MINI PROJECT 03

Neural Networks

Daniel Alejandro Morales Castillo

Real Estate
Price Prediction

Abstract

BRIEF INTRODUCTION

Getting a good estimate of the price of a house is hard even for the most talented real estate agents. This why we we want to predict the price of houses given their features using artificial neural networks.

Real Estate
Price
Prediction

Index

Basic Data Analysis

Real Estate
Price
Prediction

Real Estate
Price
Prediction

Data domain

Real Estate

Real estate refers broadly to the property, land, buildings, and air rights that are above land, and the underground rights below it.

Land

Refers to the earth's surface down to the center of the earth and upward to the airspace above, including the trees, minerals, and water.

Residential real estate

Any property used for residential purposes.

Variables

Real Estate Price Prediction

TRANSACTION_DATE

The transaction date (for example, 2013.250=2013 March, 2013.500=2013 June, etc.)

HOUSE_AGE

Age of the house

DISTANCE_NEAREST_MRT

Distance to the nearest MRT station (unit: meter)

NUMBER_CONVENIENCE_STORES

Number of convenience stores in the living circle on foot

Variables

Real Estate Price Prediction

LATITUDE

Geographic coordinate

LONGITUDE

Geographic coordinate

HOUSE_PRICE

House price of unit area (10000 New Taiwan Dollar/Ping, where Ping is a local unit, 1 Ping = 3.3 meter squared)

Real Estate Price Prediction

How the data was recollected?

The market historical data set of real estate valuation are collected from Sindian Dist., New Taipei City, Taiwan by Prof. I-Cheng Yeh

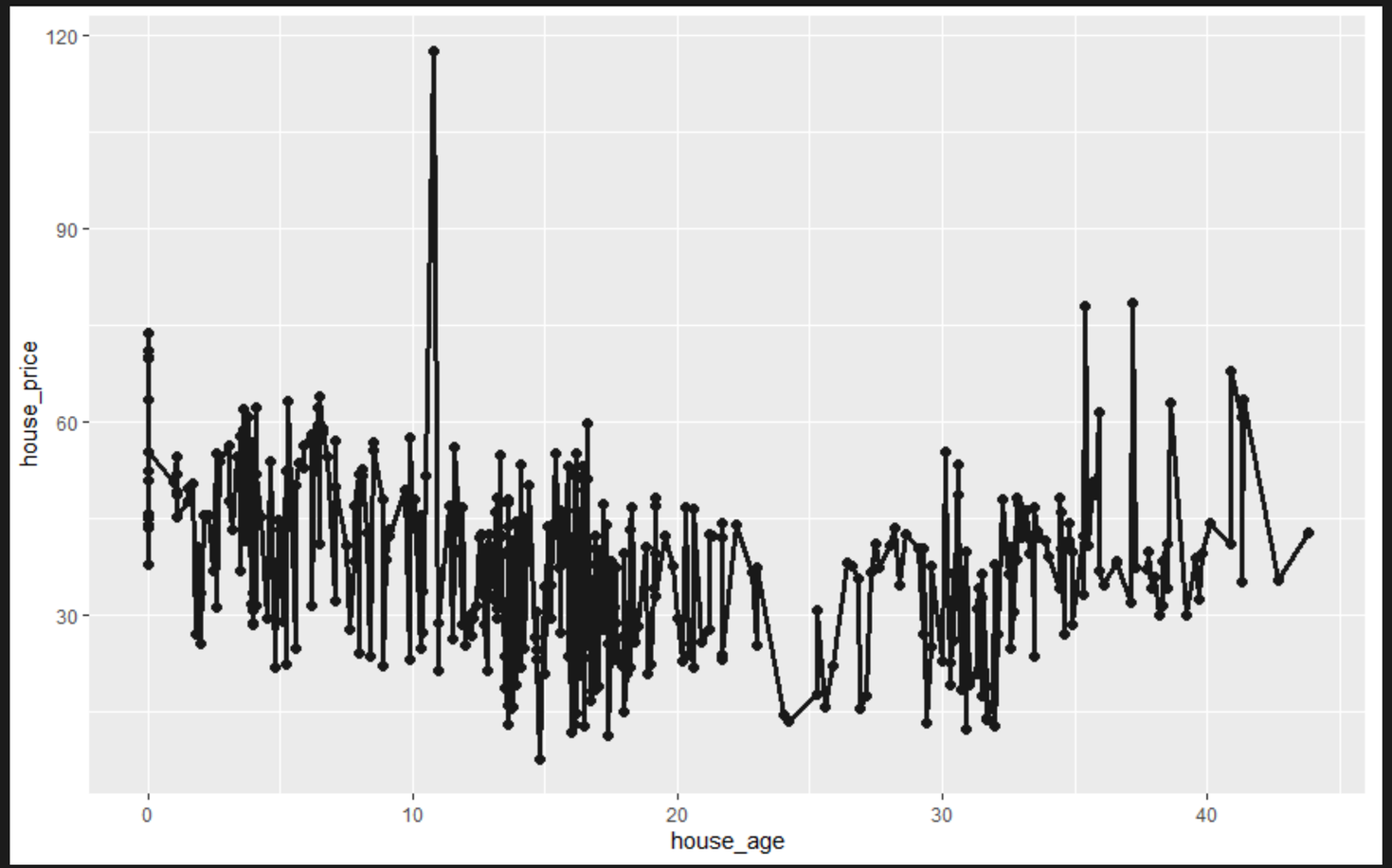
Limitations of study

It's only able to predict the prices for New Taipei City, Taiwan, limited to 416 observations.

Disadvantages

The data it's not that new, so prices now are a bit more expensive, if we want to predict modern price, we need to make some adjustments to the data set.

