LUT
Lappeenranta
University of Technology

LUT School of Business and Management

Building Models for Housing Price Prediction with Machine Learning Methods

Quantitative Data Analysis

Nguyen Tran
April 2021

Table of Contents

1.Introduction *3*

2. Research Background *3*

3. Data and Methodology *4*

3.1 Data Understanding 4

3.2 Methodology 7

4. Result *8*

4.1. Principal Component Regression 8

4.2. Linear Regression 11

4.3. Decision Tree Regression 11

4.4. Supporting Vector Machine 12

4.5. Partial Least Squares Regression 13

5. Conclusion *14*

References *15*

Appendix I *16*

Appendix II – Code *19*

1.Introduction

The housing price has a significant impact one economy, it also creates concerns for the buyers, dealers, and the property owners. The growth in real estate market also increases the needs of housing price estimation. Building a housing price estimation model can help the property companies to calculate the housing price and enhancing the accuracy of the housing estimation policies in the future. There are many different factors that can affect the price of a house. Changchun and Hui (2018) divided the key factors that influence the housing prices into following categories: location, transportation, housing conditions and supporting facilities. In fact, machine learning has been used in the recent years to analyse, visualise, and predict the price of a property. However, it is necessary to compare the accuracy of the machine learning models to increase the efficiency of the prediction model.

In this paper, our focuses are investigating the elements that affecting the housing price and applying different regression models to predict the housing price. This project contains the Linear Regression model, Principal Components Regression, Support Vector Machine (SVM), Decision Tree and Partial Least Squares Regression. The objective of the report is comparing the performance of these machine learning algorithms and developing more accurate housing price prediction model.

2. Research Background

The housing price estimation is described as common topics in data science. First, in order to have the overview on the applying machine learning on housing price estimation, we conducted some basic literature research. There are many different methods and models are constructed on this topic. Lishun Yuan (2019) conducted a multiple linear regression model to estimate the factors that affect the house sale price in Los Angeles County on 140 properties. He applied linear regression model to the dependent variable housing price variable and the predictor independent variables, these independent variables are not highly correlated to each other. This is considered as simple but powerful method to forecast the future value. However, Changchun and Hui (2018) thought that the nonlinear regression also contains significant impact on housing price. They compared the linear regression and random forest to improve the model accuracy.

Besides, the other type of machine learning method Support Vector Machine (SVM) is used by Wang, Wen, Zhang, and Wang (2013) to build real estate forecasting price with the model. They chose SVM method as it is considered as a structured risk minimization principle and received more forecasting accuracy model. The research from Shinde and Gawande (2108) used the Support Vector Machine, Logistic Regression, Lasso Regression and Decision Tree techniques to build forecasting model. The result shows that the Decision Tree has the most accuracy between these methods. On the other hand, a research which was conducted by Mu, Wu, Zhang (2014) applied Support Vector Machine (SVM), Least Squares Support Vector Machine (LSSVM) and Partial Least Squares (PLS) to predict the house value. As the data used in this research is nonlinear data, the SVM and LSSVM receive more accuracy result than PLS. Besides, the SVM is the most accuracy among 3 methods.

In this report, we selected 5 different machine learning methods which are Linear Regression, Principal Components Regression, Support Vector Machine (SVM), Decision Tree Regression and Partial Least Squares (PLS) Regression. We applied all the methods that can perform better both linear and nonlinear system to investigate the data characteristic and compare the performance of these models. Based on the literature research, we realized that SVM is better to deal with nonlinear system while other selected methods more suitable with linear system. (Mu, Wu, Zhang, 2014)

3. Data and Methodology

3.1 Data Understanding

The dataset, which is used in our analysis, contains the historic data of house sales in King County, Washington State also covers Seattle, USA between May 2014 to May 2015. The dataset and was obtained from Kaggle and was published from CCO: Public Domain. It consists of 21613 observations and 20 variables. The dataset is considered as complete and high-quality dataset, with no missing values and high usability rating on Kaggle. The data contains 14 discreet variables that represent the house description, the other 4 categorical variables are used for the houses' view, conditions, and grade. The final variable is the year that house is built.

First step we decided to remove the irrelevant variables. Out of 20 variables, the id and date observations are removed and excluded out of the research because we considered these variables as irrelevant. We removed the date of sale the house due to the purpose of the report is investigating internal elements from properties that affects the decision of buying the house. Next, we conducted the data exploration analysis includes data summaries and the descriptive statistics. It is necessary to check again whether there is missing value, and this data does not have missing value. However, we based on measures of central tendency includes mean, median and mode from descriptive statistic, we realized that some variables have the outliers. Therefore, the main purposes in the exploratory analysis part include exanimate the distribution and confirm the exist of outlier in every variable. The histogram and pie chart are used for different types of variables to understand the data, and the boxplot is conducted with all continuous variables to finalize the existing of outliers.

Understand from plotting the data, we can see that the price and bedroom variable normally distributed, right skew, has outliers. The price, bedroom and bathroom variables also have significant numbers of outliers, the most common number of bathrooms is 2.5 which means that there are 2 bathrooms with the shower and 1 without shower.

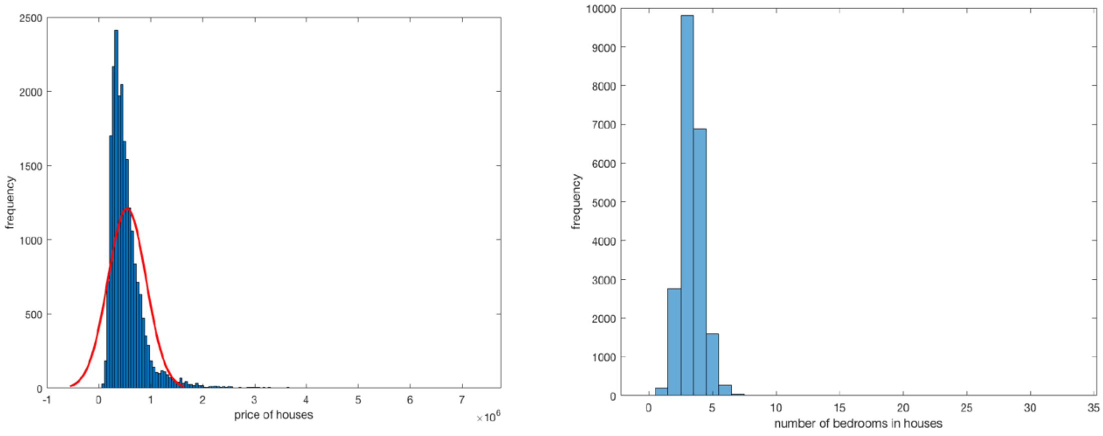


Figure 1: Histogram of price and bedroom of the house

Next, we also use the histogram and boxplot to visualize all other numerical variables and conclude that all the numerical variables in the dataset have outliers, right-skew. Except the latitude variable has the left skew distribution. The most common floor of the house in this data has 1 floor, the second common is 2 floors. Besides, the most popular value for the basement appear is 0 which means that there is no basement space in the house. The visualization of other numerical variables is attached in the appendix section.

We used the pie chart to visualize the categorical variables, that are waterfront, condition, and view. From these pie chart, we realized that 99.2% of the houses in the dataset have no overlook at waterfront, 65% of the houses have average condition, 90% of the houses have no good view.

