Analyzing Housing Price dataset through Visualization and Machine Learning technique

Halilou Junior Timite

Tangbufan Wei

Zhiyi Lin

Department of Mathematics and Statistics

STAT5703 -- Data Mining I

University of Carleton, Ottawa

Submitted to: Professor Name

April 30th , 2023

PROBLEM STATEMENT

The housing market is an essential aspect of the economy, and house prices have significant implications for homeowners, home buyers, real estate agents, and policymakers. The housing market can be complex, with numerous factors influencing prices, such as location, size, condition, age, and features of the home. Understanding these factors and how they interact is crucial for making informed decisions about buying, selling, or investing in real estate.

The Housing Price dataset is a valuable resource for exploring the factors affecting house prices. This dataset contains information about houses sold between May 2014 and May 2015 including various features of the houses and their sale prices. In this project, we aim to use this dataset to develop machine learning models to investigate various features.

To achieve this goal, we will apply a range of data science techniques to the Housing Price dataset. First, we will use visualization techniques to gain insights into the relationships between different variables and identify potential outliers or unusual patterns. We will then apply dimension reduction techniques to identify the most important factors affecting prices and reduce the dimensionality of the dataset. We will use both unsupervised and supervised learning techniques to develop models on various features.

Overall, this project aims to provide valuable insights into the factors affecting housing prices and to create tools that can be used by home buyers, sellers, and real estate professionals to make more informed decisions. By developing accurate and reliable models for predicting house prices, we hope to contribute to a better understanding of the housing market and its dynamics.

Keywords: Housing market Price , Outlier, Feature selection, Machine learning

INTRODUCTION

The dataset being used in this report contains information about house such as including price, bedrooms, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterfront, view, condition, grade, sqft_above, sqft_basement, yr_built, yr_renovated, zip code, lat, long, sqft_living15, and sqft_lot15. To better understand the dataset, a data dictionary is provided.

Section 1: Exploratory Data Analysis of this report focuses on the exploratory analysis of the dataset. This includes data cleaning, summary statistics, and visualization to gain insights into the dataset.

Section 2: Feature Selection aims to identify the most important features affecting housing prices and select a subset of these features for use in our machine learning models. We will use various techniques such as correlation-based feature selection and variance-based feature selection to identify the features that have the highest predictive power and reduce the dimensionality of the dataset. By selecting a smaller set of informative features, we aim to reduce overfitting and improve the performance of our models. We will also consider the interpretability of our models and select features that are easy to understand and explain.

Section 3: Application of Machine Learning involves the application of both supervised and unsupervised learning techniques to the dataset. The purpose of this section is to analyze the performance of different machine learning models and to identify the best-performing model. Various machine learning techniques such as regression tree, random forest, bagging applied for supervised learning tasks. Unsupervised learning techniques such as Principal Component Analysis, K-means Clustering, Hierarchical Clustering are also applied for analyzing patterns and insights in the dataset.

Data Dictionary

Variable Name	Brief Definition	Data Type
id	Unique identifier for each house	Integer
date	Date when the house was sold	Date
price	Price of the house	Numeric
bedrooms	Number of bedrooms in the house	Integer
bathrooms	Number of bathrooms in the house	Numeric
sqft_living	Square footage of the house's interior living space	Numeric
sqft_lot	Square footage of the land on which the house is situated	Numeric
floors	Number of floors in the house	Numeric
waterfront	Whether the house has a view of the waterfront (1 = yes, 0 = no)	Binary
view	An index from 0 to 4 of how good the view of the property was	Ordinal
condition	An index from 1 to 5 on the condition of the house	Ordinal
grade	An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high-quality level of construction and design	Ordinal
sqft_above	Square footage of house apart from basement	Numeric
sqft_basement	Square footage of the basement	Numeric
yr_built	Year when the house was built	Integer
yr_renovated	Year when the house was last renovated (0 if never)	Integer
zip code	Zip code of the area where the house is located	Categorical
lat	Latitude coordinate of the house's location	Numeric
long	Longitude coordinate of the house's location	Numeric
sqft_living15	Average square footage of interior living space for the 15 nearest neighbors	Numeric
sqft_lot15	Average square footage of the land lots of the 15 nearest neighbors	Numeric

Figure 1. Data dictionary for the Housing Price dataset

I. EXPLORATORY DATA ANALYSIS

In this section, we delve into the process of exploring the dataset to create new data by utilizing the variables that we extract from the original dataset. Initially, we start by eliminating all the missing values to ensure a clean dataset. Next, we conduct an in-depth examination of our raw dataset.

a. Data Cleaning

Data cleaning is a crucial step in the data analysis process that involves identifying and handling missing, incorrect, or inconsistent data in the dataset. In the context of the Housing Price dataset, we have 21614 rows, and we did not observe any missing values.

b. Summary Statistics

variables	mean	sd	kurtosis	skewness	min	Q1	median	Q3	max	IQR	mode	count
price	540088.1	367127.2	34.57726	4.02379	75000	321950	450000	645000	7700000	323050	450000	21613
bedrooms	3.370842	0.930062	49.05203	1.974163	0	3	3	4	33	1	3	21613
bathrooms	2.114757	0.770163	1.279329	0.511072	0	1.75	2.25	2.5	8	0.75	2.5	21613
sqft_living	2079.9	918.4409	5.241603	1.471453	290	1427	1910	2550	13540	1123	1300	21613
sqft_lot	15106.97	41420.51	285.0116	13.05911	520	5040	7618	10688	1651359	5648	5000	21613
floors	1.494309	0.539989	-0.48489	0.616134	1	1	1.5	2	3.5	1	1	21613
waterfront	0.007542	0.086517	127.6027	11.38432	0	0	0	0	1	0	0	21613
view	0.234303	0.766318	10.89022	3.395514	0	0	0	0	4	0	0	21613
condition	3.40943	0.650743	0.525364	1.032733	1	3	3	4	5	1	3	21613
grade	7.656873	1.175459	1.190379	0.77105	1	7	7	8	13	1	7	21613
sqft_above	1788.391	828.091	3.401239	1.446564	290	1190	1560	2210	9410	1020	1300	21613
sqft_basement	291.509	442.575	2.714668	1.577856	0	0	0	560	4820	560	0	21613
yr_built	1971.005	29.37341	-0.65753	-0.46977	1900	1951	1975	1997	2015	46	2014	21613
yr_renovated	84.40226	401.6792	18.69655	4.549178	0	0	0	0	2015	0	0	21613
lat	47.56005	0.138564	-0.67643	-0.48524	47.1559	47.471	47.5718	47.678	47.7776	0.207	47.6846	21613
long	-122.214	0.140828	1.048981	0.884992	-122.519	-122.328	-122.23	-122.125	-121.315	0.203	-122.29	21613
sqft_living15	1986.552	685.3913	1.596449	1.108104	399	1490	1840	2360	6210	870	1540	21613
sqft_lot15	12768.46	27304.18	150.728	9.506083	651	5100	7620	10083	871200	4983	5000	21613

