House Pricing in Ames, Iowa

ACTUPS 5841 Data Science in Finance & Insurance

Professor Yubo Wang

Woon Sup Kim (wk2371)

Emmelyn Luveta (el3132)

Owen Ou (jo2641)

Elsie He (xh2472)

I. Introduction

Regarding reasons for choosing this topic, buying or renting a house is a vital part of almost every person’s life. If a person asks anyone about their dream house, they probably will not start with the material of the roof, size of swimming pool, or proximity to the railroad. In fact, there are a lot of factors that we can pay attention to and affect housing prices. The project aims to analyze the important factors of house pricing. Understanding these factors will help people navigate dynamically changing environments and devise appropriate strategies.

II. Data Description

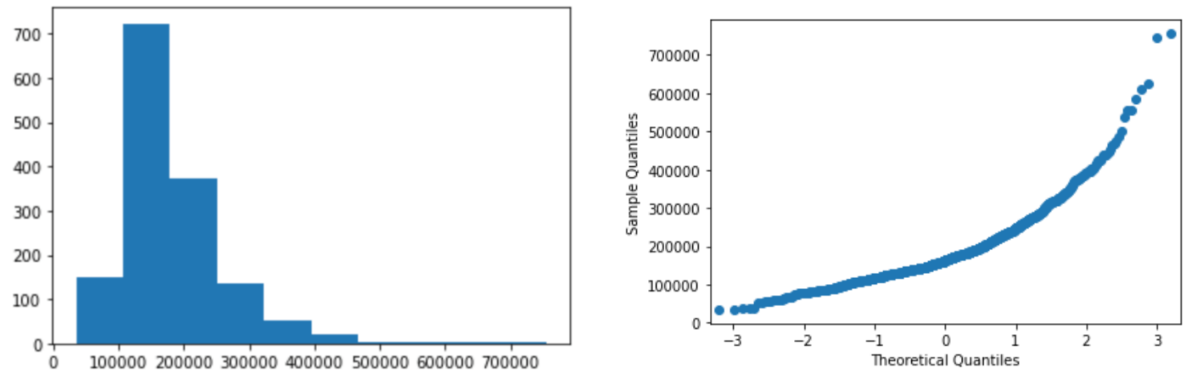
The Ames Housing dataset was compiled by Bart De Cock for use in data science education in 2011. It includes 1460 data points, 80 features, and 1 response variable, SalePrice. And it is the dataset of house prices in Ames, Iowa from 2006-2010.

III. Data Preprocessing

Response Variable

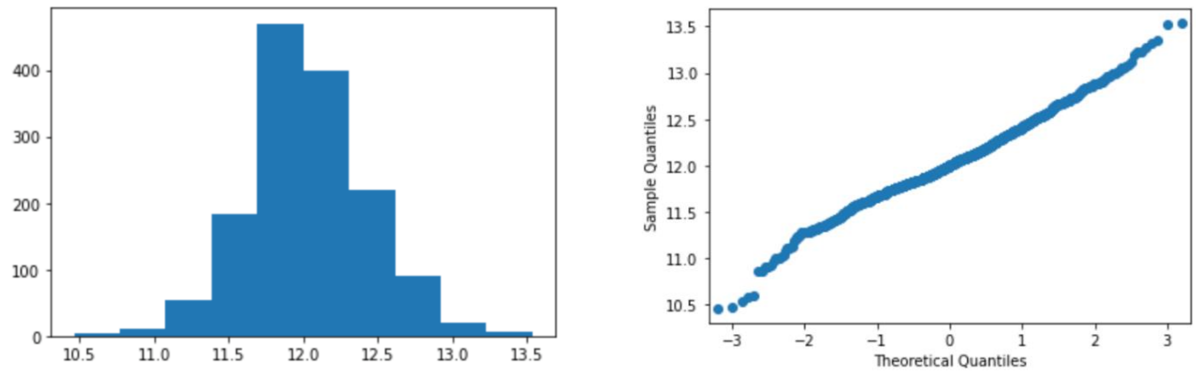
Before: SalePrice: Skew=1.8809

```
λ =  1.8829
```