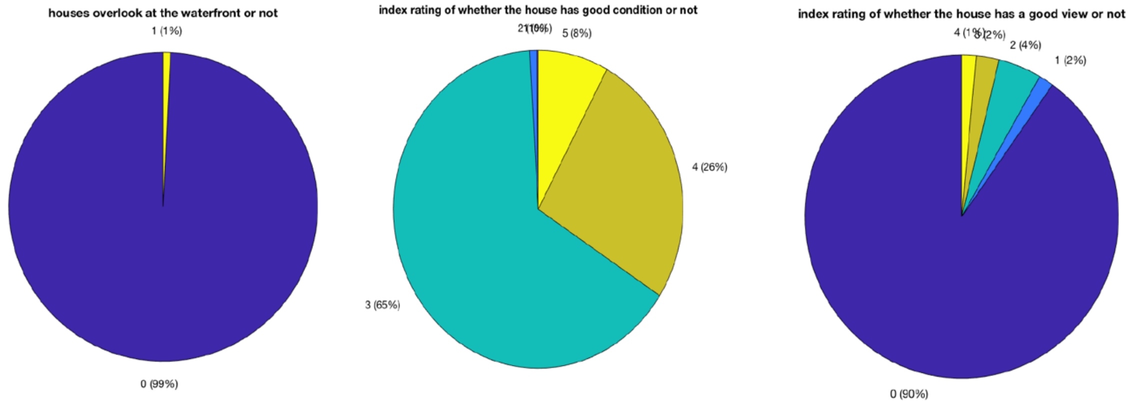


Figure 2: Pie chart for the Waterfront, condition and view of the house

Besides, we also applied the histogram to visualize the grade and year built variables. The most common zip code is 98103, grade in the data is 7 and the period from 2014 to 2017 is the time that most of the houses are built.

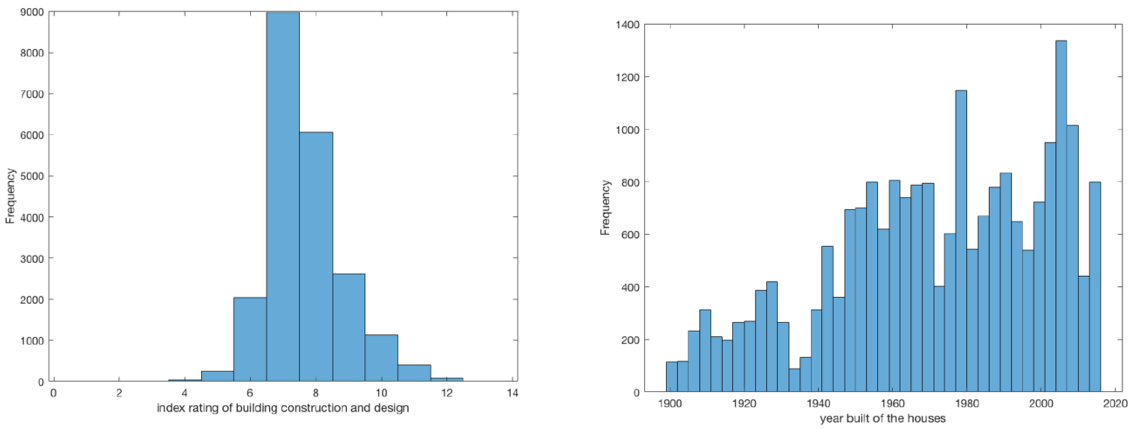


Figure 3: Histogram for the grade and year built of the house

After plotting to understand the distribution and outlier existing of the data, we decided to assign new value for the outliers by using the linear interpretation among others. We analysed the cleaned data by comparing the price in different groups of categorial variables. The housing price with the waterfront is higher than without. The 0 score for view variable group has the lower housing price compare to other groups. The 1 and 2 score groups in term the condition of the apartment has the lower price compare to others. However, the 3 score on the other hand has the slightly higher price than 1 score in term of housing conditions. We also plot the scatter plot to describe the relationship between price and year built and grade. The graph shows that houses built in recent years tends to have higher grade, however, the price spreads evenly no matter in which year the houses are built and their grades.

3.2 Methodology

Methodology is the framework of our steps to achieve the research objective. In this part, we applied many different data analysis techniques and machine learning concepts. Multiple Linear Regression uses more than one independent variable to predict the dependent one, the independent variables are not highly correlated. (Dagar & Kapoor, 2020) Principal Components Regression (PCR) applies the principal component analysis to group the correlated variables into new features that uncorrelated and perform Linear Regression with these new features as predictors. The Partial Least Squares Regression (PLS) algorithm also constructs the set of linear combinations to the features. However, different from PCR, the PLS model add the response outcome to the principal components. This helps the PLS model receive information from not only feature but also the response.

Decision Tree Regression applies the tree-alike structure for regression and classification models, it split data into smaller subset with decision nodes and leaf nodes. The advantage of this method is that it can handle both categorical and numerical data. The SVM model has the ability map the non-linear relationship into quite linear by higher dimension space called kernel trick. In this report, we are dealing with the regression problem to predict the continuous variable. We select algorithms from both linear and nonlinear regression in order to investigate the characteristic between the variables and evaluate the accuracy of the models. The linear regression techniques such as Multiple Linear Regression, PCR and PLS assume the exiting relationship that can be visualize by formulation and easy to interpret. While SVM and Decision Tree methods do not fit the model by formula and can provide more accurate prediction but hard to interpret.

Mean Square Error (MSE) is used for comparing the accuracy of these 5 models. The MSE measures the average square of error, which describes the difference between the predicted value and actual values. The objective of the research is looking for the minimum value of MSE, therefore, it is the most accuracy model for the housing price estimation.

Plotting the data with boxplot, we recognized that the dataset has 10 variables, which have the outliers. As a result, we decided to assign new value for the outliers by using the linear interpretation among others. Besides, the dataset was split into 70% data in training set and 30% data remain in testing set. Therefore, the training data is 15130 variables and testing data is 6483 variables.

First, we checked pairwise correlation between variables. From the correlation matrix, we observed that some variables have high correlation between them. We applied the PCA on the predictor training data to construct independent new variable which are linear combinations of the original variables. We applied the PCR method by using the training data to learn the model and applied the model with the testing data. The comparation from 2 to 10 principal components are conducted to find the least error and well explained majority of the variables. We also graph the predicted value and real value to compare the accuracy of the model visually. MSE is calculated and compare between different models of PCR to find the minimum value. The K-fold validation is applied to investigate the optimal numbers of principle components. The number of PC selected in PCR is the model that has the smallest MSE.

Secondly, the linear regression model is conducted to the training data and applied to testing data. We applied all the variables into the linear regression model with the first model, next we looked for the correlation matrix between the variables to optimize and remove the variables that do not have the significant statistic effect on price. Two other models are developed by removing or optimizing the variables. In the last step, all 3 models are compared visually by graph and statistically by MSE.

Thirdly, we applied the decision tree regression, SVM and PLS models for the training data to learn the model and used the testing data to evaluate the model. Graphs to compare the model and actual data are also conducted, MSE is calculated with each model to compare the accuracy of the model. Finally, the comparation between MSE all the models from different types of algorithms are conducted. The objective of this action is finding the minimum error and most accuracy prediction model for the dataset.

4. Result

4.1. Principal Component Regression

In this method, we firstly looking for the principal components of the independent variables in the independent training data by applying the principal component analysis technique on the predictor variables. From the figure below, we can see that the first 3 component variables 98,3% of variance in the independent training data.