By using descriptive analysis, we can gain a better understanding of our data and describe the different variables. Based on our analysis, we observed that the price has a wide range from 75,000 to 7.7 million, with a mean of 540,088 and standard deviation of 367,127. There are outliers in the dataset as seen in the maximum

values of sqft_lot, waterfront, and yr_renovated. Also, the kurtosis and skewness values indicate non-normal distribution in some variables such as bedrooms, sqft_lot, and view.

c. Data Visualization

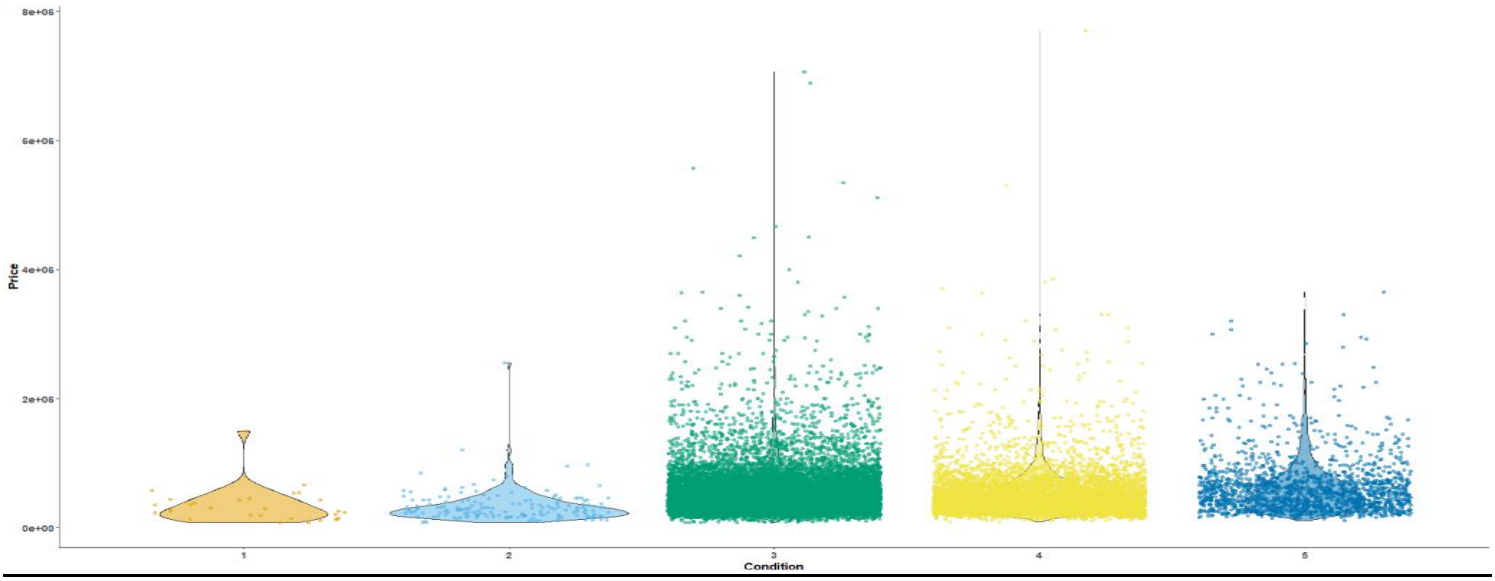


Figure 1. Violin plot between price and conditions of the house.

A violin plot like Figure 1 is like a box plot in that it shows the median, quartiles, and range of the data, but it also displays the shape of the distribution through a kernel density plot. The density plot is mirrored and rotated to form a symmetric shape that resembles a violin. The width of the violin at a given point represents the density of the data at that point. The thinner parts of the violin indicate lower density, while the wider parts indicate higher density. Overall, we can find that condition 3 has the most density in the dataset while condition 1 has the least. The same thing happens to the range of points.

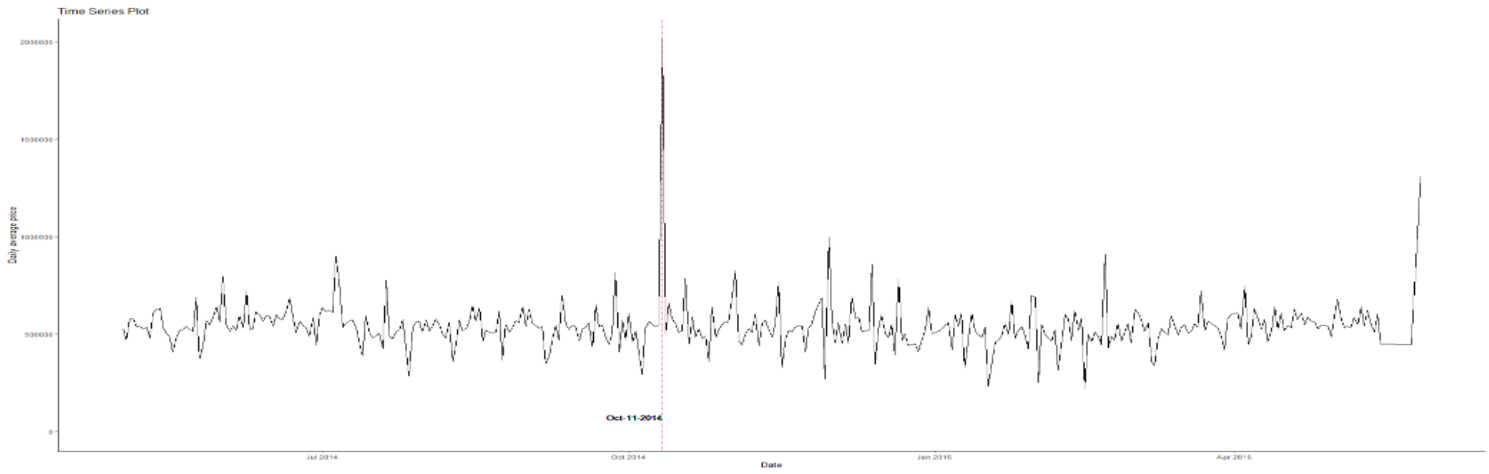


Figure 2. Time series between date and daily average price

A

time series is a set of data points that are measured over time and are typically displayed in chronological order. In this case, Figure 2 relates to the daily average price for houses from May 2014 to May 2015. It is important to note that there is an outlier in the data, which was found on October 11, 2014. An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In this context, it means that the daily average price for houses on October 11, 2014, is significantly different from the other values in the time series. This outlier could have been caused by various factors such as a one-time event or a data error, which we will confirm later.