Interesting plots

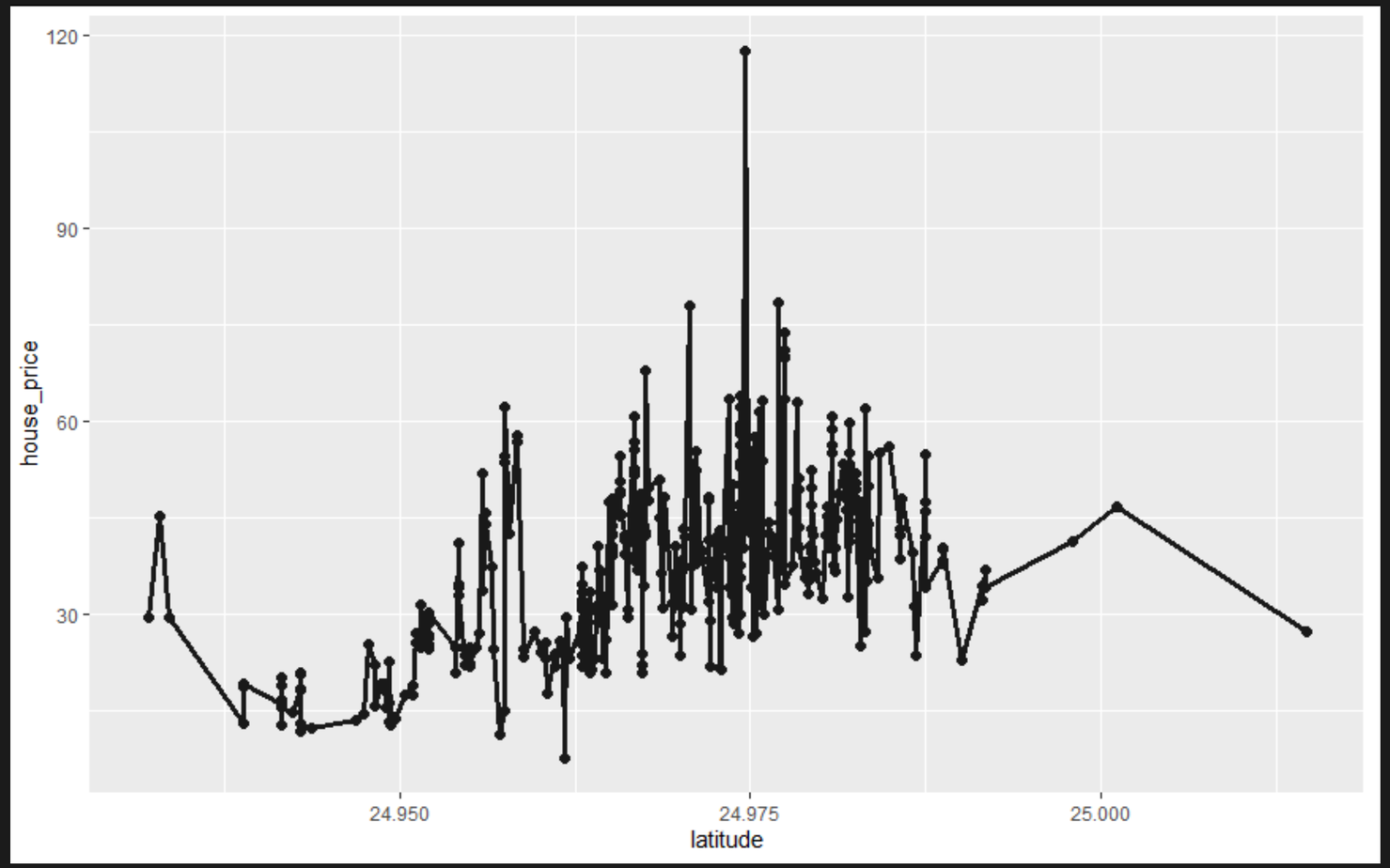


#House Age x House Price line plot

```
ggplot(realestate, aes(x = house_age, y =
house_price)) +
 geom_point(color = "#1b1b1b", size = 2) +
 geom_line(color = "#1b1b1b", size = 1.2)
```

