House Price Prediction Using Machine Learning

University of California Los Angeles

Master of Applied Economics

Chang Cong

Dr. Patrick Convery

June 6, 2019

Table of Contents

1. Abstract

Purchasing a real estate is an important decision both in terms of having a home and investment. Many people devote their whole life to buy a dream house. Although the reasons for owning real estate vary, buying and selling a real estate is one of the biggest purchases of an average person. Buying and selling a house at a reasonable price is financially important to everyone. With quantitative and qualitative attributes of a property like how many bedrooms, how, many square feet of the first floor, and what location, I estimated a fair price using the technique of Machine Learning.

2. Introduction

Traditionally, when people decided to list their house or buying a house, the real estate agents would help them determine the value of the house using a comparative market analysis method, which is normally conducted by comparing the house with recently sold properties with similar characteristics in the same area. In order to perform the comparison, the appraiser needs to carefully look at what those houses have in common. For example, it would be better to compare the listing house with those have the same number of bedrooms, bathrooms and the same size. Also, the appraiser needs to reconcile the differences between the listing house and the compared houses. However, you can never find two identical properties and the adjustments for the differences are sometimes biased and subjective. Thus, finding a method that is more accurate and independent of the judgment of the appraiser or real estate agents would be good news for those are willing to sell or buy a real estate.

In this project, I am going to solve this problem using machine learning. The dataset I use originates from Kaggle. This dataset provides a detailed description of the residential home in Ames, Iowa from 2006 to 2010. The dataset has a total of 1460 observations and 80 features.

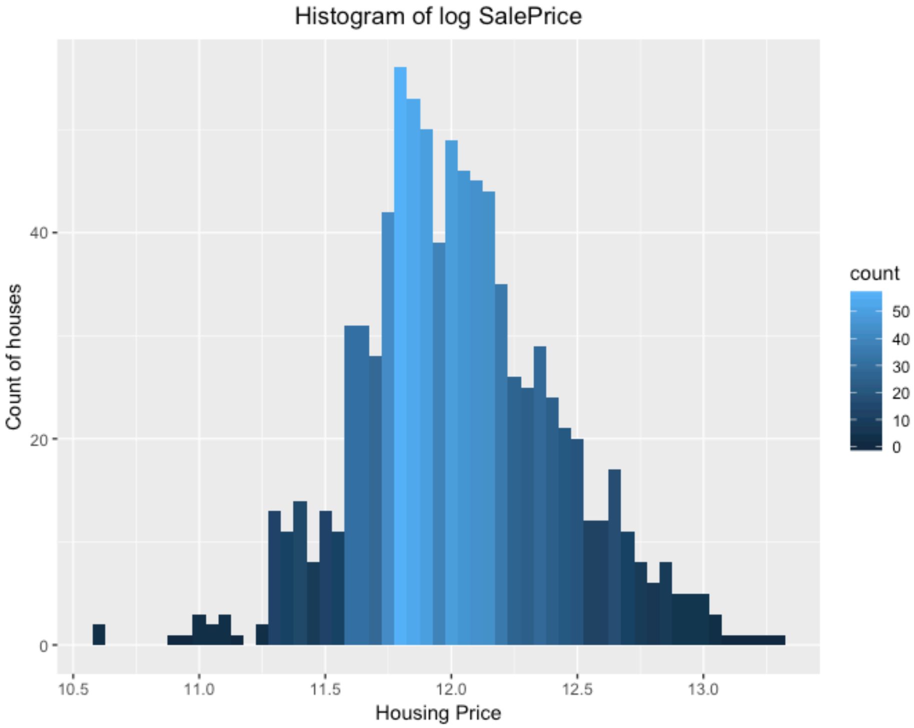
In this report, I will start with a brief summary and introduction of the task. Then I will provide a summary for the dataset and describe the data. Since the dataset contains more than 60 features, I would select features that are most related to the target variables using several techniques and remove those features that are not quite relative to the target variables. In order to find out a model that has the highest accuracy, I developed several models, evaluated the performances of those models. After comparing those models, I kept the model that enjoys the highest predictive power.

