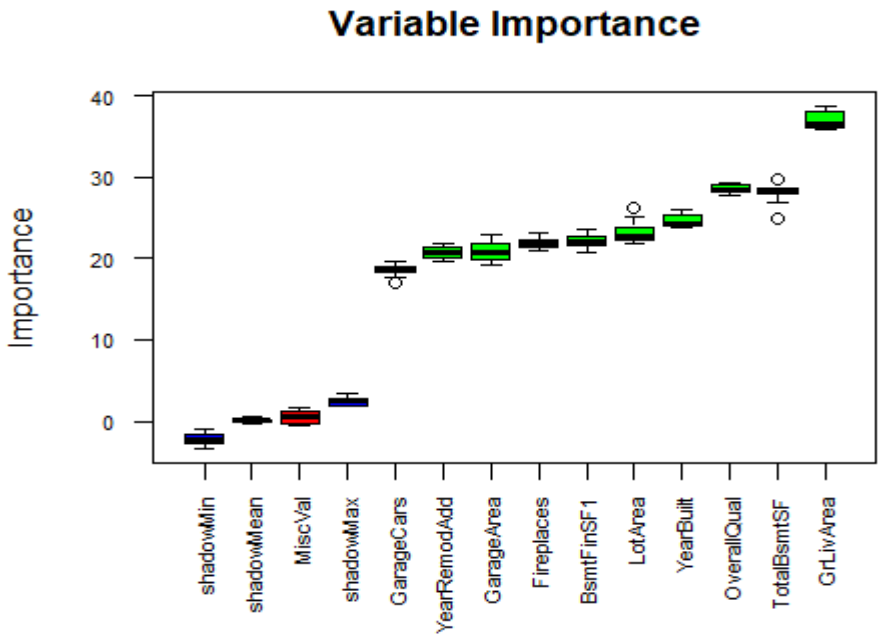

Figure 5.2

5.2 Lasso:

- Critical Model Selection:

For this dataset, after transferring character variables to dummy variables, it includes more than 200 variables, which means it is important to select the significant features for this model, providing simpler and easier interpreted model. Here we chose Lasso to improve the performance of linear regression, since Lasso can shift many insignificant coefficients to 0 and it can do the feature selection automatically. Also, with extra penalty, it's a better way to analyze data and capture relationships in the data and avoid overfitting.

- Model Design & Implementation:
    1. Train the Lasso Regression Model on train dataset
    2. Choose tuning parameter for Lasso Regression:

a. Define the tuning parameter range, here I chose lambda = (0.001,0.002,0.003,0.005,0.01,0.015,0.02).
b. Run the Lasso Model for each of these parameters.
c. Calculate the R-Square for each model and choose the one with highest R-Square (lambda= 0.005). Figure 5.3 above shows the R-Squared for each tuning parameter.

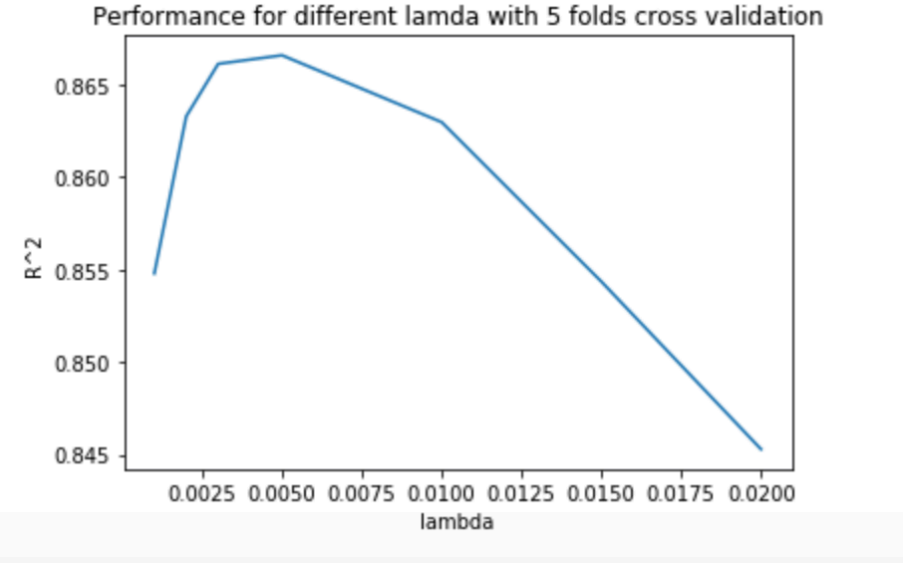

Figure 5.3

3. Train the Lasso Regression on train dataset with best tuning parameter
4. Calculate the MSE and R-Squared for the train and test dataset