Real Estate Price Prediction

Interesting plots

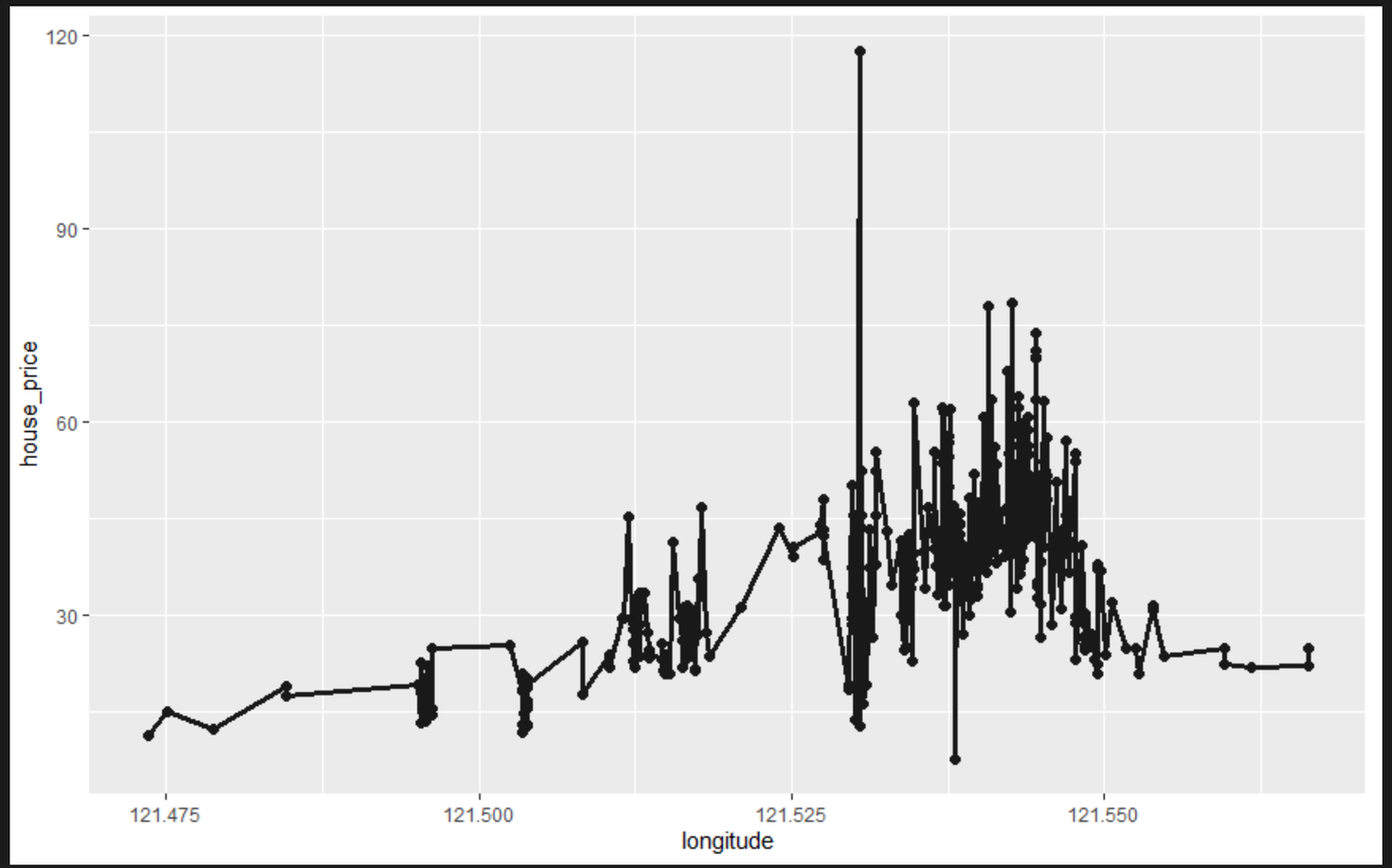


#Latitude x House Price line plot

```
ggplot(realestate, aes(x = latitude , y =
house_price)) +
 geom_point(color = "#1b1b1b", size = 2) +
 geom_line(color = "#1b1b1b", size = 1.2)
```

Real Estate Price Prediction

Interesting plots

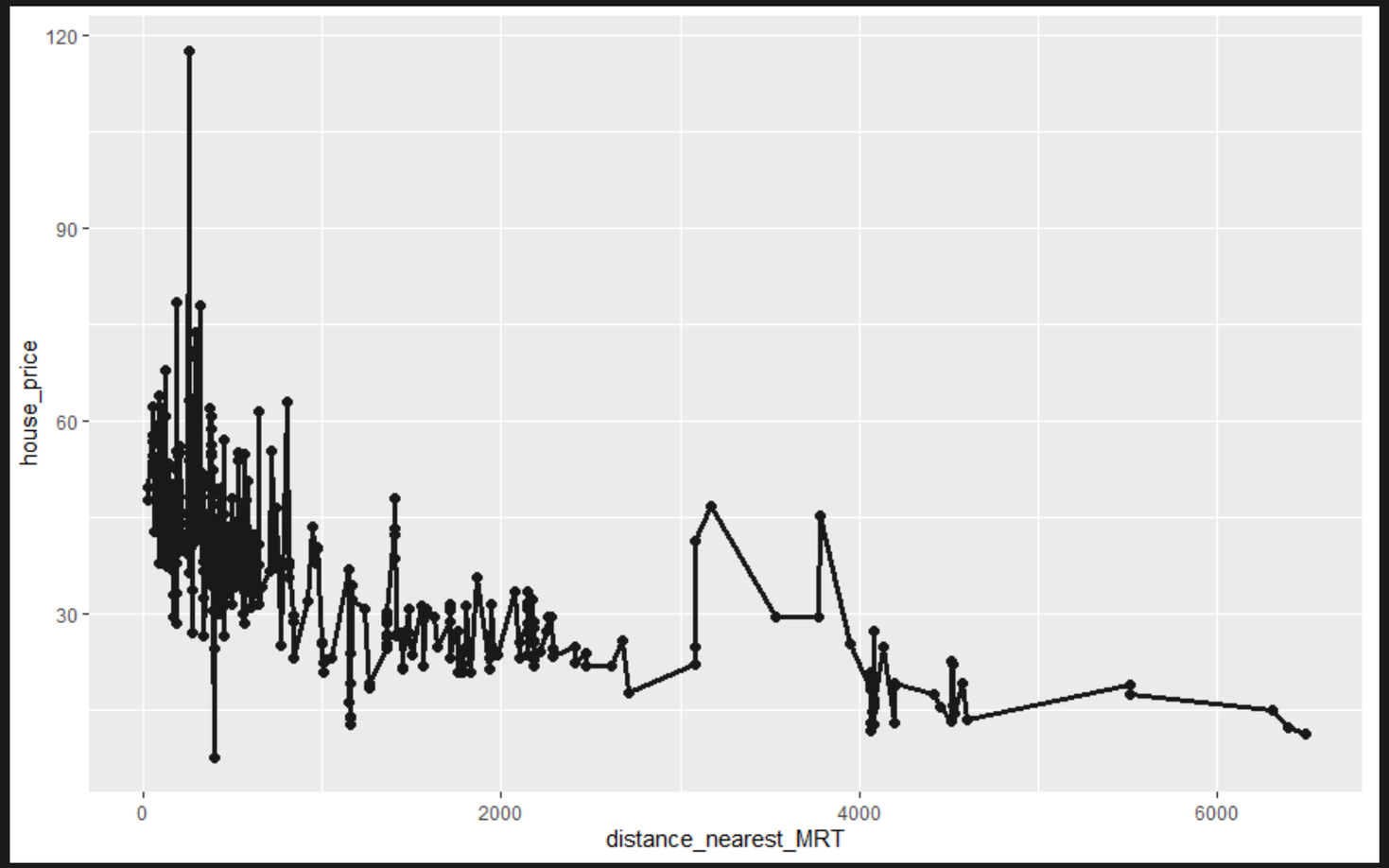


#Longitude x House Price line plot

```
ggplot(realestate, aes(x = longitude, y = house_price)) +
 geom_point(color = "#1b1b1b", size = 2) +
 geom_line(color = "#1b1b1b", size = 1.2)
```