3. Data Description

There are two data types in this dataset: numeric and categorical. 46 features are categorical. For instance, there are categorical features identifying the type of dwelling involved in the sale, the general zoning classification of the sale, type of utilities available, physical locations within Ames city limits, proximity to various conditions, exterior covering on house, style of dwelling, roof material and so on. The rest of the features (33 out of 79) are all numerical. 13 of the numerical features are discrete and 20 of them are continuous. The discrete features mainly quantify numbers of bedrooms above grade, numbers of kitchens and full bathrooms, and also more detailed information like numbers of fireplaces and number of basement half bathrooms and so on. The continuous variables recorded the linear feet of street connected to the property, the year garage was built, the size of garage in car capacity, total square feet of the basement area, first and second floor square feet, low quality finished square feet, open porch area in square feet, screen porch area in square feet and so on.

Before exploring the data, I first deal with the missing values. Among 80 features, 19 of them have missing values. For the numeric features, I replaced the missing value with the average value of the data, and for the categorical features, I replaced the missing value with “None”.

The target variable is the sale price of the property. The value of those properties ranges from $34900 to $755000 with a average of $180921. The sale price is right-skewed and most of the price clustered around $100000 to $250000. The right skewed target variable may cause problem when doing further analysis. In order to solve this problem, I took a log of the sale price. As we can see from the histogram of the log sale price, it was transformed into a normal distribution.



4. Data Exploration and Features Selection

Among the 79 features, the class of 36 of the features is an integer, and the class of the rest of the features is all character. Before building models, I selected the features that are most relevant to the target variable. Some features are correlated with each other. For example, rates the overall material and finish of the house kind of decide rates the overall condition of the house, the quality of the material on the exterior has something to do with the present condition of the material on the exterior. Thus, it would generate multicollinearity problems if I keep all of them. Also, some features may lack predictable power to the target variable or barely explain the target variables. Those variables should be identified and removed as well.

I tried three feature selection methods: chi-squared, Extra Trees and AIC feature selection. Then I got three sets of 20 best features.

Chi-squared: I used SelectKBest and chi-squared score function provided in python scikit-learn library. This function tests the level of dependency of each variable to sale prices and gives a score to each feature. I ranked all features according to its scores and kept 20 best features that had the highest scores, which means relatively highly related.

Extra Trees: I used python built-in feature_importance_ attribute of ExtraTreeClassifier class to evaluate the importance of each variable to the Sale Price.The algorithm behind Extra Trees is that it “builds an ensemble of unpruned regression trees according to the classical top-down procedure. ”(Pierre, Damien & Louis, 2006).

AIC: AIC, stands for Akaike information criterion, serves as a commonly used mean for model selection. In this method, I found the best model that had 20 features. The results of these three methods are shown in this table.

method	Selected features
Chi-squared	'LotArea', 'MiscVal', '2ndFlrSF', 'BsmtFinSF1', 'PoolArea', 'BsmtFinSF2', 'BsmtUnfSF', 'LowQualFinSF', 'GrLivArea', 'TotalBsmtSF', '3SsnPorch', 'ScreenPorch', 'WoodDeckSF', '1stFlrSF', 'EnclosedPorch', 'GarageArea', 'OpenPorchSF', 'MSSubClass', 'Neighborhood', 'Exterior2nd'
Extra Trees	'MoSold', '1stFlrSF', 'BsmtUnfSF', 'LotArea', 'GarageArea', 'GrLivArea', 'YearRemodAdd', 'YrSold', 'YearBuilt', 'TotalBsmtSF', 'BsmtFinSF1', 'TotRmsAbvGrd', 'Neighborhood', 'OpenPorchSF', 'OverallQual', 'WoodDeckSF', 'Exterior2nd', 'BsmtFinType1', 'Exterior1st', '2ndFlrSF'