- Model Output:
  MSE and R-Square for the train and test dataset as follows:

Train	MSE	0.31
Train	R^2	0.94
Test	MSE	0.28
Test	R^2	0.90

Figure 5.4
Top 100 important features and their coefficients as follows:

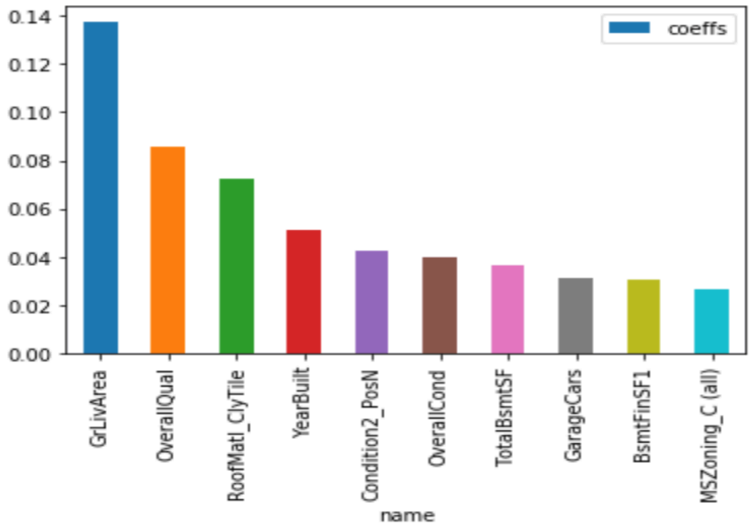

Figure 5.5

5.3 Decision Tree:

- Critical Model Selection
  Decision tree model is one of the simplest but most powerful tree-based models of machine learning methods. According to Wikipedia, decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.
  The advantages of decision tree model are as follow. First of all, when we fit a decision tree to a training set, the tree will automatically choose the most important variables within all the features and split according to the feature importance. Secondly, nonlinear relationships between parameters do not affect tree performance because decision tree does not depend on linearity assumption. Last but not least, decision tree model is easy to understand and interpret compared to other complicated machine learning models.

- Model Design & Implementation

After importing the training set and test set, we ran 5-fold cross validation to determine the parameter of max_depth, which stands for the maximum depth of the tree. We ran cross validation within the range from 1 to 30. The result is plotted below:

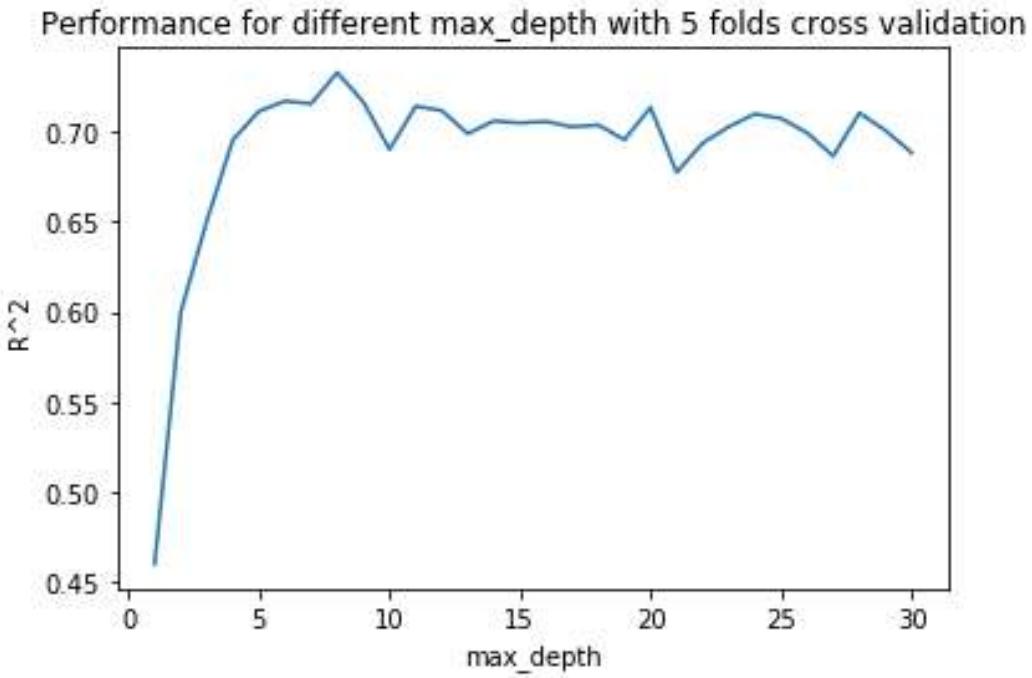

Figure 5.7

According to Figure 5.7, the R-squared increases shapely at the beginning, peaks at 7, then fluctuates around 0.7. Hence, we chose 7 max_depth as our parameter and applied the model in the test set.