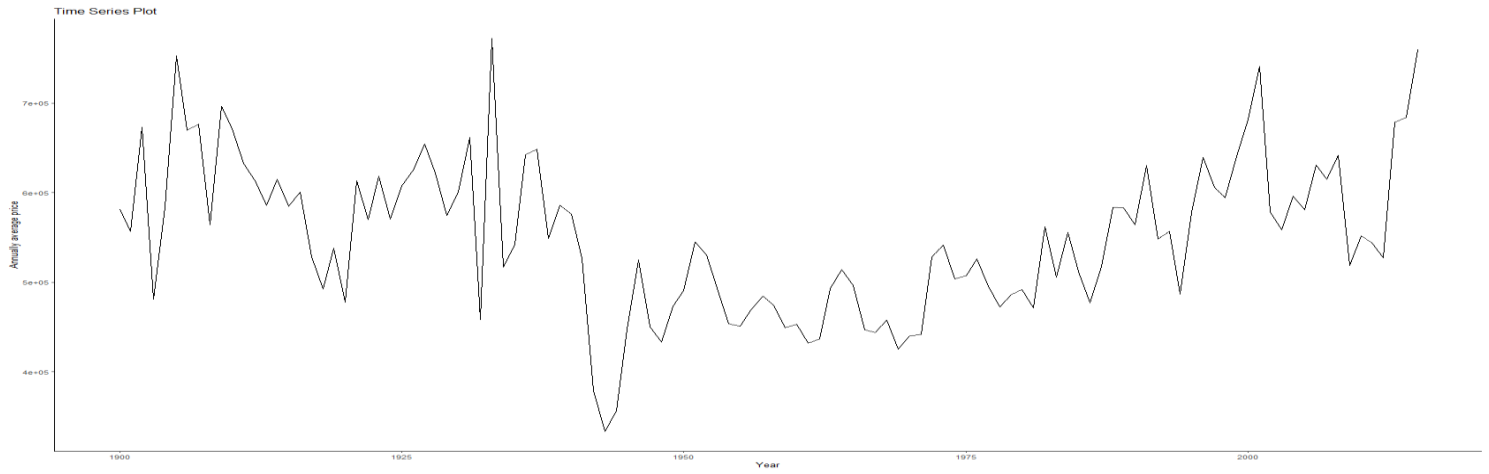


Figure 3. Time series between built year and annually price.

Figure 3. Average Arrival and Departure Delay between (A) Los Angeles and New (B) York Chicago and Miami (C) Atlanta and New York (D) LOS Angeles and San Francisco

In Figure 3, the time series relates to the annual average price for houses built between the years 1900 to the present day. The data shows a significant decrease in house prices during World War II and the 2008 economic crisis. These events had a profound impact on the housing market, with the demand for housing decreasing and causing prices to fall. However, during the last few decades of the last century, the housing market saw a flat growth trend, with prices remaining relatively stable. This could be attributed to several factors, such as increasing population, urbanization, and changes in government policies regarding housing. Overall, this time series illustrates the impact of major events.