AIC	'KitchenAbvGr', 'GarageCars', 'RoofMatl', 'RoofStyle', 'KitchenQual', 'CentralAir', 'OverallQual', '1stFlrSF', 'GarageQual', '2ndFlrSF', 'Condition2', 'LandSlope', 'GrLivArea', 'Utilities', 'BsmtFullBath', 'PoolQC', 'Street', 'BsmtHalfBath', 'FullBath', 'LowQualFinSF'

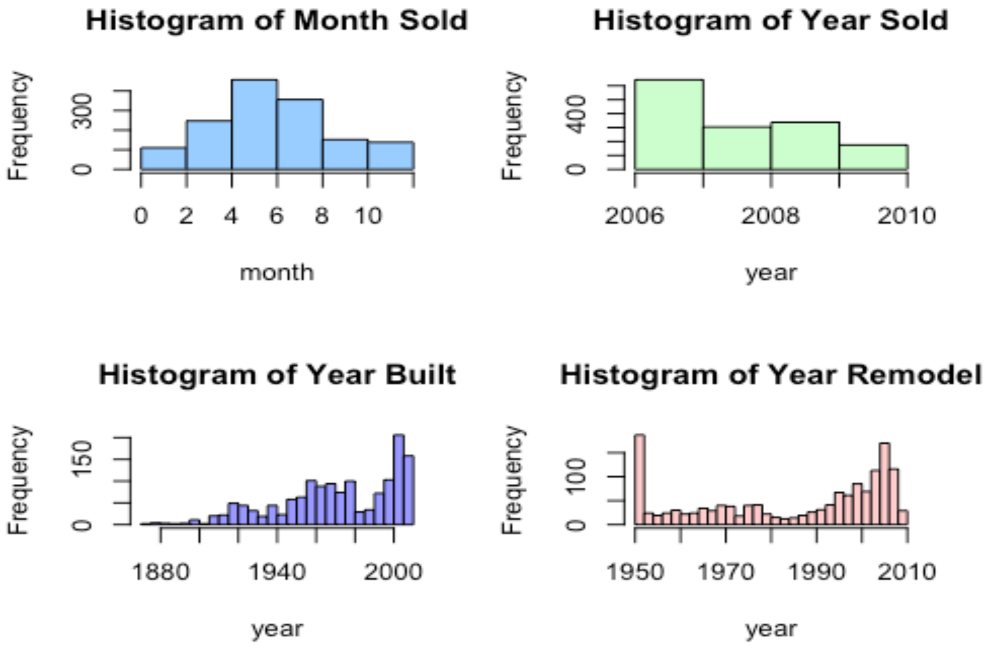
According to the table, we can see many interesting findings. There is a great overlap among the selected features in the three methods. The features 2ndFlrSF(Second-floor square feet), 1stFlrSF(First Floor square feet), and GrLivArea(Above grade (ground) living area square feet) were selected in all the three methods. Another finding is that many features selected are related to the area in square feet. For example, LotArea evaluates the size of lot in square foot, GarageArea measures the size of a garage in square foot, PoolArea stands for the size of the pool in square foot, BsmtFinSF1 is the area of the finished basement area.

After comparing the results of these three methods, I decided to choose the second model. Unlike the other two methods, the second method considered the month and year sold, original construction date and remodel date. Rather than simply considered the layout of a property, usually in practice, the year that a property was built and remodeled have a great impact on the functionality of a property and is also a great determining factor in buying a house. Also, a property's price could be affected by the month and year sold.