Real Estate Price Prediction

Interesting plots

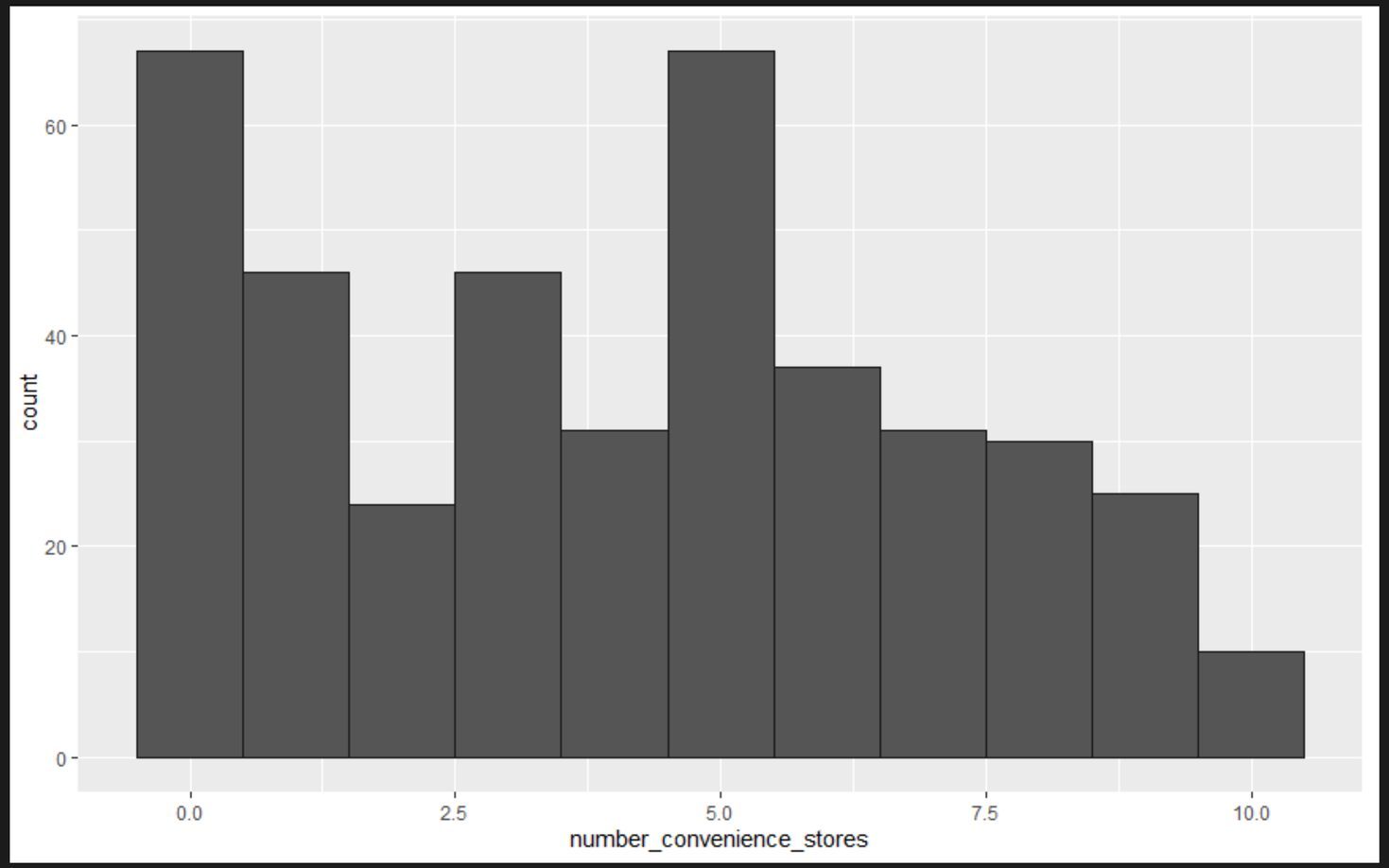


#Distance to the nearest MRT x House Price

```
ggplot(realestate, aes(x =
distance_nearest_MRT , y = house_price)) +
  geom_point(color = "#1b1b1b", size = 2) +
  geom_line(color = "#1b1b1b", size = 1.2)
```