After: Log(SalePrice): Skew=0.1212

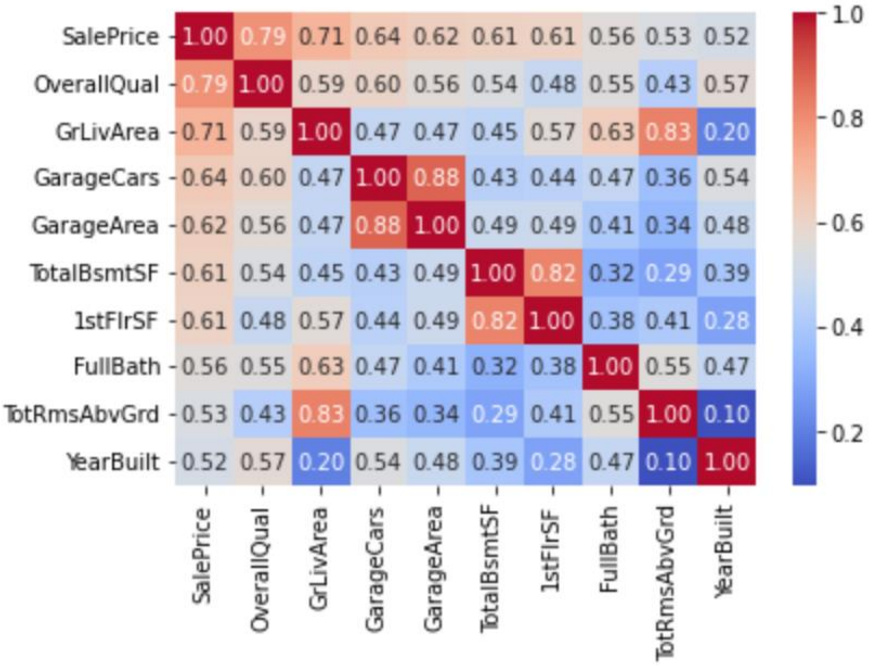
```
λ =  0.1213
```



At the beginning, we conducted data observation and cleaning. The SalePrice was right skewed, which was expected as few people could afford very expensive houses. The skewness 1.88 of the response variable was too high and the Q-Q plot showed SalePrice was not normally distributed. To fix it, we took log transformation of SalePrice. The skewness after transformation was now quite low and the Q-Q plot also looked better. We conducted the log transformation since regression assumes multivariate normality that means regression requires all its variables to be normal. And having skewed data violates the assumption of normality. We also normalize our explanatory variables

Explanatory variable

1. Correlation (Numerical)



We drew a heat map to analyze the correlation between numerical explanatory variables and the response variables (SalePrice) for a basic understanding of the data. We can see there are strong collinearities among those features, which suggest some data preprocessing is needed.

We remapped the MSSubClass column since according to the data description it is supposed to be a categorical variable but has been recorded as integers (20,30,40, etc.). Based on correlation graph and data observations, We combined and integrated some variables.

a. YearBuilt (original construction date), YearRemodAdd(Remodel date), and YrSold: we used two columns instead to represent those three columns: Remod and HouseAge. The first one indicates whether the house has been remodeled or not, and the second one is calculated by YrSold minus the remodel date.

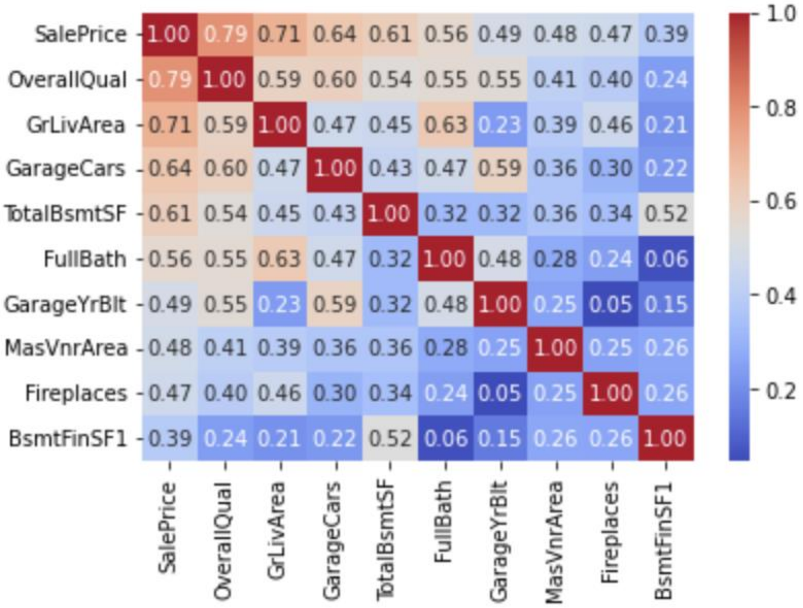
b. We also observed 4 columns which represent the number of bathrooms, basement full bathrooms, basement half bathrooms, full bathrooms above grade and half bathrooms above grade. We added those four features together and multiplied each half bathroom column by 0.5, so we can furtherly reduce 3 columns over here.