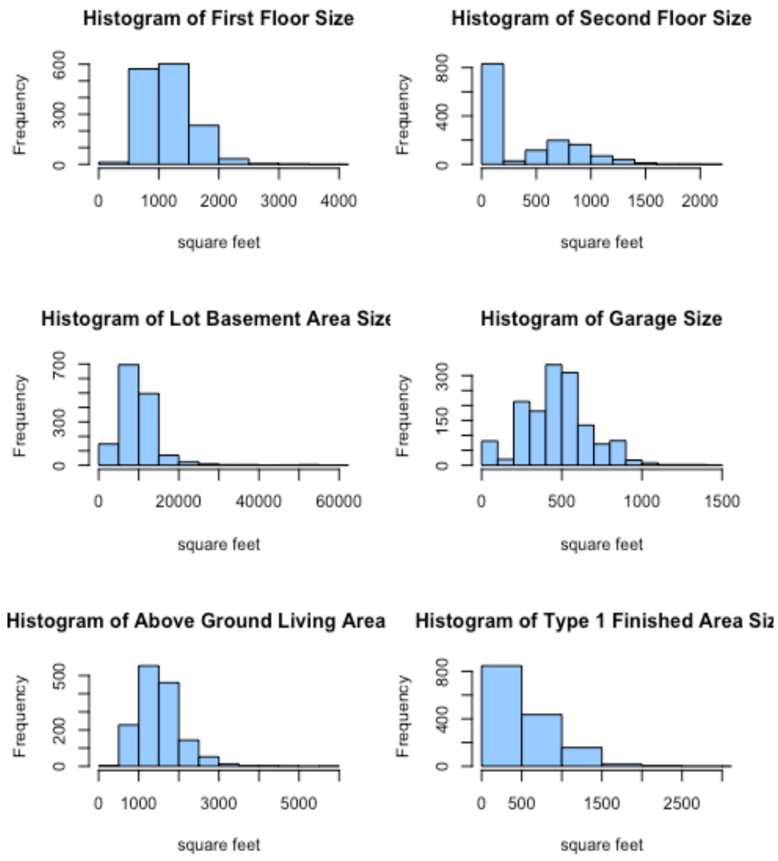
The 20 features selected were: Month Sold, Year Sold, Year Built, Remodel date, First Floor square feet, Second floor square feet, Unfinished square feet of basement area, Lot size in square feet, Size of garage in square feet, Above grade (ground) living area square feet, Total square feet of basement area, Type 1 finished square feet, Wood deck area in square feet, Open porch area in square feet, Total rooms above grade, Physical locations, Rates the overall material and finish of the house, Exterior covering on house, Exterior covering on house (if more than one material), and Quality of basement finished area.

5. Data Visualization

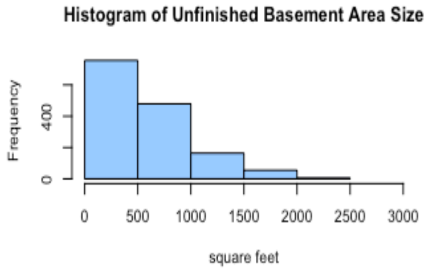
As shown in the graph, nearly half of the properties were sold between April and August, only a very small fraction of the house were sold in the first two months. The housing market was most booming during 2016, then from 2017 to 2010, the housing market has undergone a recession. This may be the result of the house bubbles. Most of the houses sold were built in the recent 60 years. In the year of 1950, around 200 of the houses were rebuilt, but after that, for about nearly 40 years, people seemed to lost interest in remodeling their houses.

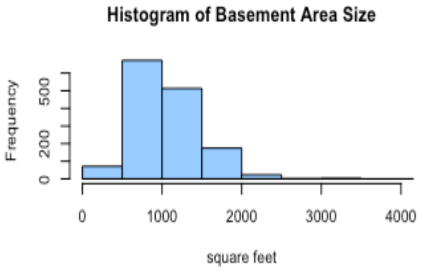


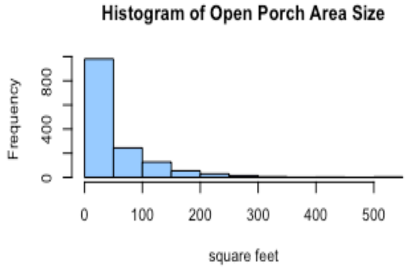
As we can see from this plot, the size of the first floor is mainly around 500 square feet to 1500 square feet and the size of the second floor is mostly smaller than 250 square feet. The mean of lot basement size is around 1000 square feet while the mean size of the garage on average is 500 square feet. Most of the ground living area for a house is 1000 to 2000 square feet. About 90 percent of the size of type 1 finished area is smaller than 1000 square feet. There are plenty of unfinished basement area on average and usually less than 1500 square feet.