Real Estate Price Prediction

Interesting plots

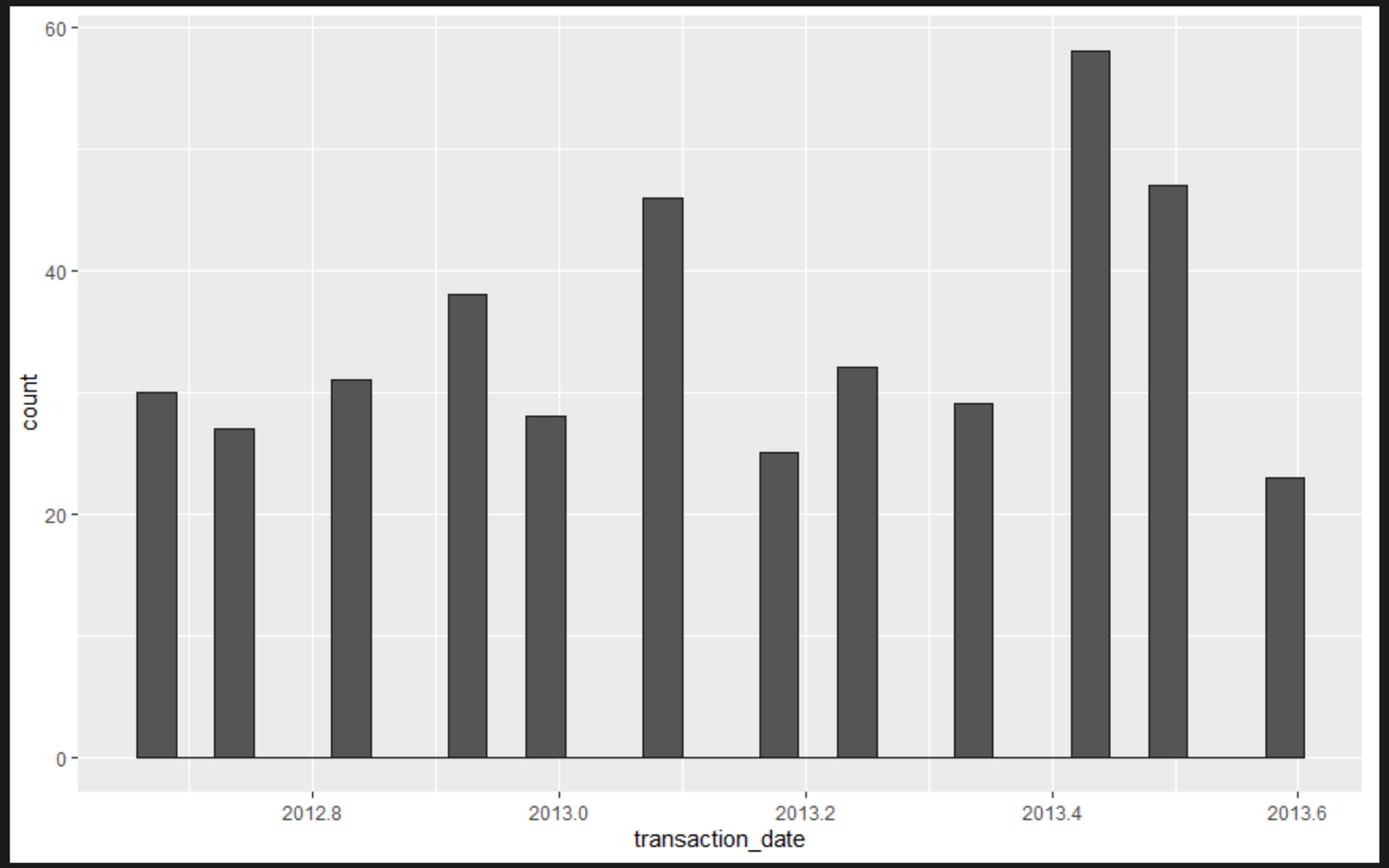


#Convenience Stores Histogram

```
ggplot(realestate,
aes(x=number_convenience_stores)) +
  geom_histogram(color = "#1b1b1b",
binwidth=1)
```