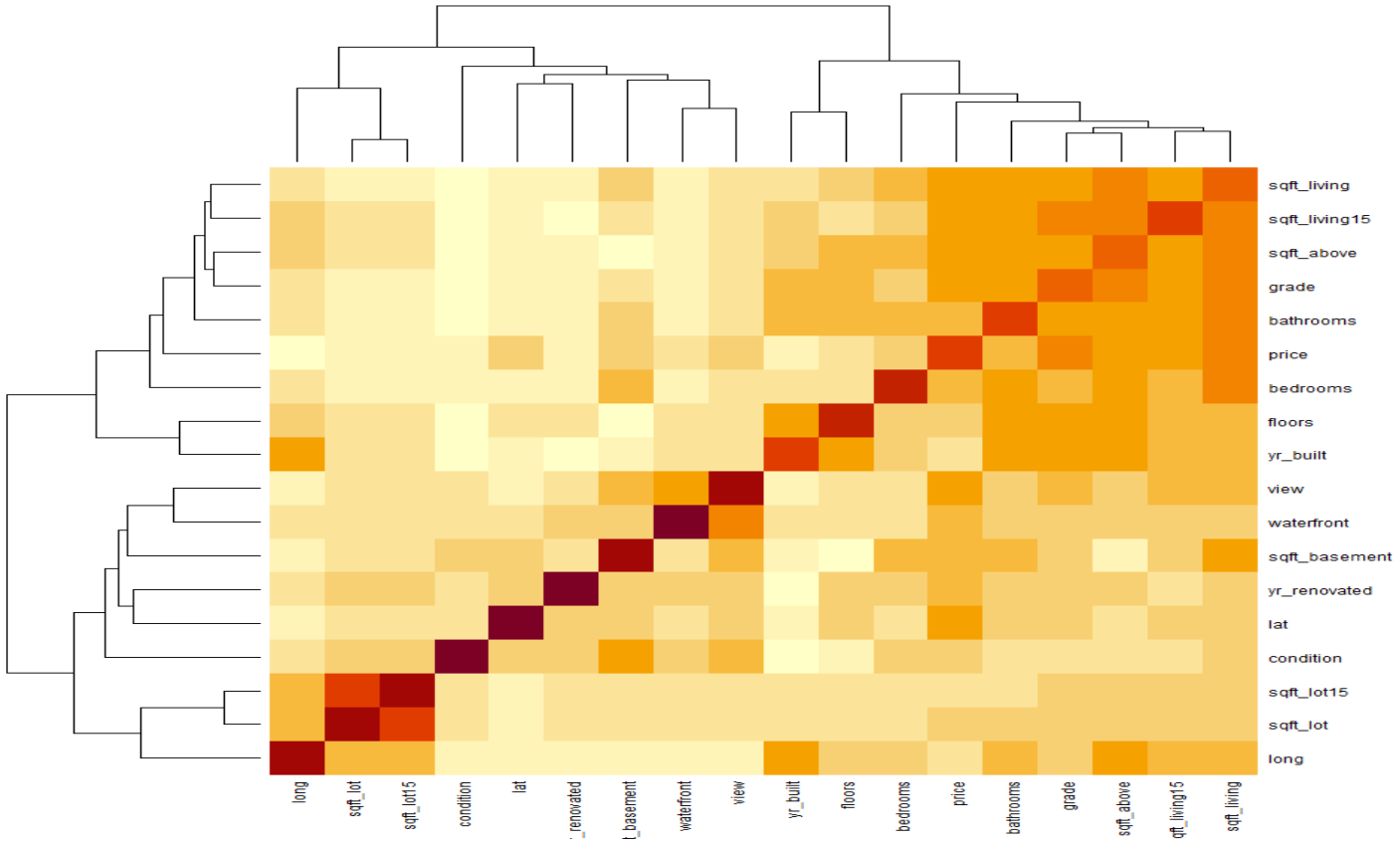


Figure 4. Heatmap between those important variables.

Next, we have a correlation heatmap regarding most variables with few useless variables like ID removed in Figure 4. The darker the color is, the deeper these two variables correlated. Squares on the diagonal are dark red since they are variables to themselves. It is obvious that the sqft_lot15 and sqft_lot have a strong correlation. Meaning that the internal living space and total land space are highly correlated. Then, we can find in the right up corner,

most of the squares have colorful squares. Consequently, sqft_living, sqft_living15, sqft_above, grade, bathrooms, price and bedrooms. We will discuss more about relation between price and theses variables later.

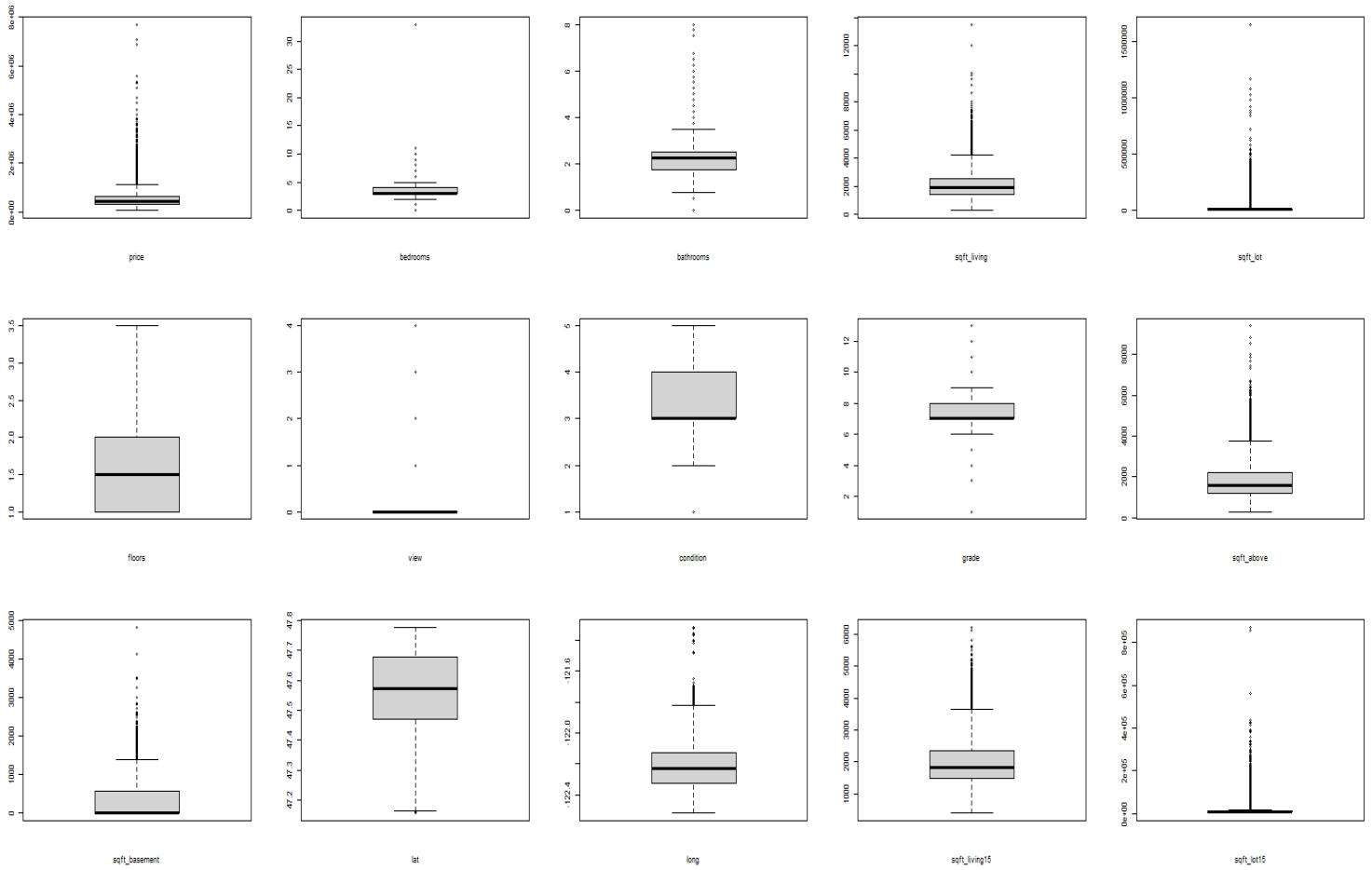


Figure 5. Boxplots for 15 meaningful variables.