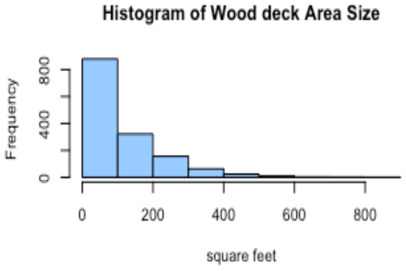


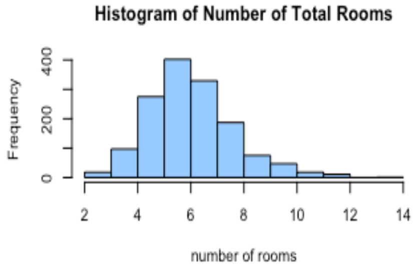
The distribution of the basement area is very similar to the distribution of the size of the first floor. The distribution is left-skewed, and the size of the basement area is clustered between 0 to 1000 square feet. The open porch area is much smaller and most of them are less than 50 square feet. As we can see from the distribution of the size of the wood deck area, the size barely exceeds 400 square feet. The distribution of the number of total rooms is more like a normal distribution. The mean of the number is around 6, which means in average, houses in Ames have about 6 rooms.


Histogram of Unfinished Basement Area Size

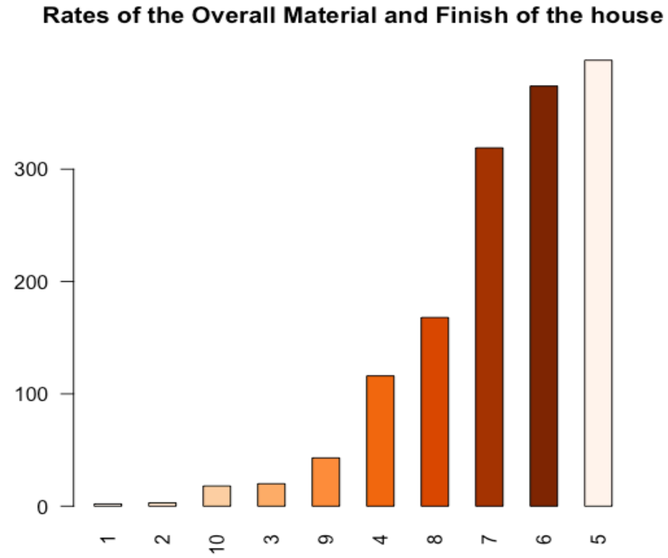

Histogram of Basement Area Size

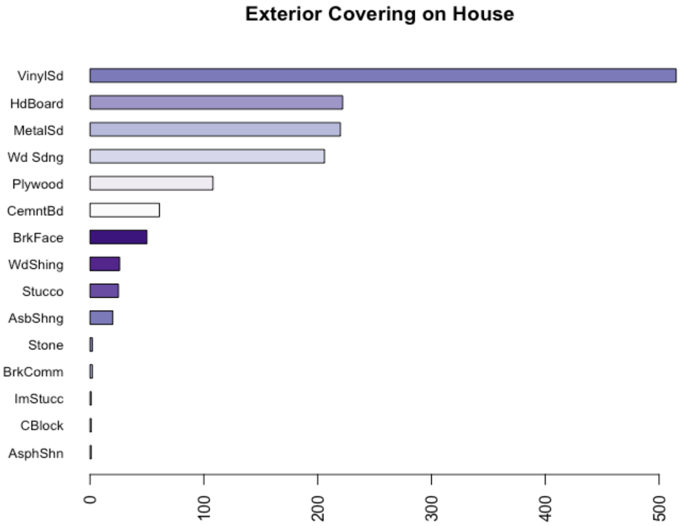

Histogram of Open Porch Area Size


Histogram of Wood deck Area Size

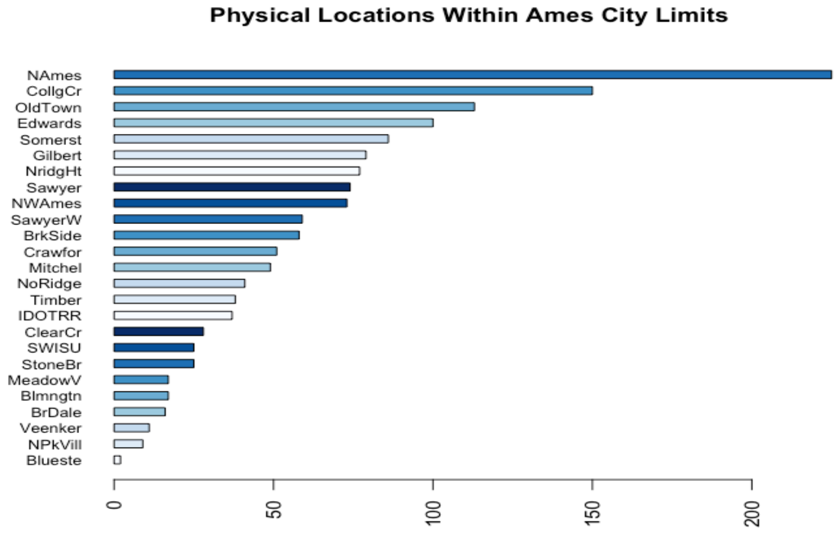

Histogram of Number of Total Rooms

There are also three categorical features in the selected model: rates of the overall material and finish of the house, exterior covering on house, and physical locations of the neighborhood. The top three rates of the overall conditions of the property were 5, 6, and 7 out of 8. Only a very small fraction of the houses was rated 1 and 2, which means fairly unsatisfying. The most famous material covering the exterior of a house is Vinyl Siding, followed by Hard Board and Metal Siding. The five least used material on the exterior of the houses in Ames are Asbestos Shingles, Cinder Block, Imitation Stucco, Cinder Block and Stone.





Among the 1460 recorded houses in Ames City, over 220 houses were located in North Ames. College Creek ranked the second followed by Old Town and Edwards. They all have over 100 houses located. The three locations that had the least number of houses located are Bluestem, Northpark Villa and Veenker. There are less than 20 houses recorded in these three areas. With the limit of the number of houses recorded in these areas, our model may not perform well when predicting the prices in these areas.



6. Split Data

The data need to be separated into two sets, one set is used for training and the other is used for testing. I did the separation with the help of train_test_split function which is provided by sklearn package. In total, we have 1460 pieces of data, I used 1/4 of it as testing data and 3/4 of it as training data.

7. Develop Model

7.1 Neural Networks

I built a neural network model using the python tensorflow library. The initial plan is to build a network with one hidden layer that contains 60 nodes. The number of hidden layers and the number of nodes in each hidden layers can be adjusted to get the best training result. I created placeholder tensors for input data and variable tensors for each weight and bias. All variable tensors are initialized with a normal distribution. Then I connect the input layer to the hidden layer and then connect the hidden layer to output layers. The activation function of each layer is set to “relu” function.