Real Estate Price Prediction

Interesting plots

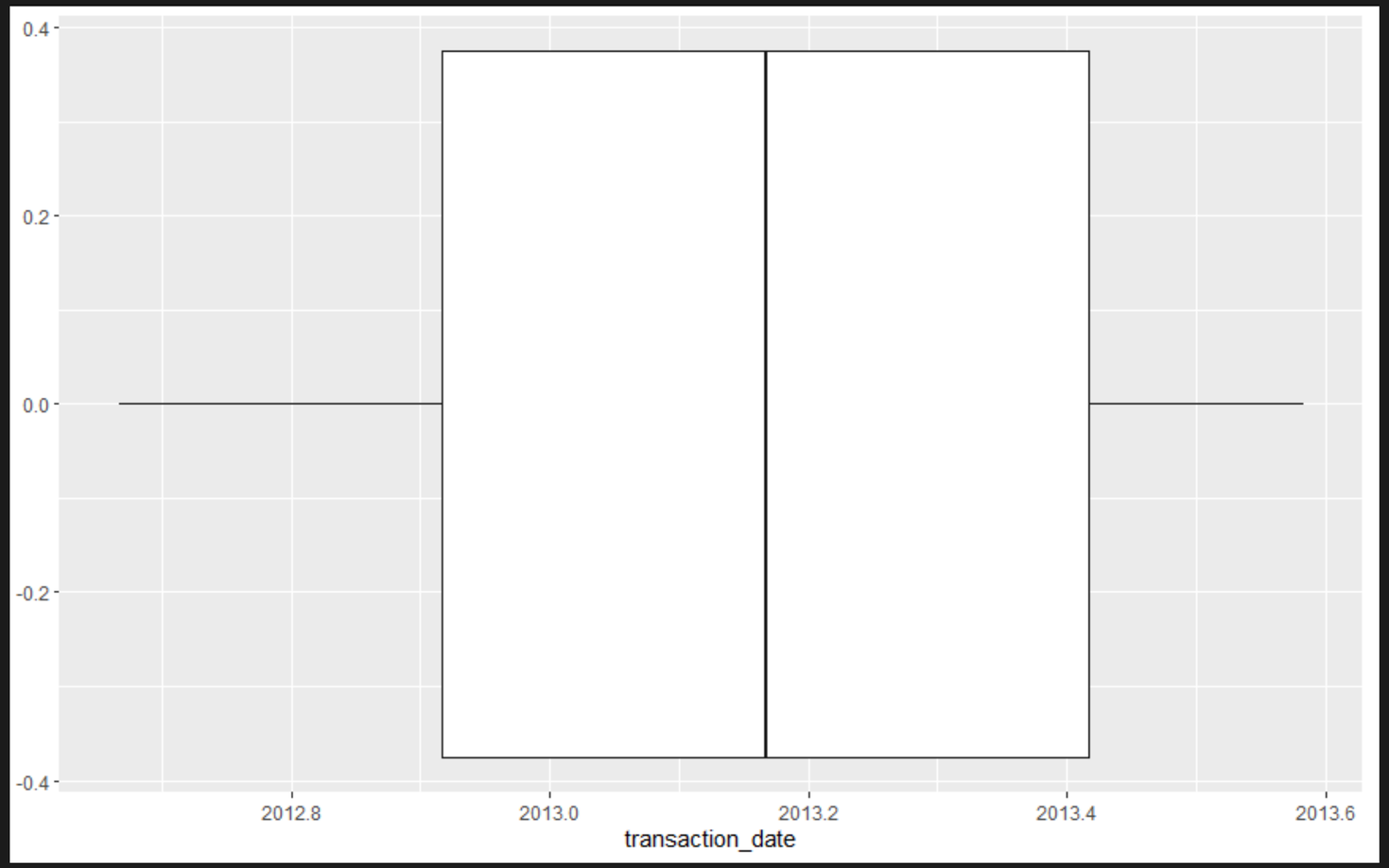


#Transaction Date Histogram

```
ggplot(realestate, aes(x=transaction_date)) +
  geom_histogram(color = "#1b1b1b")
```

Real Estate Price Prediction

Interesting plots

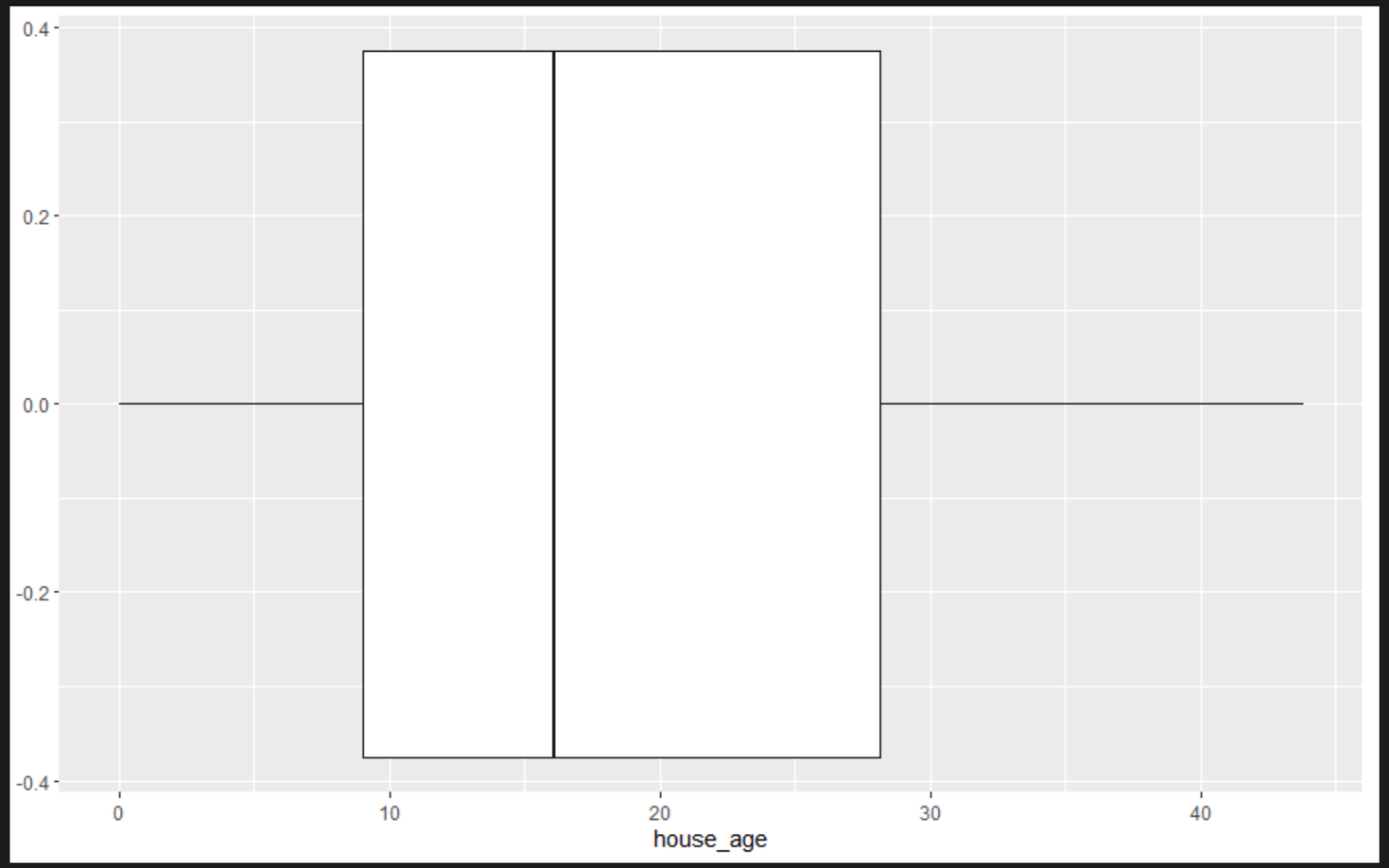


#transaction_date Box plot

```
ggplot(realestate, aes(x=transaction_date)) +
geom_boxplot(color = "#1b1b1b")
```