c. For those highly dependent pairs, like TotRmsAbvGrd and GrLivArea (cor = 0.83), GarageArea and GarageCars (cor = 0.88), we select one from each pair which has higher correlations with SalePrice. Therefore, according to our heatmap, we dropped TotRmsAbvGrd and GarageArea.

d. For variables 1stFlrSF, 2ndFlrSf and GrdLivArea, we find there is only approximately 1% of data that total above ground does not equal to first + second floor area, so we dropped the columns for first and second floor features.

Finally, we reduced the number of features from 80 to 72. At last, we also centralized and normalized all numerical features and imputed all the missing values including categorical features.

2. Correlation again



After the above processing, we graphed the correlation heat map again. The color patterns became more congruent and there are no correlations that were as high among explanatory variables as our previous heat map.

IV. Feature Selection

After completing the data cleaning and after encoding all the categorical features into one-hot encoding, there remained 333 features. In order to narrow down the features into only the ones that were good predictors of the response variable, a Lasso regression was conducted to regularize the features. This was an important step to avoid multicollinearity and to reduce the standard error of the regression models. The Lasso regression was conducted in two different ways:

1. Conducting the Lasso regression only on the numerical features and combining the encoded categorical variables to the data set.
2. Conducting the Lasso regression on the entire dataset including the encoded categorical variables.

When a simple linear regression was done to compare the two approaches, the latter returned a smaller 10-fold cross-validation RMSE, and therefore this approach was used.
As a result of the Lasso regularization, the number of features was reduced to 117.

Some of the variables that were removed include:

1. LotFrontage: Linear feet of street connected to property
2. BsmtUnfSF: Unfinished square feet of basement area
3. MSSubClass: The building class

This regularized data set of 1460 observations of 117 features will be used for predicting the response variable in the next section.

V. Modeling

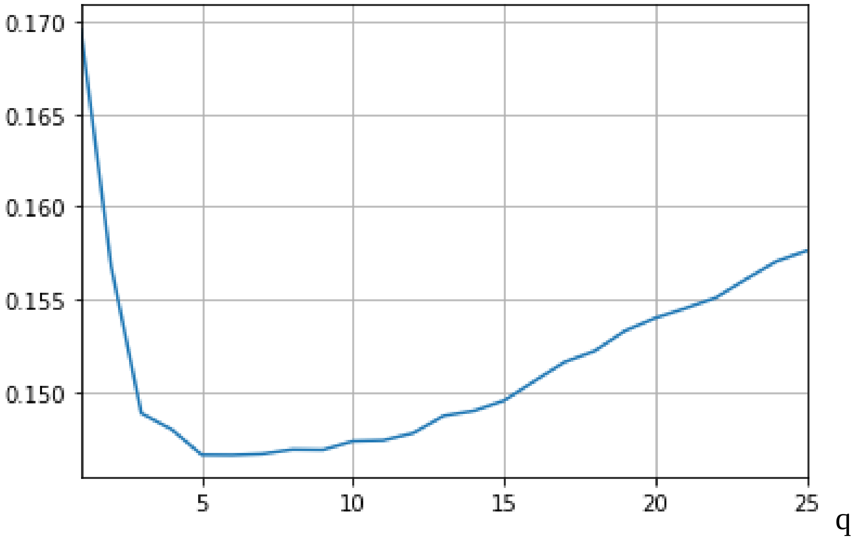
As it is difficult to visually inspect the relationship of the response variable to all the features of the dataset, our approach to the prediction was to model the response variable using various regression methods and to compare the cross-validation RMSE values to determine which model is the most appropriate.