At the output layer, the mean square error is computed as the loss and then fed into AdamOptimizer. Then I run the optimizer to minimize the loss iteratively until the change of loss is smaller than a threshold. In each cycle, I choose to use the mini batch method to put data into the network. All training data is shuffled and divided into batches of size 30. And all these mini batches are used to train the network in series. Since the mean square error loss is the square of the average difference between our prediction and actual selling price, I took the square root of loss that represents the actual difference, which is more meaningful. In each cycle, I printed the in-sample loss, square root of in sample loss, out sample loss, square root of out sample loss.

7.2 Training Result of Neural Network

The initial structure of neural network I built has 1 hidden layer and the layer contain 60 nodes. In order to get the smallest in sample and out sample loss, I had to try different structure. Here are the what I tried and the loss I get:

# of hidden layers	# of nodes in each hidden layers	square root of in sample loss	square root of out sample loss
0(equal to linear regression)	NA	39092.54	54536.83

1	[60]	33447.76	47727.521
1	[40]	33144.79	47047.36
1	[100]	33348.82	47053.68
2	[100,100]	197161.72	198843.76

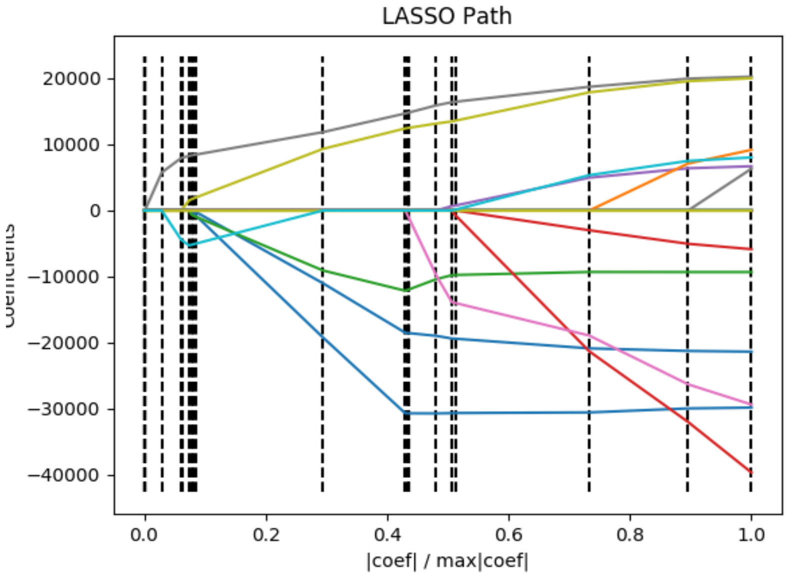
When I set the number of hidden layers to 0, the algorithm of neural network would be linear regression. Yet the result is not satisfying. The in-sample loss in around $40000, which accounts for around 22% of the average sale price. I can't reduce the loss to an acceptable level by changing the number of layer and number of nodes in each layer. The reason of large loss can due to bad choice of layers and nodes, but it takes too much time to find the perfect structure. Also, I only have 1460 data and only 3/4 of them are used for training, lack of training data could also be the reason. In conclusion, the neural network method is not a good choice for solving the problem.

7.3 LASSO

The second model I used is LASSO, which stands for Least Absolute Shrinkage and Selection Operator. Instead of estimate coefficient of regression using the Ordinary Least Square (OLS), it adds a penalty term when computing the residual sum of squares. This regularization would force the coefficients to be 0 and thus solve the high variance problem of OLS.

I performed the LASSO Regression using the python sklearn library. The function is Lasso(). In this function, one parameter is the tuning parameter lambda. I set a wide range of lambda to check how the function converges and how the coefficients were forced to be exact 0 when alpha changed. I also normalized the data to ensure they are on the same scale. I set the maximum iteration number to be 100000. Then I fitted the Lasso Regression on the training data and training target variable. Then I applied the trained model on the test data to check the accuracy of the model. I compared the predicted target with the actual target in the testing dataset and computed the square root of the mean square error loss.