Real Estate Price Prediction

Interesting plots

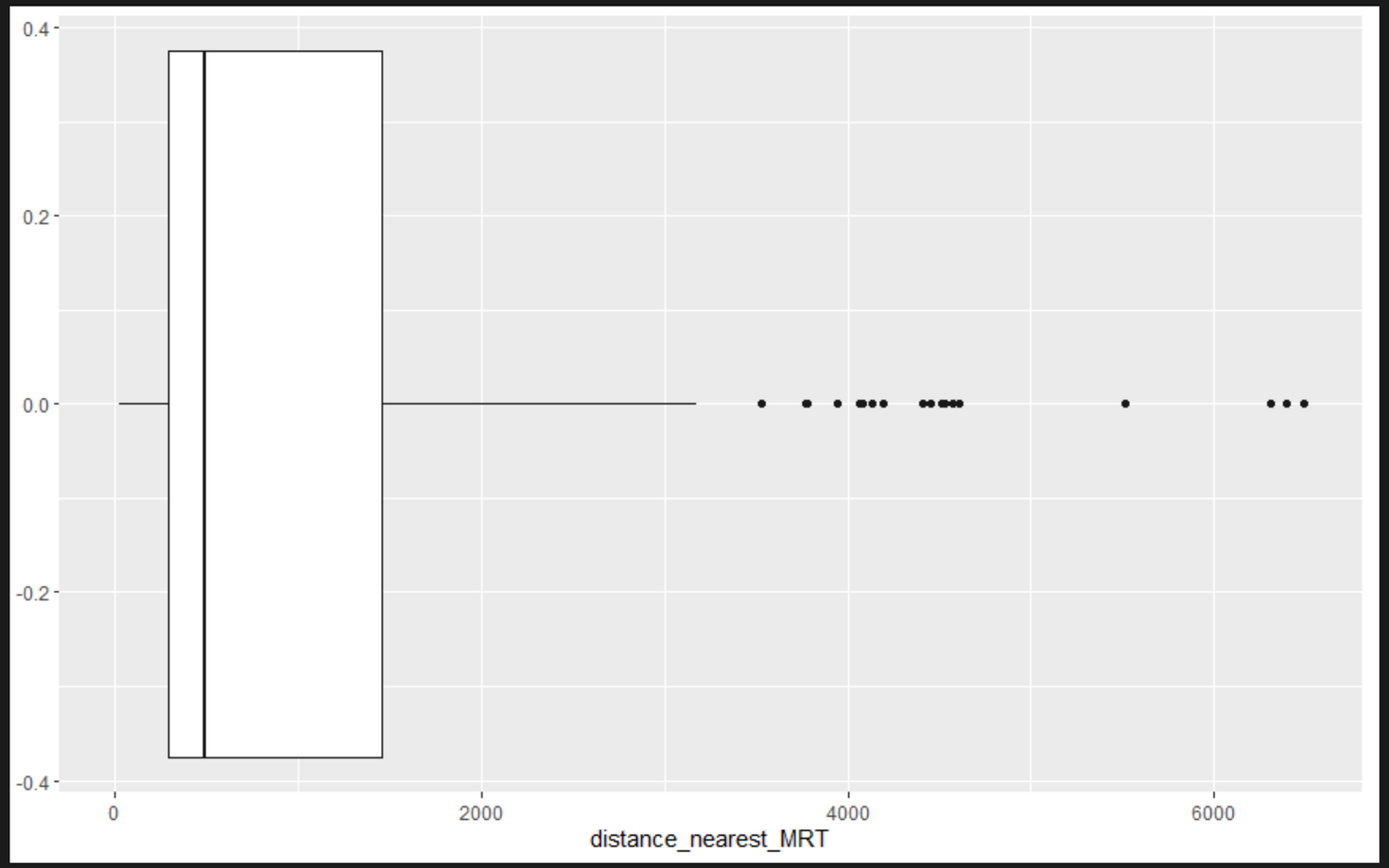


#house_age Box plot

```
ggplot(realestate, aes(x=house_age)) +
geom_boxplot(color = "#1b1b1b")
```