After, we part away from some other unrelated variables like year built, year renovated and zip code and draw boxplots for the rest variables in Figure 5. We found that many variables left have lots of outliers comparing to the interval on boxplots except those categorical variables. Among those, we can find the flattest variable is lat. Recall

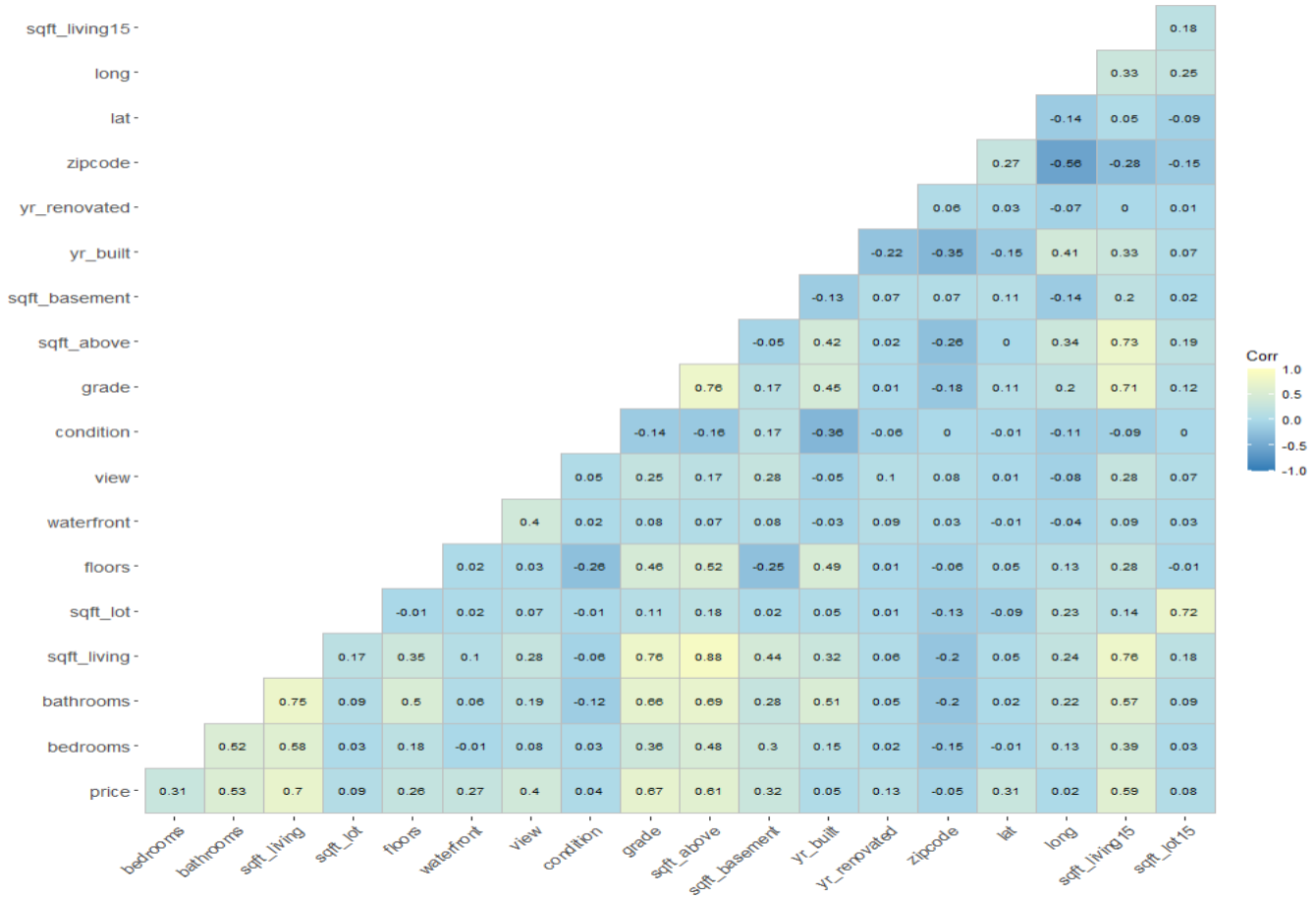
what we observed above, sqft_lot and sqft_lot15 have a strong correlation. We can confirm that by viewing the boxplots of those two variables, which are almost the same.

II. Feature Selection

The goal is to choose a subset of informative features that have the highest predictive power and are easy to interpret. We will apply correlation-based feature selection and variance-based feature selection to reduce the dimensionality of the dataset to improve the performance of the machine learning models.

a. Correlation-Based Feature Selection

Correlation-based feature selection is a method used to identify the most important features in a dataset by



examining their correlation with the target variable. In this method, we first compute the correlation matrix between the target variable and all the others numeric predictor variables. We then select the features with the highest absolute correlation values. This is usually done by setting a correlation cutoff value and selecting all the features that have a correlation value greater than the cutoff. In the Housing Price dataset, we decide to use the price as the target variable and a cutoff of 0.5.

After performing correlation-based feature selection on the dataset, we found that there are five variables that have a correlation greater than 0.5 with the target variable (price). These variables, ranked in order of highest to lowest correlation, are sqft_living, grade, sqft_above, sqft_living15, and bathrooms.

b. Variance-Based Feature Selection

VARIABLE	VARIANCE
sqft_lot	1715658774
sqft_lot15	745518225.3
sqft_living	843533.6814
sqft_above	685734.6673
sqft_living15	469761.2399
sqft_basement	195872.6684
yr_renovated	161346.2119
yr_built	862.7973
grade	1.3817
bedrooms	0.865
bathrooms	0.5932
view	0.5872
condition	0.4235
floors	0.2916
long	0.0198
lat	0.0192
waterfront	0.0075

Based on the variance-based feature selection method, we calculated the variance of each numeric variable in the dataset. We have eight variables with high variance as you can see in the table. The highest variance is sqft_lot with a variance of 1715658774, followed by sqft_lot with a variance of 745518225.3404. These variables could be considered as important features for our analysis.

c. Final Selection of Feature

After performing both correlation-based and variance-based feature selection, we have identified the most important variables in our dataset. These variables are sqft_living, sqft_above, sqft_living15, and sqft_basement. These variables have a high correlation with the target variable (price) and have a variance above the threshold. Therefore, we will use these variables for our final analysis. We also aggregate our data set by grade and zip code and compute the mean value for each most important variable in our dataset.

III. Application of Machine learning Techniques

In the third part of our analysis, we applied various machine learning techniques to predict the housing prices based on the selected features. We used both supervised and unsupervised learning techniques to explore the data and develop models. The supervised learning techniques included Random Forest, Regression Tree, Bagging. On the other hand, the unsupervised learning techniques included Isolation Forest, Local Outlier Factor (LOF), Distance to All Points, and Distance to Nearest Neighbors (k-Distance).

a. Supervised Learning Techniques

i. Random Forest

Random forest is a popular supervised learning algorithm for both classification and regression tasks. It is an ensemble learning method that builds multiple decision trees on randomly selected subsets of the training data and then combines their outputs to make the final prediction. Each decision tree is constructed using a different subset of the features, making the algorithm less prone to overfitting, and improving its generalization performance. One advantage of random forest is its ability to provide estimates of feature importance, which can be used for feature selection and understanding the relationship between the features and the target variable. The importance of a feature is measured by the mean decrease impurity (MDI) or mean decrease accuracy (MDA), which calculates the reduction in the impurity or accuracy caused by the feature. A higher value indicates a more important feature. The math formula for MDI can be expressed as: $MDI(j) = \frac{\sum impurity(t) - impurity(t|j)}{B}$.