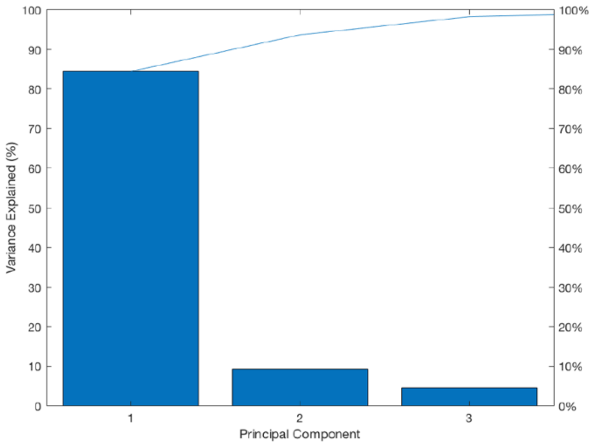


Figure 4: Principal Component on independent training house data

Next, we conducted PCR models from 2 to 10 principal components on the training data and applied the learned model to the testing data. Finally, we examine the prediction value with the actual value to find the more accuracy model for this method.

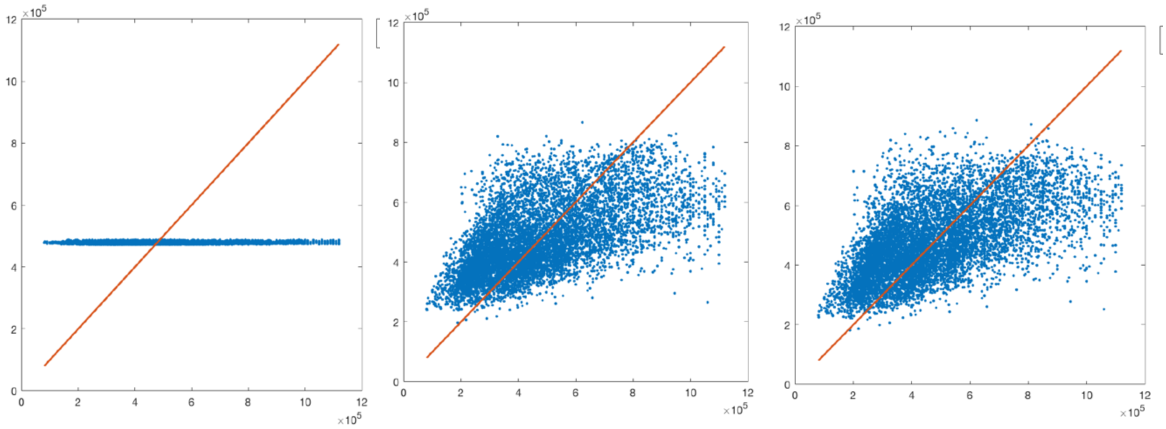


Figure 5: Comparation between prediction and actual price value in 2 PC, 3PC and 4 PC

The MSE of 4 PC has the minimum MSE value among 3 models. Besides, we also used the k-fold validation method to investigate the optimal numbers of PC. z

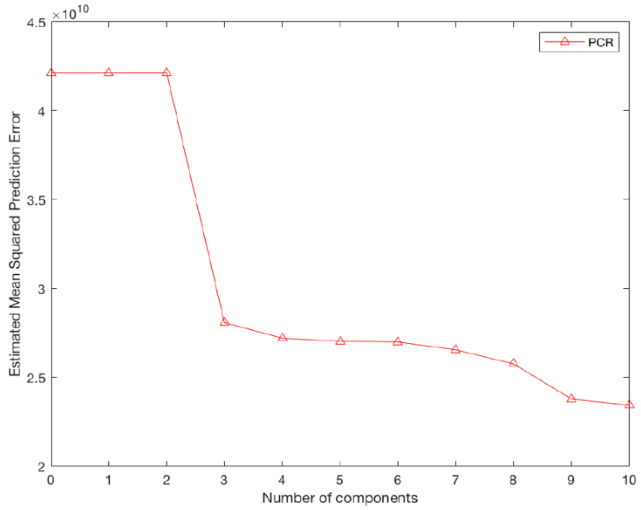


Figure 6: K-fold validation for number of components

Based on the graph, 3 PCs is the best choice because from 2PCs to 3PCs, Estimated Mean Squared Prediction Error (EMSPR) improved significantly. The greater number of PCs, the less EMSPR though the difference is little after 3 PCs. In the PCR prediction model, the cross-validation technique is applied to minimize the error during the prediction processes. This helps to avoid the overfitting problem. In the PCR processes, we conducted the K-fold validation. Different from PCA method, the PCR model has different comparison. The PCR model chooses the minimum MSE as the optimal number of principal component choice. As a result, we applied all models from 2 to 10 PCs into training data to learn and use the predictor testing data to forecast the housing price. The MSE is calculated by comparing the prediction value with the actual one. The objective is to minimise MSE and the model with 10 PCs has smallest MSE value (2.3404e+10) among all. Therefore, at this point, PCR model with 10 PCs predicts most correctly.

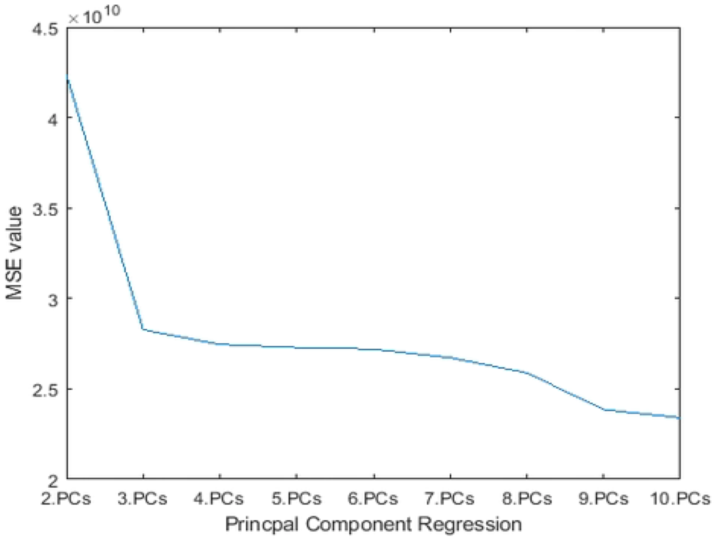


Figure 7: Comparation of MSE models from 2PCs to 10PCs

4.2. Linear Regression

We built 3 different model by the linear regression model. First all the variables are applied to the default linear regression model. Secondly, we adjusted the model by checking the correlation matrix on price variable. We removed the variables do not have effect on the price include sqft_lot and sqft_lot15, then a model is built with the remained models. Thirdly, we improved the model by adding or removing variables using 10 steps. The training independent data is used to learn the model and testing data to evaluate the accuracy of the models. The MSE is calculated and compared as the final step of this method to find the most effective model for this method.

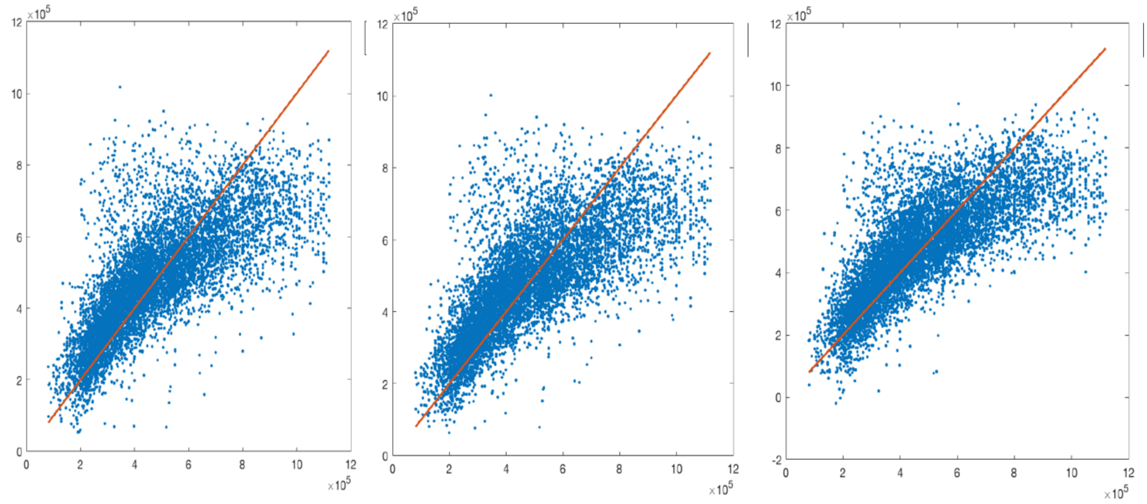


Figure 8: *Comparation between prediction and actual price value in all default linear regression, adjusted and step-adjusted model*

The step-adjusted model has the smallest MSE value 1.7542e+10 while default linear regression received the value 1.8732e+10 and adjusted model got 1.9044e+10. Therefore, the self-adjusted model correctly predicted compare to other 2 models.