- Model Output

  After comparing the prediction results with test set sale price, our model got 0.759 R-squared and 0.035 mean square error.

  We also extracted the most important 10 features within the decision tree model, which is 'OverallQual', 'GrLivArea', 'TotalBsmtSF','CentralAir', 'BsmtFinSF1', 'GarageArea', '2ndFlrSF', 'OverallCond', 'MSZoning_C (all)' and 'GarageCars'.

These feature importance is plotted below:

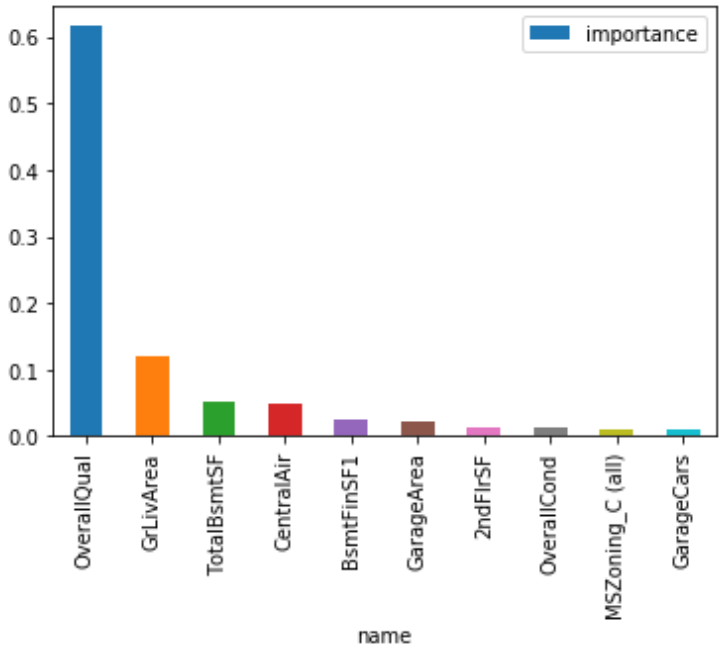

Figure 5.8

We can find from the Figure 5.8 that 'OverallQual' is much more important than other features in our predictive model.

5.4 Random Forest:

- Critical Model Selection
  Random forests construct a multitude of decision trees at training time and get the mean prediction of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set and reduce the variance.

- Model Design & Implementation
  The training algorithm for random forests applies the general technique of bagging to tree learners. Which means it repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples. After training, predictions for unseen samples can be made by averaging the predictions from all the individual regression trees.

- Model Output
  The mean squared error is 0.0177, and the R squared is 0.893. Both are very good, which means the random forests regression model fits the data very well. And using this model, we got 10 features which are very important, and they are shown in Figure 5.9:

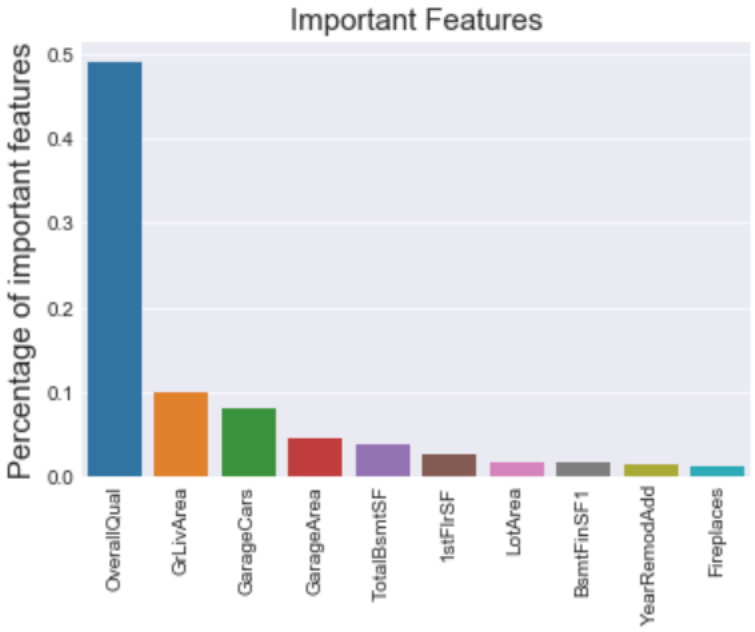

Figure 5.9

6. Performance & Evaluation:

6.1 Model Performance Comparison

	OLS	LASSO	Decision Tree	Random Forest
R-squared	0.9474	0.901	0.759	0.893
MSE	0.1707	0.2815	0.035	0.0177