Linear Regression

As we have seen in the “Data Preprocessing” section above, the log transformation of the response variable was approximately normal. Therefore, we expect that a linear regression model will be a good predictor. Furthermore, the correlation plot revealed that there were obvious features that were highly correlated to the response variable. The linear regression resulted in a train RMSE of 0.1023 and a 10-fold cross-validation RMSE of 0.1326.

K-Nearest Neighbors Regression

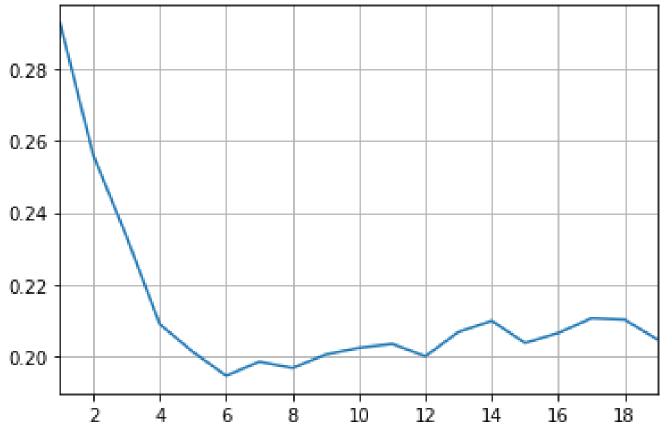
Intuitively, regardless of the distribution of the data, it would make sense for houses with similar features to be similar in price. Therefore, the k-nearest neighbors regression was used to predict the house prices. In order to find the optimal hyperparameter for the model, the KNN regression was conducted using various numbers of “nearest neighbors” and was verified against the 10-fold cross-validation RMSE. The plot below shows the change in 10-CV RMSE against the number of “nearest neighbors” used to model the KNN regression.



The plot resulted in the minimum 10-CV RMSE when 6 nearest neighbors were used to model the KNN regression model. Using 6 nearest neighbors, the KNN model resulted in a train RMSE of 0.1334 and a 10-fold cross-validation RMSE of 0.1474.

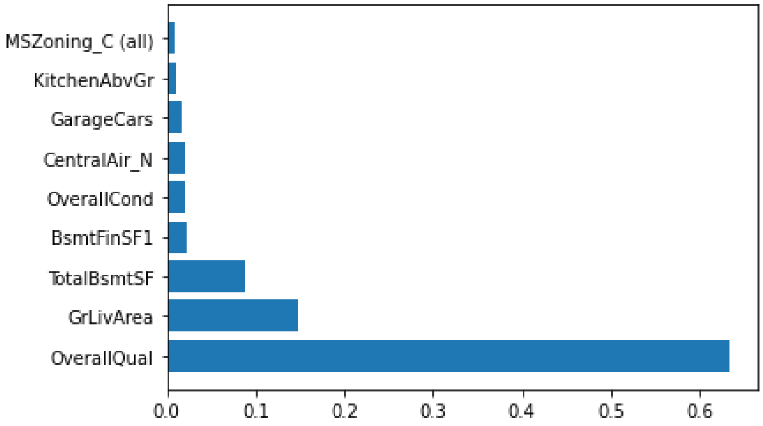
Regression Tree

Regression Tree was another non-parametric candidate model that we took under consideration. The regression tree may capture some non-linear relationships between the response variable and the features. Moreover, it is believed that the regression tree more closely mirrors human decision-making compared to the other models above.
Similar to the KNN, a 10-fold cross-validation was used to find the optimal hyper parameter of the regression tree. The plot below shows the change in 10-CV RMSE against the max depth of the tree.



The plot resulted in the minimum 10-CV RMSE when the max depth of 6 was used to model the tree regression model.
Using 6 as the max depth, the tree regression model resulted in a train RMSE of 0.1343 and a 10-fold cross-validation RMSE of 0.1924.

Another unique advantage of the regression tree is its interpretability. The plot below shows the most important features according to the regression tree.