4.3. Decision Tree Regression

First, we applied the training independent and dependent variables into the Regression Decision Tree model by fitting regression decision tree into the model. Next, we used the testing data of independent and dependent variables to predict the price and calculate the MSE of the decision tree. The MSE value received in this model is 1.8650e+10.

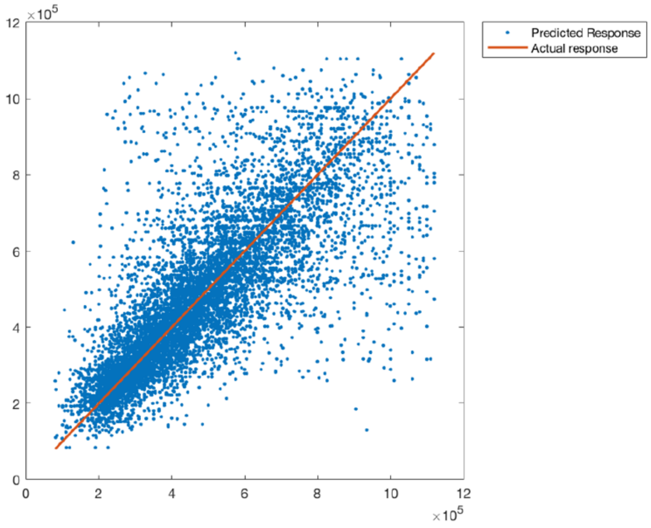


Figure 9: Comparation between prediction and actual price value in Regression Decision Tree Model

4.4. Supporting Vector Machine

The same with the Decision Tree method, with the Support Vector Machine by fitting a support vector machine regression model into the training data. It is significant to check whether there is converges or not in the model. With the housing dataset, there is a model converges. The pricing prediction is developed based on the independent testing data and MSE is calculated by comparing the prediction value and dependent value in the testing data. The MSE received in this model is 3.7268e+10.

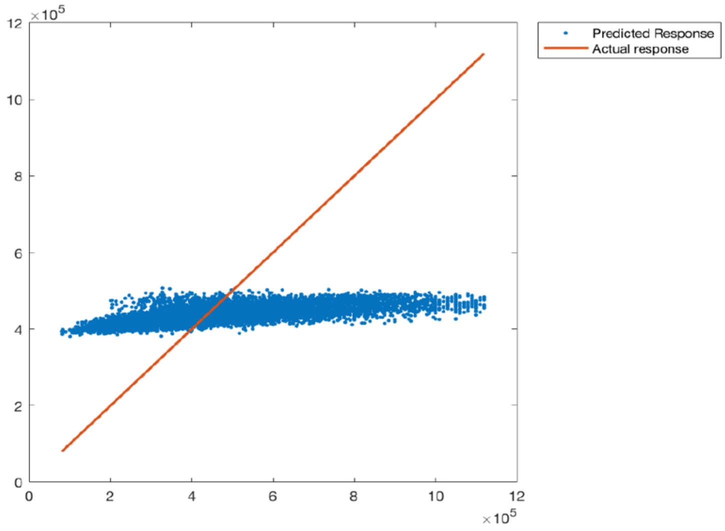


Figure 10: Comparation between prediction and actual price value in Supporting Vector Machine

4.5. Partial Least Squares Regression

In this technique, we used the data contains correlated predictor variables. This method is the combination between multiple linear regression and principal component analysis, which means that it includes the information from both predictors and responses while also consider the correlation matter. We used this technique on training data with 10 components of responses in price variable by the predictors. After learning the model, we applied it into the testing independent data to predict the new housing price and compare with the actual price. The MSE calculated by actual and estimated price is 2.2242e+10.

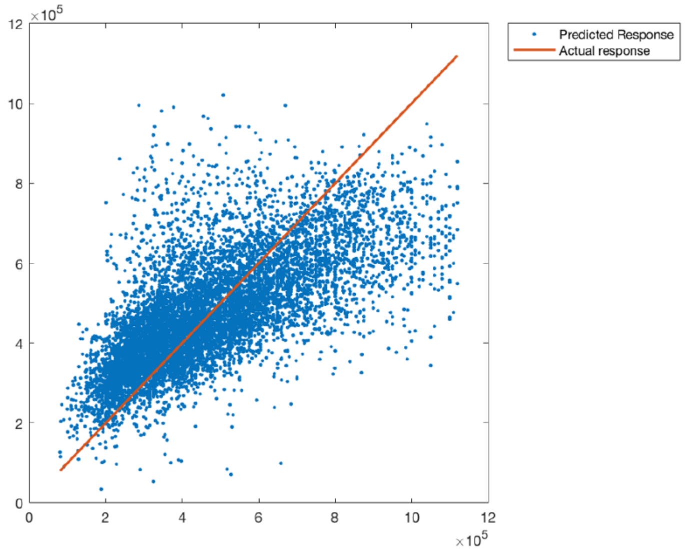


Figure 11: Comparation between prediction and actual price value in Supporting Vector Machine

Comparing the MSE values from the most accuracy model from 5 methods, we realized that in the prediction of housing data the most correctly model is the Linear Regression technique. This model is applied the step adjusted with smallest MSE value. And the SVM created the biggest MSE, due to the reason that this model is more suitable with the nonlinear system. The interesting found out is that the Linear Regression (Step Adjusted model) and Decision Tree models are the first and second most accuracy models. A further research with these 2 methods is recommended to build more accuracy prediction model in the future.

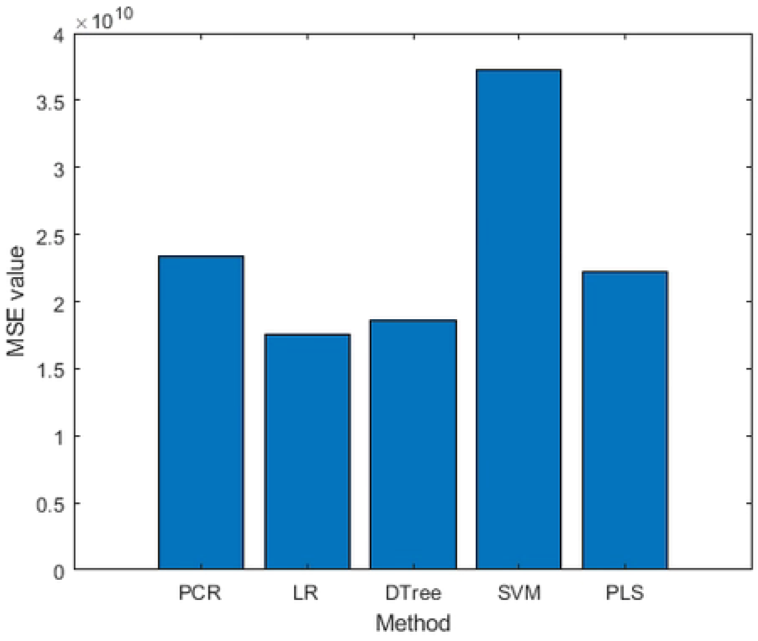


Figure 12: MSE Comparation between 5 models

5. Conclusion

In this paper, we focused on prediction task in housing price. Many machine learning techniques are applied and analysed in order to forecasting the continuous variable. The housing price estimation is supervised learning, which is the predictive model based on the input and output data. The results were obtained by applying variety of regression methods such as Linear Regression, Principal Components Regression, Support Vector Machine (SVM), Decision Tree Regression and Partial Least Squares (PLS) Regression.