We apply random forest to the housing price dataset:

- First, we randomly split the dataset into a training set and a testing set. 80% of the data is used for training and 20% is used for testing.
- Then, we fit three different random forest models to the data. The dependent variable in all three models is **grade** and the independent variables are all the other variables in the dataset.
- For each random forest model (home.rf.3, home.rf.4, and home.rf.5), we specify the number of trees in the forest (ntree = 500) and the number of variables to be considered at each split (mtry).

For mtry = 3

```
Call:
 randomForest(formula = grade ~ ., data = train, ntree = 500,      mtry = 3, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 0.2888674
                    % Var explained: 94.58
```

For mtry = 4

```
Call:
 randomForest(formula = grade ~ ., data = train, ntree = 500,      mtry = 4, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 0.2888845
                    % Var explained: 94.58
```

For mtry = 5

```
Call:
 randomForest(formula = grade ~ ., data = train, ntree = 500,      mtry = 5, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 0.290207
                    % Var explained: 94.55
```

ii. Regression Tree

Regression tree analysis is a widely used method in machine learning and statistics for predicting continuous numerical values. It is a decision tree-based model that recursively partitions the data into subsets based on the independent variables and constructs a tree-like model to predict the dependent variable.

Regression trees are easy to interpret and visualize and are particularly useful when the relationship between the independent and dependent variables is non-linear or complex. In addition, regression trees can handle both continuous and categorical independent variables and are resistant to outliers.

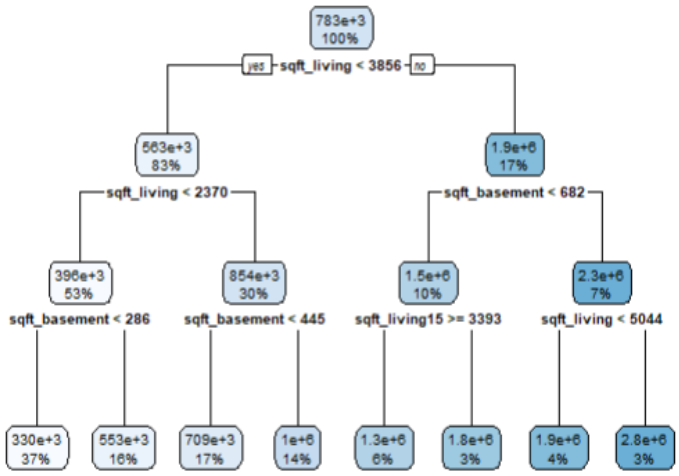
We first created a new dataset that contains only the relevant columns from our feature selection from the original dataset. Next, we split the new dataset into training and test sets with 80% of the data used for training and the remaining 20% for testing.

We use the rpart() function to create a regression tree by specifying the formula for the tree and the data to use. After that, we visualize the tree structure and decision-making process using the rpart.plot() function.

```
n= 347

node), split, n, deviance, yval
      * denotes terminal node

 1) root 347 1.468638e+14  783111.2
   2) sqft_living< 3855.5 288 3.034481e+13  562883.3
     4) sqft_living< 2369.796 183 5.036907e+12  395594.5
       8) sqft_basement< 286.1443 129 1.884847e+12  329871.8 *
       9) sqft_basement>=286.1443 54 1.263728e+12  552598.9 *
     5) sqft_living>=2369.796 105 1.126077e+13  854443.7
      10) sqft_basement< 445.3254 58 3.348459e+12  708871.6 *
      11) sqft_basement>=445.3254 47 5.166468e+12 1034086.0 *
   3) sqft_living>=3855.5 59 3.436765e+13 1858122.0
     6) sqft_basement< 681.682 33 1.133227e+13 1510598.0
      12) sqft_living15>=3392.857 22 5.248410e+12 1346369.0 *
      13) sqft_living15< 3392.857 11 4.303766e+12 1839055.0 *
     7) sqft_basement>=681.682 26 1.399135e+13 2299210.0
      14) sqft_living< 5043.75 15 4.999272e+12 1945826.0 *
      15) sqft_living>=5043.75 11 4.564492e+12 2781098.0 *
```



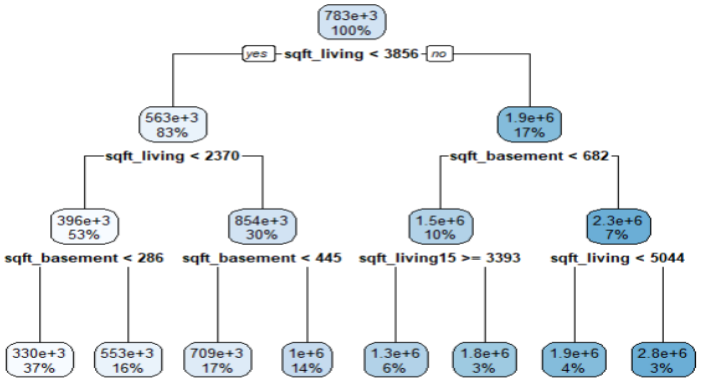
We then use our model to predict the value from the test data and then we evaluate the performance of the model by calculating a metric such as the mean squared error.