Figure 6.1

In the field of R-squared, among all the model we used, OLS has the highest 0.9474 R-squared, which means the regression line is highly fitting to the data. On the other hand, in the field of MSE, Random Forest model can do the best job, with 0.0177 mean squared error, which stands for the high quality of our estimators.
Such consequence is not surprising, because we found from data understanding that lots of variables followed linear trend so linear regression model can show optimal

results. In addition, we have more than 70 independent variables in our model, the more variables we have, the more accuracy the Random Forest model. Overall, we think both of OLS and Random Forest model can have great performance in housing prediction with OLS can perform highest R-squared and Random Forest will have lowest MSE.

6.2 Feature Selection

Figure 6.2 (*The dark labels are important features for all four models)

Variable Explanation:
GrLivArea: Above grade (ground) living area square feet
OverallCond: Overall condition rating
TotalBsmtSF: Total square feet of basement area

	OLS	Lasso	Decision Tree	Random Forest
First important	GrLivArea	GrLivArea	OverallQual	OverallQual
Second important	TotalBsmtSF	OverallQual	GrLivArea	GrLivArea
Third important	OverallQual	RoofMatl_ClyTile	TotalBsmtSF	GarageCars
Fourth important	YearBuilt	YearBuilt	CentralAir	GarageArea
Fifth important	LotArea	Condition2_PosN	BsmtFinSF1	TotalBsmtSF

YearBuilt: Original construction date
LotArea: Lot size in square feet
GarageCars: Size of garage in car capacity
GarageArea: Size of garage in square feet
CentralAir: Central air conditioning
BsmtFinSF1: Type 1 finished square feet
Condition2: Proximity to main road or railroad (if a second is present)
RoofMatl: Roof material

From the Figure 6.2, we found that 'GrLivArea' and 'OverallQual' are the top 2 important features that show on all four models. 'TotalBsmtSF' and 'YearBuilt' also play significant roles when pricing the house. In the data understanding we built a correlation matrix between 'SalePrice' and all other features. In that matrix we also found that 'OverallQual', 'GrLivArea' and 'TotalBsmtSF' are strongly correlated with 'SalePrice'. So the results of the analysis almost the same.

6.3 Internal Strength and Improvement:

- There are only 1460 observations in the data set. So we need more data to improve the degree of accuracy.
- Trying to find more complete data with less missing value and improving the data quality.
- Finding more features related to house pricing. Adding more features to analyses the data.
- Using more models to do the prediction, such as Support vector machine and Linear discriminant analysis.

7. Technical Appendix & Files

- OLS Regression:
  OLS is a popular method in statistic. The main idea is to estimate the unknown coefficients in the linear model. OLS chooses the best linear model by minimizing the Sum of squares residual between the actual y value and predicted y value. We ran OLS model using R. We used 'forecast' package to predict the test data and to get the error table. We used AIC method to select the best model. We also used 'Boruta' package to analyze the top 10 important variables in the model.
- Lasso Regression:
  Lasso Regression is an improvement of Linear Regression by adding an extra penalty to achieve the model regulation. The objective of lasso is to solve following functions:

$$\min_{\beta_0,\beta}\left\{\frac{1}{N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2\right\}\ \text{subject to}\ \sum_{j=1}^{p}|\beta_j|\le t.$$

  For Lasso Regression, it is also important to choose a good tuning parameter and here we use cross validation to solve this problem.
  In python, we use sklearn package of lasso regression and cross validation and matplotlib package to do visualization. Programming codes can be found in code file.

- Decision Tree:
  According to Wikipedia, decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. We applied decision tree regression in our dataset with Python. We used packages numpy, pandas and matplotlib to import the training set as well as test set and plot our results. Relating to modeling, we used package sklearn to build our model, applied cross validation and extract important features as well as prediction accuracy.
- Random Forest:
  Random Forest is an improvement model of decision, since it separate data into may subgroups. Each decision tree in the forest considers a random subset of features when forming questions and only has access to a random set of the training data points. This increases diversity in the forest leading to more robust overall predictions. And we achieve this algorithm using sklearn random forest package and using matplotlib package to visualize it.
- Related Files:
  Code File: “Code_Python.py”, “Code_R.rmd”
  Data File: “X_train_dummy.csv”, “X_test_dummy.csv”, “y_train_dummy.csv”, “y_test_dummy.csv”, “train_ols.csv”, “test_ols.csv”
  Instruction: “Instructions.pdf”