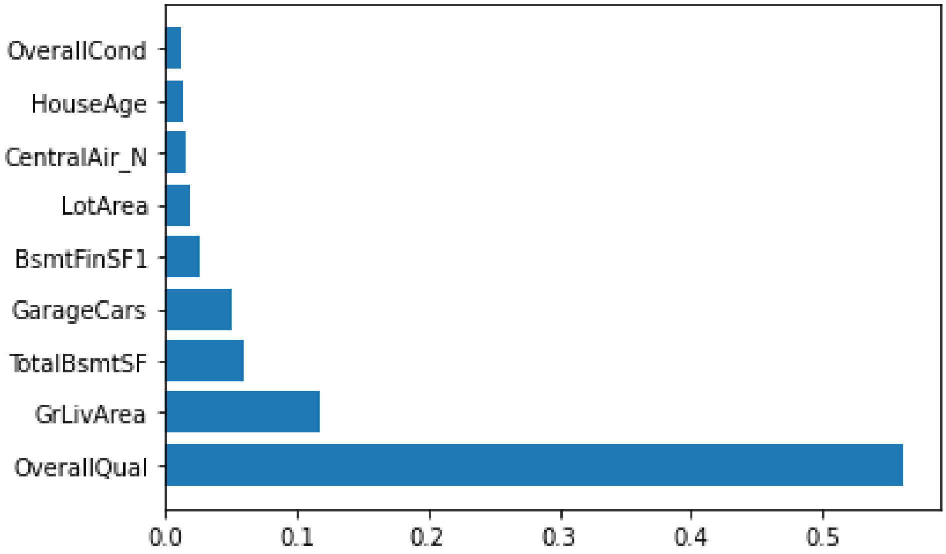


While a regression tree has many benefits, it is worth noting that it is considered a weak learner due to its high variance. To overcome this weakness we will consider ensemble methods in the following two models.

Random Forest

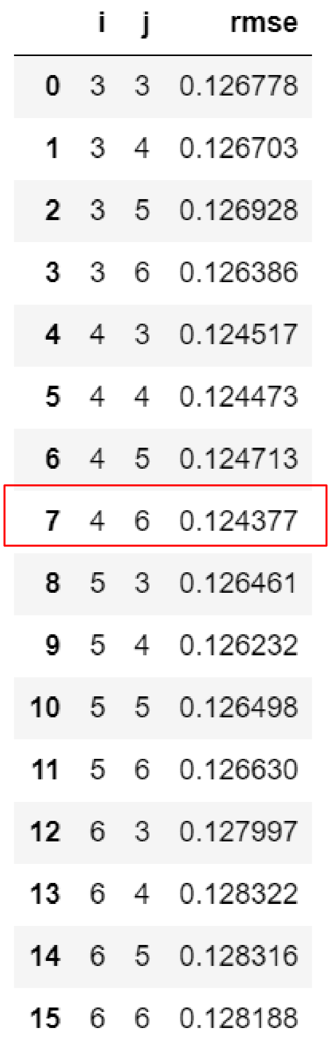
From the above bar plot of the important features according to the regression tree, it is apparent that the feature “OverallQual” is a very strong predictor. Therefore, simply using bagging methods will not make much improvements as all the bagged trees will be highly correlated. This is the reason why a Random Forest approach was used instead of the traditional bagging methods.
Sklearn’s default method for Random Forest Regressor was used. (100 trees).
This resulted in a train RMSE of 0.05217 and a 10-fold cross-validation RMSE of 0.1414.

The interpretation of these ensemble methods are much more difficult than a single tree. The plot below shows the summary of the top 9 important features based on reduction in the Gini index.