After data understanding stage, we conducted data preparation by exploratory analysis and identify the existing of outliers. We assigned new value for the outliers by using the linear interpretation among others and exam the correlation between variables in the data. Next, in order to avoid overfitting, we used the cross-validation technique to split into 70% data in training set and 30% data remain in testing set. After these preparations, we started applied the regression methods to predict the housing price.

With the Linear Regression method, we run 3 different models include: default linear regression, adjusted and step-adjusted models. The step-adjusted model predicted the most accuracy result among 3 models with the smallest MSE. In the Principal Components Regression method, we first applied the Principal Component Analysis then K-fold validation to find out the optimal numbers of principle components that can describe the dataset. After that, we started to apply the PCR model to different numbers of principle components.

Different from PCA, the number of principle component selection in PCR model is the model that created the smallest MSE. We found out that the 10PCs model generated the smallest PC and will be the optimal selection for PCR model.

The SVM, Regression Decision Tree and PLS Regression techniques are applied into the training data set to develop the prediction models. The testing data is used to predict the house price and compared to the actual value. Comparing these techniques, we realized that SVM has the biggest MSE. From the linear characteristic of this dataset, this solution is not the optimal choice.

After comparing the MSE values between 5 different methods, we found out the most accuracy for the housing price prediction is Linear Regression (Step Adjusted Model) with the minimum MSE value. However, the gap between MSE value of Decision Tree Regression and Linear Regression (Self-adjusted Model) is considered as not much. As a result, we recommend a further analysis with 2 method Decision Tree Regression and Linear Regression.

References

Xibin Wang, Junhao Wen, Yihao Zhang, Yubiao Wang (2013) Real estate price forecasting based on SVM optimized by PSO. *Optik*. 125. 1439-1443

Lishun Yuan (2019) A regression model of single house price in LA contructing a predicted model for house price. [www document]. [Accessed 15 May 2021]. Available https://scholarworks.calstate.edu/downloads/dj52w646n

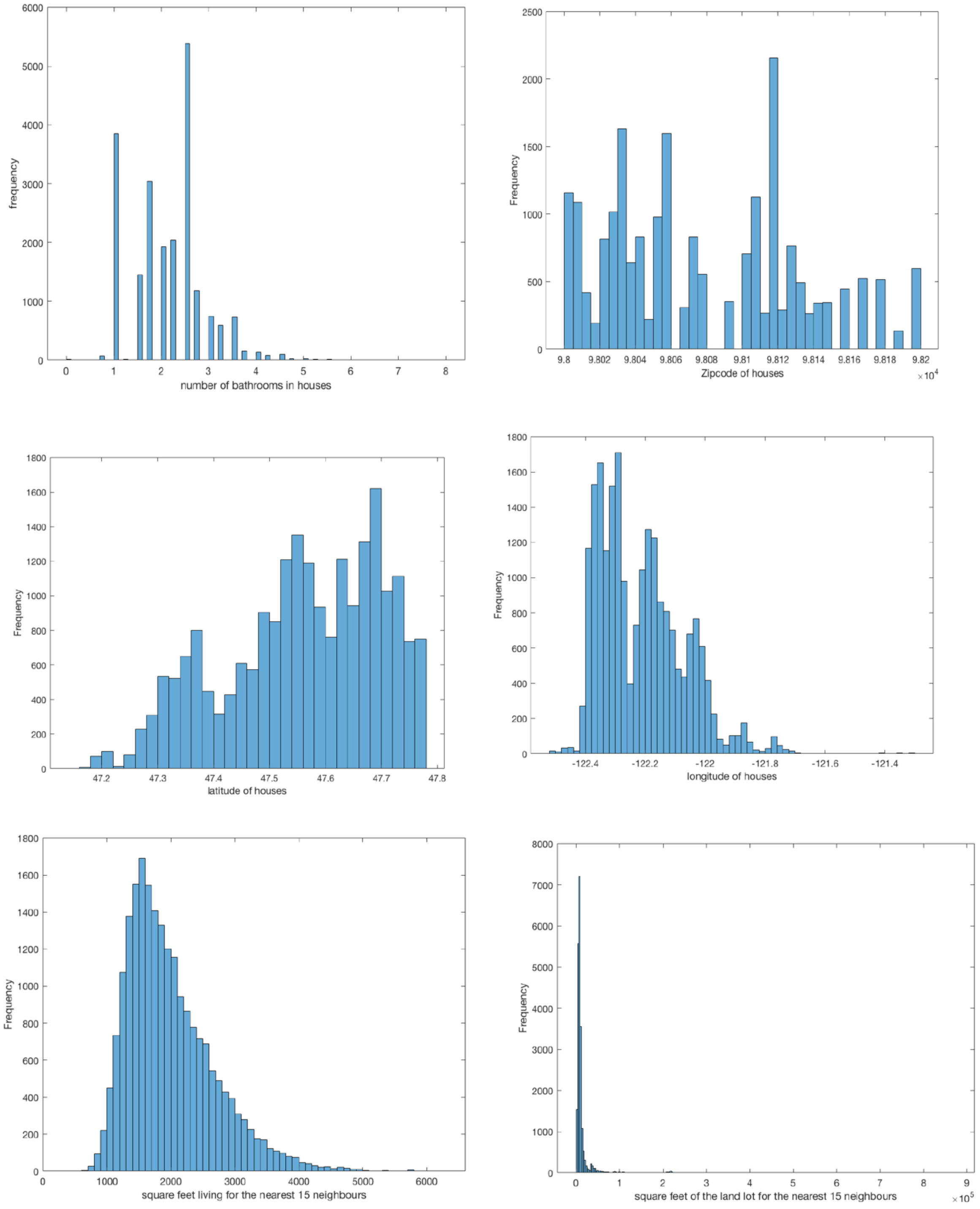
Neelam Shinde, Kiran Gawande (2018) Valuation of House Price Using Techniques, International Journal of Advances in Electronics and Computer Science Volume-5. Issue-6. June 2018

Jingyi Mu, Fang Wu and Aihua Zhang (2014) Housing Value Forecating Based on Machine Learning Methods. Volume. 2014. Article ID 648047. p 7. [www document]. [Accessed 15 May 2021]. Available https://doi.org/10.1155/2014/648047