We prune the tree using the prune() function to improve its performance and visualize the tree structure of the prune.

```
n= 347

node), split, n, deviance, yval
      * denotes terminal node

 1) root 347 1.468638e+14  783111.2
   2) sqft_living< 3855.5 288 3.034481e+13  562883.3
     4) sqft_living< 2369.796 183 5.036907e+12  395594.5
       8) sqft_basement< 286.1443 129 1.884847e+12  329871.8 *
       9) sqft_basement>=286.1443 54 1.263728e+12  552598.9 *
     5) sqft_living>=2369.796 105 1.126077e+13  854443.7
      10) sqft_basement< 445.3254 58 3.348459e+12  708871.6 *
      11) sqft_basement>=445.3254 47 5.166468e+12 1034086.0 *
   3) sqft_living>=3855.5 59 3.436765e+13 1858122.0
     6) sqft_basement< 681.682 33 1.133227e+13 1510598.0
      12) sqft_living15>=3392.857 22 5.248410e+12 1346369.0 *
      13) sqft_living15< 3392.857 11 4.303766e+12 1839055.0 *
     7) sqft_basement>=681.682 26 1.399135e+13 2299210.0
      14) sqft_living< 5043.75 15 4.999272e+12 1945826.0 *
      15) sqft_living>=5043.75 11 4.564492e+12 2781098.0 *
```



iii. Bagging

Bagging is another popular ensemble method for improving the accuracy of predictive models. The basic idea of bagging is to create multiple copies of the original dataset using bootstrap sampling, and to train a model on each of these bootstrap samples. The final prediction is then made by averaging the predictions of all the models.

We used the same slipping and data set as we used for the previous model. After that, we can apply Bagging model to the training data using the train() function from the caret package.

	Length	Class	Mode
y	347	-none-	numeric
X	0	-none-	NULL
mtrees	25	-none-	list
OOB	1	-none-	logical
comb	1	-none-	logical
xNames	4	-none-	character
problemType	1	-none-	character
tuneValue	1	data.frame	list
obsLevels	1	-none-	logical
param	0	-none-	list

We then predict value from our test data and evaluate the performance of the Bagging model using a metric such as mean squared error.

b. Unsupervised Learning Techniques

i. Principal Component Analysis

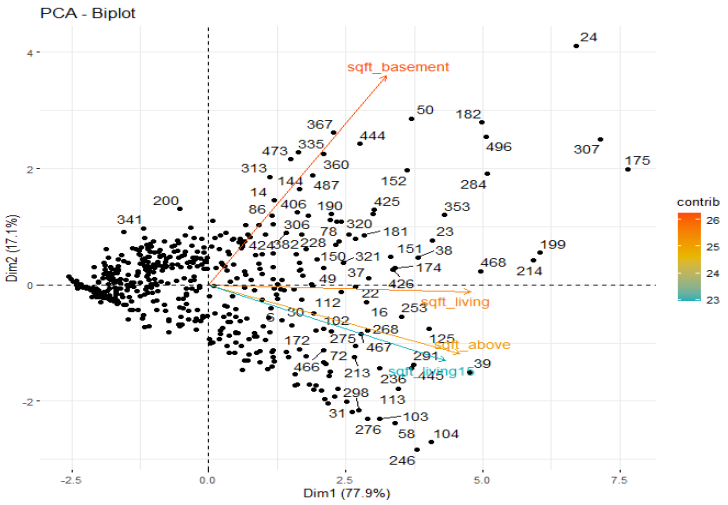
Principal Component Analysis (PCA) is a widely used dimensionality reduction technique in machine learning and data science. It is a statistical method that transforms high-dimensional data into a smaller set of orthogonal variables known as principal components (PCs) while retaining as much of the original variability in the data as possible. PCA is used for various purposes such as reducing data complexity, identifying hidden patterns, visualizing high-dimensional data, feature extraction, data compression, and data pre-processing.

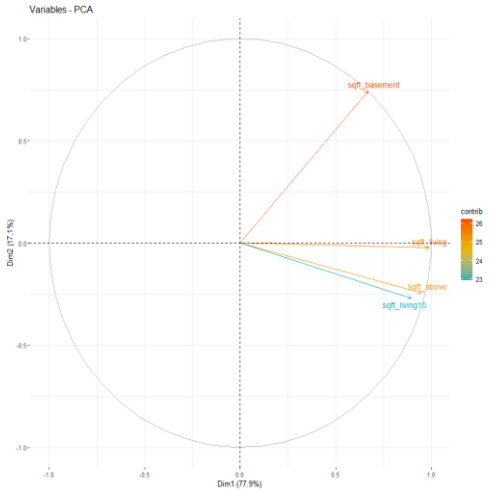
PCA works by identifying the directions in which the data varies the most, and then projecting the data onto these directions to create new variables that capture the most variation in the data. The first principal component captures the maximum amount of variance in the data, and each subsequent component captures the remaining variance while being orthogonal (uncorrelated) to the previous components. By selecting only, the top k principal components, we can reduce the dimensionality of the data from d to k, where k << d.

The PCA shows that the first principal component (Dim.1) has the highest variance (3.115) among all dimensions, accounting for 77.884% of the total variance, while the second and third principal components (Dim.2 and Dim.3) have a lower but significant amount of variance (0.684 and 0.201, respectively) and account for 17.089% and 5.027% of the total variance, respectively. The fourth principal component (Dim.4) has zero variance and therefore does not contribute to the analysis.

Looking at the individuals, we can see that the first ten individuals have been analyzed in terms of their distance (Dist) from the center, the contribution of each dimension (Dim.1, Dim.2, and Dim.3) to their position, and the squared cosine (cos2) of the angle between the individual and each dimension.

Finally, the variables have been analyzed in terms of their contribution to each dimension (Dim.1, Dim.2, and Dim.3) and their squared cosine (cos2) to the dimension. The variable "sqft_living15" has the highest contribution to the first principal component (Dim.1) with a cosine value of 0.972, followed by "sqft_above" with a cosine value of 0.898, and "sqft_living" with a cosine value of 0.799. The variable "sqft_basement" has the lowest contribution to all three dimensions with a cosine value of 0.003.