Gradient Boosting

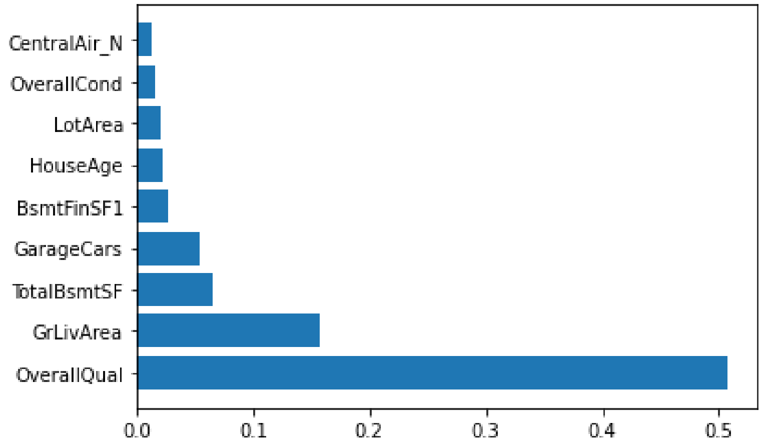
The Random Forest model showed a significant improvement on the RMSE. However, the large difference in the train RMSE and the 10-CV RMSE suggested that this model would not perform well with unseen data. Therefore, a boosting method was considered. As boosting methods learn from previous models and make improvements, we can be confident that it will return a good prediction. However, there exists a few challenges with Gradient boosting such tuning various hyper-parameters and it is very computationally expensive. For our modeling, we selected the hyper-parameters “max depth” and “min samples split” using the 10-fold cross-validation as we have done for our previous models while keeping all other hyper-parameters as its default according to the sklearn’s Gradient Boosting Regressor. The table on the right shows the change in 10-CV RMSE against the max depth “i” and the min samples split “j”.



	i	j	rmse
0	3	3	0.126778
1	3	4	0.126703
2	3	5	0.126928
3	3	6	0.126386
4	4	3	0.124517
5	4	4	0.124473
6	4	5	0.124713
7	4	6	0.124377
8	5	3	0.126461
9	5	4	0.126232
10	5	5	0.126498
11	5	6	0.126630
12	6	3	0.127997
13	6	4	0.128322
14	6	5	0.128316
15	6	6	0.128188

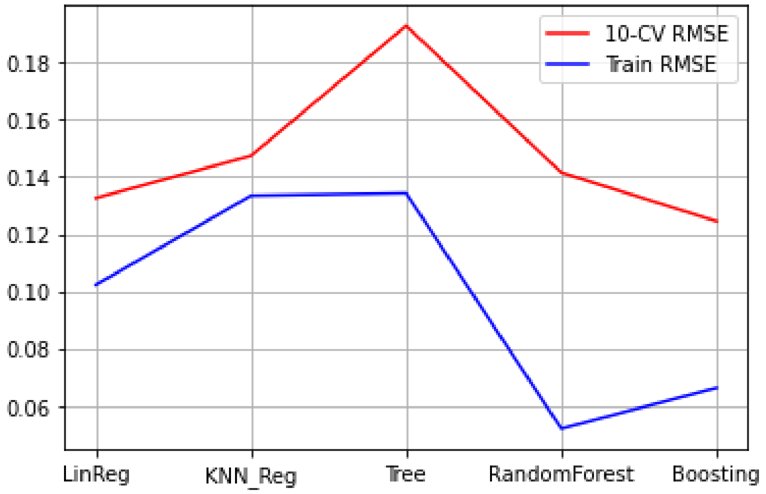
Using 4 max depth and 6 min samples split resulted in a train RMSE of 0.06630 and a 10-fold cross-validation RMSE of 0.1245.

The plot below shows the most important features according to the gradient boost method.



VI. Model Performance

The plot below summarizes the performance of each of the modeling techniques that we have used.



The Random Forest model had the lowest train RMSE of 0.05217, while the Gradient Boosting model had the lowest 10 cross-validation RMSE of 0.1245. Although the Random Forest had the lowest train RMSE, its high 10 cross-validation RMSE suggests that the model may overfit the training data and would not perform well against new data. Therefore, the Gradient Boosting is concluded to have the best performance.

VII. Conclusion

Based on our dataset of house prices in Ames, Iowa, the Gradient Boosting model lists the following features to be most influential in housing prices:

1. "OverallQual": Overall material and finish quality
2. "GrLivArea": Above grade (ground) living area square feet
3. "TotalBsmtSF": Total square feet of basement area
4. "GarageCars": Size of garage in car capacity

Therefore, when it comes to the valuation of the house, whether it is for purchase or for sale, these features are the most important to be considered and must be paid attention.

VIII. Future Research

There are at least three ways our models can be improved to yield better predictions:

1. Conducting Principal Component Analysis in our data pre-processing step to summarize the observations using only a few, uncorrelated features.
2. Tuning hyper-parameters for Random Forest and Gradient Boosting to improve their performance.
3. Conducting bagging methods with Random Forest regressor to avoid the overfit problem.

IX. Appendix

Dataset: House Prices: Advanced Regression Techniques.