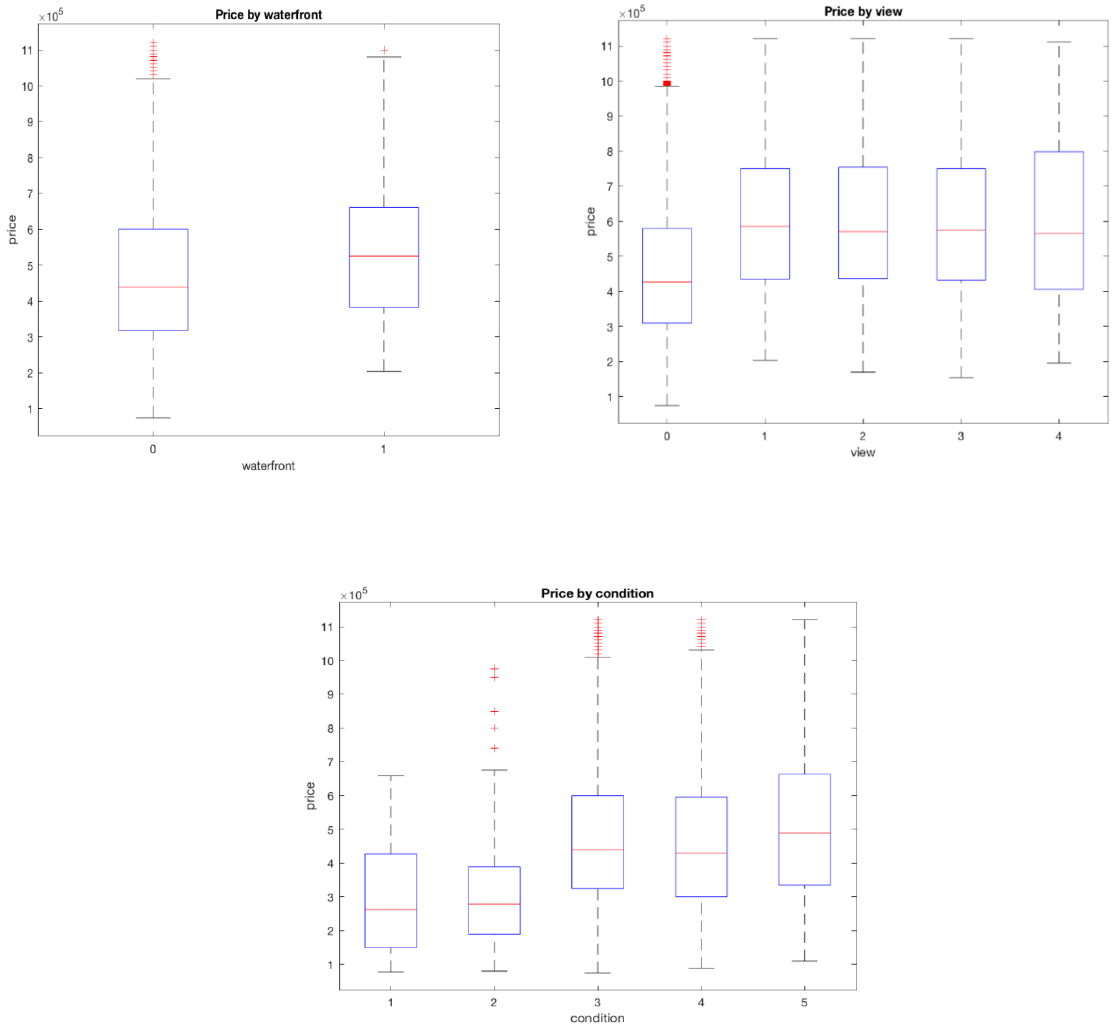
Akash Dagar and Shreya Kapoor (2020) A Comparative Study on House Price Prediction. International Journal for Modern Trends in Science and Technology. 6(12): 103-107

Appendix I

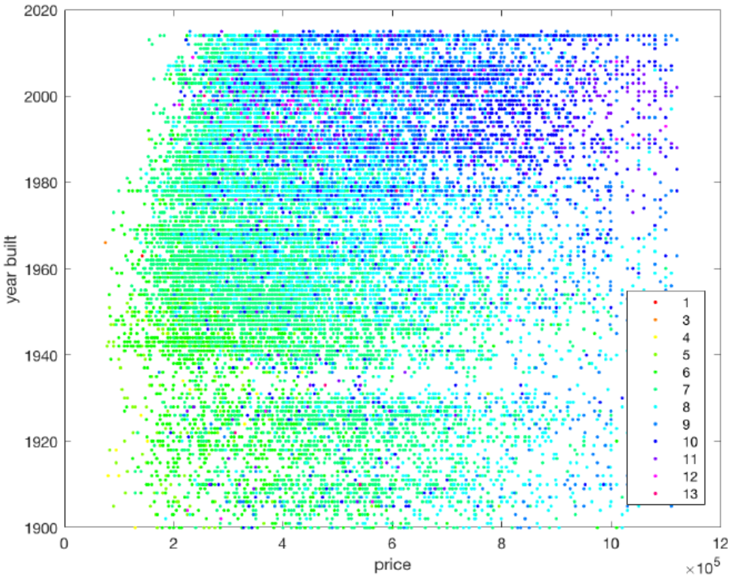
Appendix 1.1 Distribution of number of bedrooms, zip code, latitude, longitude, square feet living and land for the nearest 15 neighbours variables



Appendex 1.2 Boxplot between price and categorical variables (Waterfront, view, condition)



Appendex 1.3 Scatter plot of housing price group by grade and by year



Appendix II – Code

Input data

```
d=readtable('project_housing_data.csv')
```

Summarise the table

```
summary(d)
```

As we can see, there is no missing value in any attribute.

DATA VISUALISATION

Visualise each attribute to better understand data.

1. **Price**: Use histogram to illustrate it.

```
histogram(d.price)
    xlabel('price of houses');
    ylabel('frequency');
```

Check if it fits the normal distribution

```
histfit(d.price)
    xlabel('price of houses');
    ylabel('frequency');
```

The distribution of price seems to be right-skewed. Let's check if it has many outliers

```
boxplot(d.price)
xlabel('price of houses');
```

Based on the boxplot, we can see this attribute has many outliers. It has total 1159 outliers. We can also see its min-max values and its quartiles. Let's see its mean and mode value.

```
mean(d.price)
mode(d.price)
```

The mean price is 540 000 and the mode value is 350 000. After the outlier cleaning, the mean may change.

2. Bedrooms:

```
histogram(d.bedrooms)
    xlabel('number of bedrooms in houses');
    ylabel('frequency');
```

According to the histogram, the graph is a right-skewed distribution. Let's check the outliers with the boxplot.

```
boxplot(d.bedrooms)
    xlabel('number of bedrooms in houses')
```

Based on the boxplot, there is 1 outlier with the value 33. This doesn't make sense, we will clean the outlier data when we finish analyzing all variables.

```
mean(d.bedrooms)
mode(d.bedrooms)
```

The mean number of bedrooms in houses is 3.37 and the mode value is 3.

3. Bathrooms

```
histogram(d.bathrooms)
xlabel('number of bathrooms in houses');
```

```
ylabel('frequency');
```

Let's have a look at boxplot to see it clearer.

```
boxplot(d.bathrooms)
xlabel('number of bathrooms in houses')
```

There are outliers below and above the quantile. We will remove it.

```
mean(d.bathrooms)
mode(d.bathrooms)
```

The mean number of bathrooms in houses is 2.11 and the mode value is 2.5.

4. Square feet living (sqft_living)

```
histogram(d.sqft_living)
    xlabel('square feet living of houses')
    ylabel('Frequency')
```

The boxplot as well.

```
boxplot(d.sqft_living)
xlabel('square feet living of houses')
```

And also check out the mean and mode value

```
[mean(d.sqft_living) mode(d.sqft_living)]
```

The mean square feet living is 2 079 and its mode value is 1 300.

5. Square feet lot (sqft_lot):

Histogram:

```
histogram(d.sqft_lot)
xlabel('square feet lot of houses')
ylabel('Frequency')
```

Boxplot:

```
boxplot(d.sqft_lot)
xlabel('square feet lot of houses')
```

Mean and mode value

```
[mean(d.sqft_lot) mode(d.sqft_lot)]
```

The mean square feet of the land lot is 15 107 and the mode value is 5000.