The graph below shows how the LASSO worked in my model and how the coefficients changed along the path. The LASSO regression supposed to turn the coefficients of each variable in the model to exact 0 gradually. In the plot, each color represents how the coefficients of different variables changes with the tuning parameter. The coefficient of each variable was forced to 0 at a different speed. At the end of the path, we can see that the coefficients were all turned to 0.



7.4 Result of LASSO

The loss in LASSO Regression model was around $-750. Given the average sale price of the houses in Ames was around $180000, the loss rate accounts for 0.5% of the average price. This model performs really good and has greatly significantly improved the accuracy of the prediction compared with the neural network model.

Imagine you buying a house and the intrinsic value of the house you are willing to buy is $200000. The real estate agent told you the price should be $250000 then you can tell that he is taking advantage of you. With the help of machine learning techniques, the price of the house would be relative apparent cause you can easily estimate the value with a error at only 0.5% of the total price.

7.5 Random Forest

The third model I used was Random Forest. Decision Trees is a commonly used method in Machine Learning field for both classification and regression problem. While sometimes Decision Tree may face with the overfitting problem and have high variance especially when using too many variables and when the tree goes too deep. Random Forest solve this problem by using a collection of single trees. Random Forest trains on different set of training data and uses only a fraction of randomly chosen features in the model every time. Random Forest method works on both classification problem and regression problem. When dealing with classification problem, it will output the predicted classes. While for the regression problem, random forest take mean average of the prediction of every single trees.

I conducted the Random Forest model using the RandomForestRegressor() function in the python sklearn library. I set n, the number of trees in the forest to be 1000 in my model. Then I applied the function on the training data and training labels. After fitting the model, I applied the model on the testing data to check the accuracy of the model by comparing the predicted value with the actual target value in the testing dataset and computed the square root of the mean square error loss.

7.6 Result of Random Forest

As every time the splitting dataset process was random, the results may be slightly different. In order to improve the accuracy of the result, I did the same process 10 times and take the average of the results. In Random Forest model, the average loss was around $1425. The average loss rate was 0.7%(1425/180921). Although the accuracy rate was slightly higher than that in LASSO Regression model, it still performs really good.

8. Future Work

In this project, I only researched on the housing price of Ames city in Lowa, California from 2006 to 2010. The dataset actually is not very up-to-date and the city I chose is a small city. In the future, I would like to explore other Machine Learning models and see if I could further improve the accuracy of prediction. Moreover, a more up-to-date dataset could have more practical value, or I would just simply apply the model in a city that have different city scale.

9. Conclusion

After comparing the performance of the three models: Neural Networks, LASSO and Random Forest, we can see that LASSO performs the best with a loss at only 0.5% of the total price.
As shown in this project, different methods in Machine Learning may have very different performance on the same set of data. The predictive power of the methods depends on the number of training data, the type of data and so on. While by trying and changing parameter, we could always find a satisfying model with strong predictive power.

10. Reference

Barone, A. (2019, May 07). Comparative Market Analysis. Retrieved from https://www.investopedia.com/terms/c/comparative-market-analysis.asp

Boston Home Prices Prediction and Evaluation. (n.d.). Retrieved from https://www.ritchieng.com/machine-learning-project-boston-home-prices/

Cock, D. D. (2011). Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. Journal of Statistics Education, 19(3). doi:10.1080/10691898.2011.11889627

Deng, H., & Deng, H. (2018, October 28). Why random forests outperform decision trees. Retrieved from https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3-42. doi:10.1007/s10994-006-6226-1

Liberman, N., & Liberman, N. (2017, January 27). Decision Trees and Random Forests. Retrieved from https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991

Predicting House Prices with Machine Learning Algorithms. (n.d.). Retrieved from https://nycdatascience.com/blog/student-works/housing-price-prediction-using-advanced-regression-analysis/

Speakman, M. (2019, April 04). $33.3 Trillion Housing Market Up 49% Since 2012 – A Third of the Gain From California. Retrieved from https://www.zillow.com/research/california-leads-housing-gains-22600/