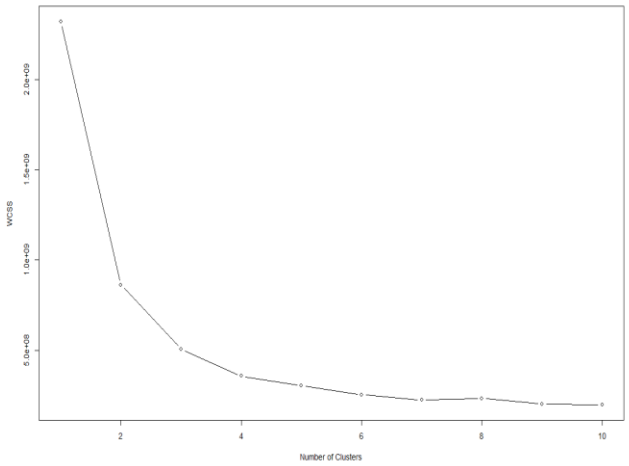


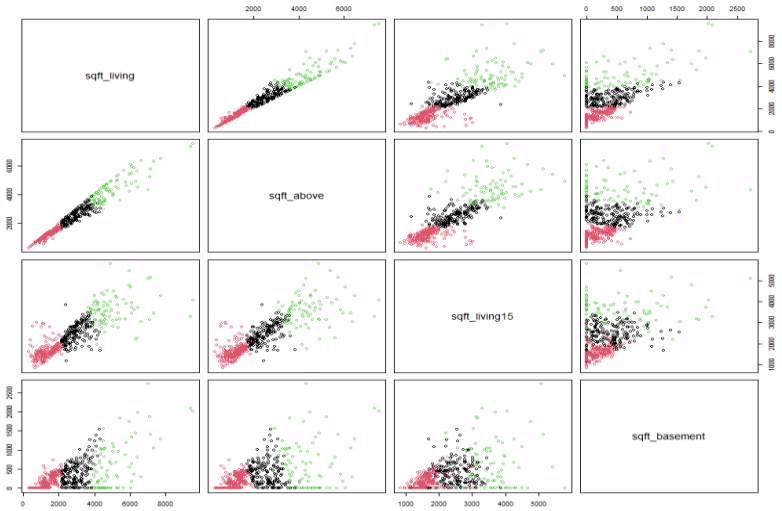


ii. K-means clustering

K-means clustering is a popular unsupervised machine learning technique used for grouping similar data points together in clusters. The algorithm works by partitioning the data into K distinct clusters based on the distance between data points and the center of the cluster, also known as the centroid. The K-means algorithm is iterative, where each iteration involves reassigning the data points to the nearest cluster centroid and then recomputing the centroid of each cluster. The algorithm stops when the within-cluster sum of squares (WCSS) no longer decreases significantly, or when the maximum number of iterations is reached. K-means clustering can be applied to a variety of datasets and is particularly useful for exploratory data analysis, segmentation, and pattern recognition.

We use the data set with the most important feature that we derived from our feature selection section. We apply the K-means clustering with different levels of cluster (K). Based on the figure below we can see that the optimal K = 3.

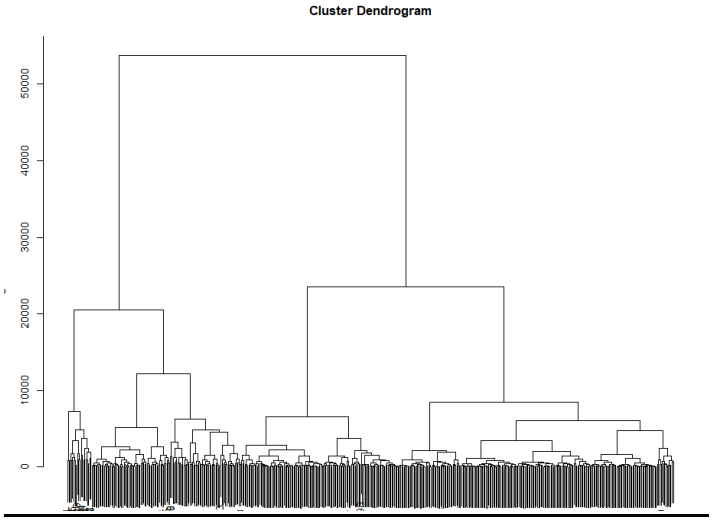


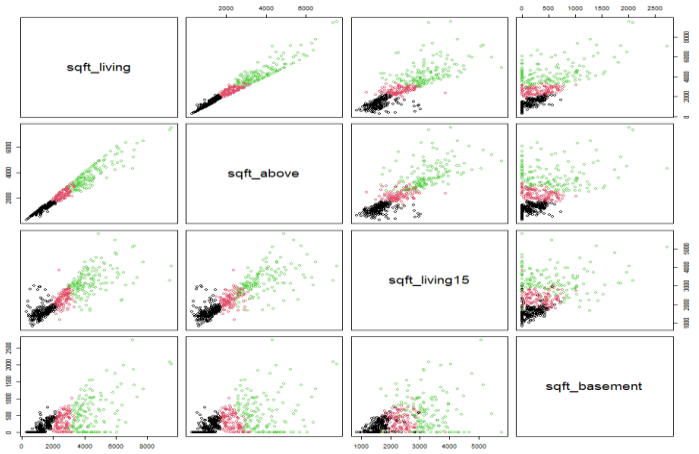


The output of the k-means algorithm shows that the data has been clustered into 3 groups with sizes of 182, 82, and 232. The cluster means for each group are also provided, which show the average value of each feature for each cluster. The clustering vector indicates which cluster each data point belongs to, with the cluster label represented by a number from 1 to 3. This information can be used to gain insight into the structure of the data and to make predictions about new data points based on their cluster membership.

iii. Hierarchical Clustering

Hierarchical Clustering is a clustering algorithm used in unsupervised machine learning to group similar data points together. The algorithm forms a tree-like structure (dendrogram) that represents the similarities and dissimilarities among the data points. Hierarchical Clustering is a powerful technique for exploratory data analysis as it helps in identifying hidden structures in the data and provides insights into the relationships among the data points.


Cluster Dendrogram
0
10000
20000
30000
40000
50000


sqft_living
sqft_above
sqft_living15
sqft_basement

CONCLUSION

In conclusion, this report provides an in-depth analysis of a housing dataset using various machine learning techniques. The report is divided into three main sections: Exploratory Data Analysis, Feature Selection, and Application of Machine Learning. The Exploratory Data Analysis section focuses on cleaning the dataset, calculating summary statistics, and visualizing the data to gain insights. In the Feature Selection section, various techniques such as correlation-based and variance-based feature selection are used to identify the most important features affecting housing prices. By reducing the dimensionality of the dataset, overfitting can be reduced, and the performance of the models improved. In the Application of Machine Learning section, both supervised and unsupervised learning techniques such as regression tree, random forest, bagging, PCA, K-means clustering, and hierarchical clustering are applied to the dataset. The purpose of this section is to analyze the performance of different machine learning models and identify the best-performing model. Overall, this report provides a comprehensive understanding of the housing dataset and the use of machine learning techniques for analyzing and predicting housing prices.

RESPONSIBILITY OF WORK:

Tangbufan Wei did the data dictionary, data visualization and coding. Halilou Junior Timite is responsible for the problem statement, feature selection, algorithms and their applications and conclusion. Zhiyi Lin form all the code into a r markdown file.