6. Floors:

```
histogram(d.floors)
xlabel('number of floors of houses')
ylabel('Frequency')
```

The graph looks good. Most of the houses has 1 floor, the second most number of floors is 2.

```
boxplot(d.floors)
xlabel('number of floors of houses')
```

This attribute doesn't have any outlier.

So, in general, price, square feet livng and square feet lot variable have upper outliers, bedrooms and bathrooms have outliers above and below the quantile while floors have no outlier.

Mean and mode value:

```
[mean(d.floors) mode(d.floors)]
```

The mean number of floors in houses is 1.49 and the mode value is 1.

7. Square feet above (sqft_above)

Histogram:

```
histogram(d.sqft_above)
    xlabel('square feet above of houses')
    ylabel('Frequency')
```

Boxplot:

```
boxplot(d.sqft_above)
    xlabel('square feet above of houses')
```

Mean and mode value

```
    [mean(d.sqft_above) mode(d.sqft_above)]
```

8. Square feet basement (sqft_basement)

Histogram:

```
histogram(d.sqft_basement)
    xlabel('square feet basement of houses')
    ylabel('Frequency')
```

Boxplot:

```
boxplot(d.sqft_basement)
    xlabel('square feet basement of houses')
```

Mean and mode value

```
[mean(d.sqft_basement) mode(d.sqft_basement)]
```

9. Zipcode:

```
histogram(d.zipcode)
    xlabel('Zipcode of houses')
    ylabel('Frequency')
```

As it is a categorical variable, let's see the summary of it.

```
summary(categorical(d.zipcode))
mode(d.zipcode)
```

The mode zipcode appears in the data set is 98103.

10. Lat:

```
histogram(d.lat)
    xlabel('latitude of houses')
    ylabel('Frequency')
```

11. Long:

```
histogram(d.long)
    xlabel('longitude of houses')
    ylabel('Frequency')
```

12. Square feet living for the nearest 15 neighbours (sqft_living15)

Histogram:

```
histogram(d.sqft_living15)
    xlabel('square feet living for the nearest 15 neighbours')
    ylabel('Frequency')
```

Boxplot:

```
boxplot(d.sqft_living15)
    xlabel('square feet living for the nearest 15 neighbours')
```

It seems to have quite a lot outliers in the upper part of the quantile.

Mean and mode value:

```
[mean(d.sqft_living15) mode(d.sqft_living15)]
```

13. Square feet lot for the nearest 15 neighbours (sqft_lot15)

Histogram:

```
histogram(d.sqft_lot15)
    xlabel('square feet of the land lot for the nearest 15 neighbours')
    ylabel('Frequency')
```

Boxplot:

```
boxplot(d.sqft_lot15)
    xlabel('square feet of the land lot for the nearest 15 neighbours')
```

Mean and mode value:

```
[mean(d.sqft_lot15) mode(d.sqft_lot15)]
```

14. Waterfront:

```
summary(categorical(d.waterfront))
```

We use pie chart to illustrate the data:

```
pie(categorical(d.waterfront));
title('houses overlook at the waterfront or not')
```

Most of the house doesn't overlook the waterfront with 21450 houses over total, accounts for 99.2%. Only 1% of houses does.

15. View:

```
summary(categorical(d.view))
```

Let's have a look at the pie chart:

```
pie(categorical(d.view))
    title('index rating of whether the house has a good view or not')
```

19489 houses have no good view over total, accounts for 90%.

16. Condition:

```
summary(categorical(d.condition))
```

Let's have a look at the pie chart:

```
pie(categorical(d.condition))
    title('index rating of whether the house has good condition or
not')
```

Based on the pie chart, 65% of houses have average condition. Better than average conditions have higher proportion than the lower part.

17. Grade:

```
summary(categorical(d.grade))
```

Let's check the histogram:

```
histogram(d.grade)
    xlabel('index rating of building construction and design')
    ylabel('Frequency')
```

Grade 7 is the median and the mode of this attribute.

18. Year built (yr_built)

```
summary(categorical(d.yr_built))
```

Histogram:

```
histogram(d.yr_built)
    xlabel('year built of the houses')
    ylabel('Frequency')
```

According to the histogram, the period 2004-2007 has the most hosues built.

CLEANING OUTLIERS

```
d.id=[]; d.date=[]; a=d;
for i = [1:5 7 8 11:13]
    a.(i)= filloutliers(d.(i),'linear','quartiles');
end
```

Now d is the original table and a is the cleaned table.

DATA GROUPING AND ANALYZING

First, set the function:

```
mystats=@(x)[min(x) mean(x) max(x)];
```

Now compare price in different groups of categorical variables:

1. Waterfront:

```
[grpW,WVals] = findgroups(a.waterfront);
statspricewaterfront=splitapply(mystats,a.price,grpW)
bar(statspricewaterfront)
    xticklabels(WVals)
```

We can see that the mean price of group 0 (house do not overlook waterfront) is lower than of group 1. So, with the waterfront, the price is higher than without.

```
boxplot(a.price,a.waterfront)
title('Price by waterfront')
    xlabel('waterfront')
    ylabel('price')
```

Same conclusion as above, the overal price of group 1 is higher than group 0.

2. View

```
[grpV,VVals] = findgroups(a.view);
statspriceview=splitapply(mystats,a.price,grpV)
bar(statspriceview)
    xticklabels(VVals)
```

The histogram shows us clearly the overal price of each rating in view.

```
boxplot(a.price,a.view)
title('Price by view')
    xlabel('view')
    ylabel('price')
```

3. Condition

```
[grpC,CVals] = findgroups(a.condition);
statspricecondition=splitapply(mystats,a.price,grpC)
```

```
bar(statspricecondition)
    xticklabels(CVals)
boxplot(a.price,a.condition)
title('Price by condition')
    xlabel('condition')
    ylabel('price')
```

4. Grade

```
[grpC,CVals] = findgroups(a.condition);
statspricecondition=splitapply(mystats,a.price,grpC)
bar(statspricecondition)
    xticklabels(CVals)
boxplot(a.price,a.condition)
title('Price by condition')
    xlabel('condition')
    ylabel('price')
```

Next, we analyze 3 variables. A scatter plot will be created to describe the relationship between price, year built and grade.

```
gscatter(a.price,a.yr_built,categorical(a.grade))
    xlabel('price')
    ylabel('year built')
```

The graph shows us that houses built in recent years tends to have higher grade, while the price spreads evenly no matter in which year the houses are built and their grades.

Cross Validation

We separate the data with 70% is training data and 30% is test data. We set matrix x as predictors and y as target

```
data = table2array(a);
y=data(:,1); x=data; x(:,1)=[];

rng('default')
cv = cvpartition(length(data),'holdout',0.30)
xtrain = x(training(cv),:);
ytrain = y(training(cv),:);
xtest = x(test(cv),:);
ytest = y(test(cv),:);
```

Principal component regression

Principal component analysis of the xtrain.

```
[coeff,scoreTrain,~,~,explained,mu] = pca(xtrain)
```

Use graph to illustrate the explained

```
figure()
pareto(explained)
xlabel('Principal Component')
ylabel('Variance Explained (%)')
```

Based on the pareto graph, we see that the first 3 components explain 98.3% of variance in Xtrain.

Choose the optimal number of pcs by k-fold validation method

```
[n,p] = size(x);
PCRmsep = sum(crossval(@pcrsse,x,y,'KFold',7),1) / n
```

```
plot(0:10,PCRmsep,'r-^');
xlabel('Number of components');
ylabel('Estimated Mean Squared Prediction Error');
legend('PCR');
```

Based on the graph, 3 pcs is the best choice because from 2PCs to 3PCS, Estimated Mean Squared Prediction Error (EMSPR) improved significantly. The more number of PCs, the less EMSPR though the difference is little after 3 PCs. Now, to make a complete anlysis, we make models with all possible components (4 components, 5 components, 6 components,...10 components), and choose the model (number of components) that minimizes MSE in test data.

PCR with 2 PCs:

```
scoreTrain2comp = scoreTrain(:,1:2);
mdl_2pcs = fitlm(scoreTrain2comp,ytrain)
scoreTest2comp = (xtest-mu)*coeff(:,1:2);
YTest_predicted_2PCs = predict(mdl_2pcs,scoreTest2comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_2PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PCR with 3 PCs:

```
scoreTrain3comp = scoreTrain(:,1:3);
mdl_3pcs = fitlm(scoreTrain3comp,ytrain)
scoreTest3comp = (xtest-mu)*coeff(:,1:3);
YTest_predicted_3PCs = predict(mdl_3pcs,scoreTest3comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_3PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PCR with 4 PCs:

```
scoreTrain4comp = scoreTrain(:,1:4);
mdl_4pcs = fitlm(scoreTrain4comp,ytrain)
scoreTest4comp = (xtest-mu)*coeff(:,1:4);
YTest_predicted_4PCs = predict(mdl_4pcs,scoreTest4comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_4PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PCR with 5 PCs:

```
scoreTrain5comp = scoreTrain(:,1:5);
mdl_5pcs = fitlm(scoreTrain5comp,ytrain)
scoreTest5comp = (xtest-mu)*coeff(:,1:5);
YTest_predicted_5PCs = predict(mdl_5pcs,scoreTest5comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_5PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PCR with 6 PCs:

```
scoreTrain6comp = scoreTrain(:,1:6);
mdl_6pcs = fitlm(scoreTrain6comp,ytrain)
scoreTest6comp = (xtest-mu)*coeff(:,1:6);
YTest_predicted_6PCs = predict(mdl_6pcs,scoreTest6comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_6PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PCR with 7 PCs:

```
scoreTrain7comp = scoreTrain(:,1:7);
mdl_7pcs = fitlm(scoreTrain7comp,ytrain)
scoreTest7comp = (xtest-mu)*coeff(:,1:7);
YTest_predicted_7PCs = predict(mdl_7pcs,scoreTest7comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_7PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PCR with 8 PCs:

```
scoreTrain8comp = scoreTrain(:,1:8);
mdl_8pcs = fitlm(scoreTrain8comp,ytrain)
scoreTest8comp = (xtest-mu)*coeff(:,1:8);
YTest_predicted_8PCs = predict(mdl_8pcs,scoreTest8comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_8PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PCR with 9 PCs:

```
scoreTrain9comp = scoreTrain(:,1:9);
mdl_9pcs = fitlm(scoreTrain9comp,ytrain)
scoreTest9comp = (xtest-mu)*coeff(:,1:9);
YTest_predicted_9PCs = predict(mdl_9pcs,scoreTest9comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_9PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PCR with 10 PCs:

```
scoreTrain10comp = scoreTrain(:,1:10);
mdl_10pcs = fitlm(scoreTrain10comp,ytrain)
scoreTest10comp = (xtest-mu)*coeff(:,1:10);
YTest_predicted_10PCs = predict(mdl_10pcs,scoreTest10comp);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,YTest_predicted_10PCs,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

Evaluate which models predict most correctly.

```
MSEpcr2comp = immse(ytest,YTest_predicted_2PCs)
MSEpcr3comp = immse(ytest,YTest_predicted_3PCs)
MSEpcr4comp = immse(ytest,YTest_predicted_4PCs)
MSEpcr5comp = immse(ytest,YTest_predicted_5PCs)
MSEpcr6comp = immse(ytest,YTest_predicted_6PCs)
MSEpcr7comp = immse(ytest,YTest_predicted_7PCs)
MSEpcr8comp = immse(ytest,YTest_predicted_8PCs)
MSEpcr9comp = immse(ytest,YTest_predicted_9PCs)
MSEpcr10comp = immse(ytest,YTest_predicted_10PCs)
```

The objective is to minimise MSE and the model with 10pcs has smallest MSE among all. Therefore, at this point, PCR model with 10 PCs predicts most correctly.

LINEAR REGRESSION

Default linear regression model

```
mdl=fitlm(xtrain,ytrain,'linear')
Ypredicted_test= mdl.predict(xtest);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,Ypredicted_test,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

Adjusted model

Correlation of all attributes.

```
corr(data)
```

Focus on the price variable, sqft_lot and sqft_lot15 have no effect on the price.

Construct another model without those variables

```
xtrain2=xtrain; xtrain2(:,[4 12])=[];
```

```
xtest2=xtest; xtest2(:,[4 12])=[];

mdl2=fitlm(xtrain2,ytrain,'linear')
Ypredicted_test2= mdl2.predict(xtest2);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,Ypredicted_test2,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

Step-Adjusted Model

Improve the model by adding or removing variables using step.

```
mdl3 = step(mdl2,'NSteps',10)
Ypredicted_test3= mdl3.predict(xtest2);
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,Ypredicted_test3,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

Evaluate models

```
MSEtest = immse(ytest,Ypredicted_test)
RMSEtest = sqrt(MSEtest)
MSEtest2 = immse(ytest,Ypredicted_test2)
RMSEtest2 = sqrt(MSEtest2)
MSEtest3 = immse(ytest,Ypredicted_test3)
RMSEtest3 = sqrt(MSEtest3)
```

RMSEtest3 has smallest value. Therefore, step-adjusted model mdl3 predicts most correctly among 3.

REGRESSION DECISION TREE

```
mdl_rtree=fitrtree(xtrain,ytrain)
y_rtree=predict(mdl_rtree,xtest);
MSEtree=immse(ytest,y_rtree)
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,y_rtree,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

REGRESSION SVM

```
mdl_rsvm=fitrsvm(xtrain,ytrain,'Standardize',true)
mdl_rsvm.ConvergenceInfo.Converged %check whether the model converges
or not
y_rsvm=predict(mdl_rsvm,xtest);
MSEsvm=immse(ytest,y_rsvm)
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,y_rsvm,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

PLS regression

```
[xL,yl,xS,yS,BETA,PCTVAR] = plsregress(xtrain,ytrain,10)
yPLS=[ones(size(xtest,1),1) xtest]*BETA;
MSEpls=immse(ytest,yPLS)
```

Let's graph the predicted value and the real value to compare

```
figure
plot(ytest,yPLS,".")
hold on
plot(ytest,ytest,'Linewidth',2)
hold off
legend('Predicted Response','Actual response','location','bestoutside')
```

Compare MSE of all models

PCR models (from 2 to 10 PCs)

```
MSEpcr2comp
MSEpcr3comp
MSEpcr4comp
MSEpcr5comp
MSEpcr6comp
MSEpcr7comp
MSEpcr8comp
MSEpcr9comp
MSEpcr10comp
```

Linear regression models (default model, adjusted model and step-adjusted model)

```
MSEtest
MSEtest2
MSEtest3
```

Regression decision tree model

```
MSEtree
```

Support vector machine regression model

```
MSEsvm
```

Partial Least Squares (PLS) regression model

```
MSEpls
```

Among all the built models, based on MSE, MSEtest3 (step-adjusted model) has the smallest value. Therefore, this model predicts most correctly.

Barplot comparation the result of 5 models

```
bar([MSEpcr10comp MSEtest3 MSEtree MSEsvm MSEpls])
xlabel('Method')
ylabel('MSE value')
xticklabels({'PRC','LR','DTree','SVM','PLS'})
```