Real Estate Price Prediction

Interesting plots

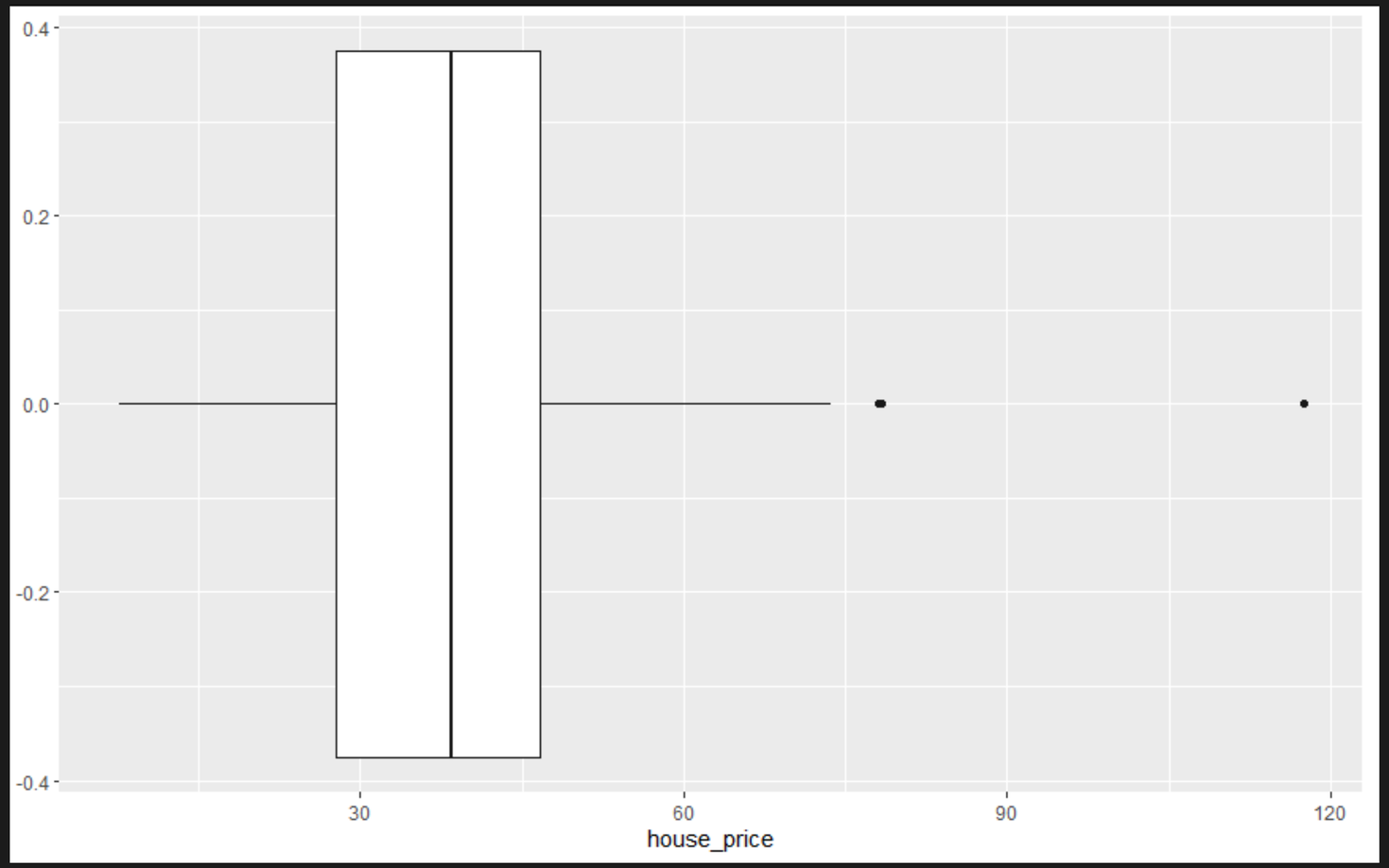


#distance_nearest_MRT Box plot

```
ggplot(realestate,
aes(x=distance_nearest_MRT )) +
geom_boxplot(color = "#1b1b1b")
```

Real Estate Price Prediction

Interesting plots

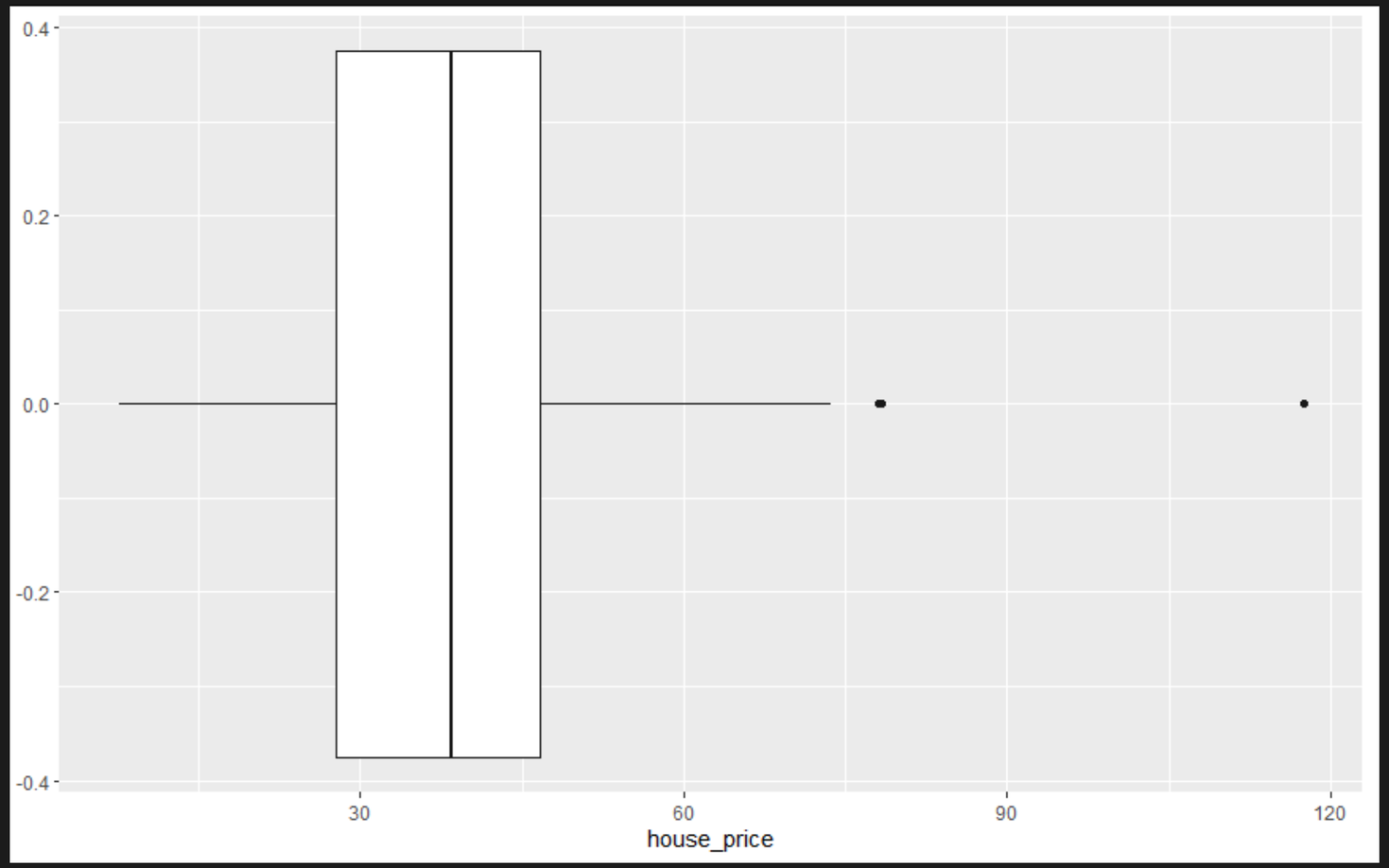


#house_price Box plot

```
ggplot(realestate, aes(x=house_price)) +
geom_boxplot(color = "#1b1b1b")
```

Real Estate Price Prediction

Interesting plots



#house_price Box plot

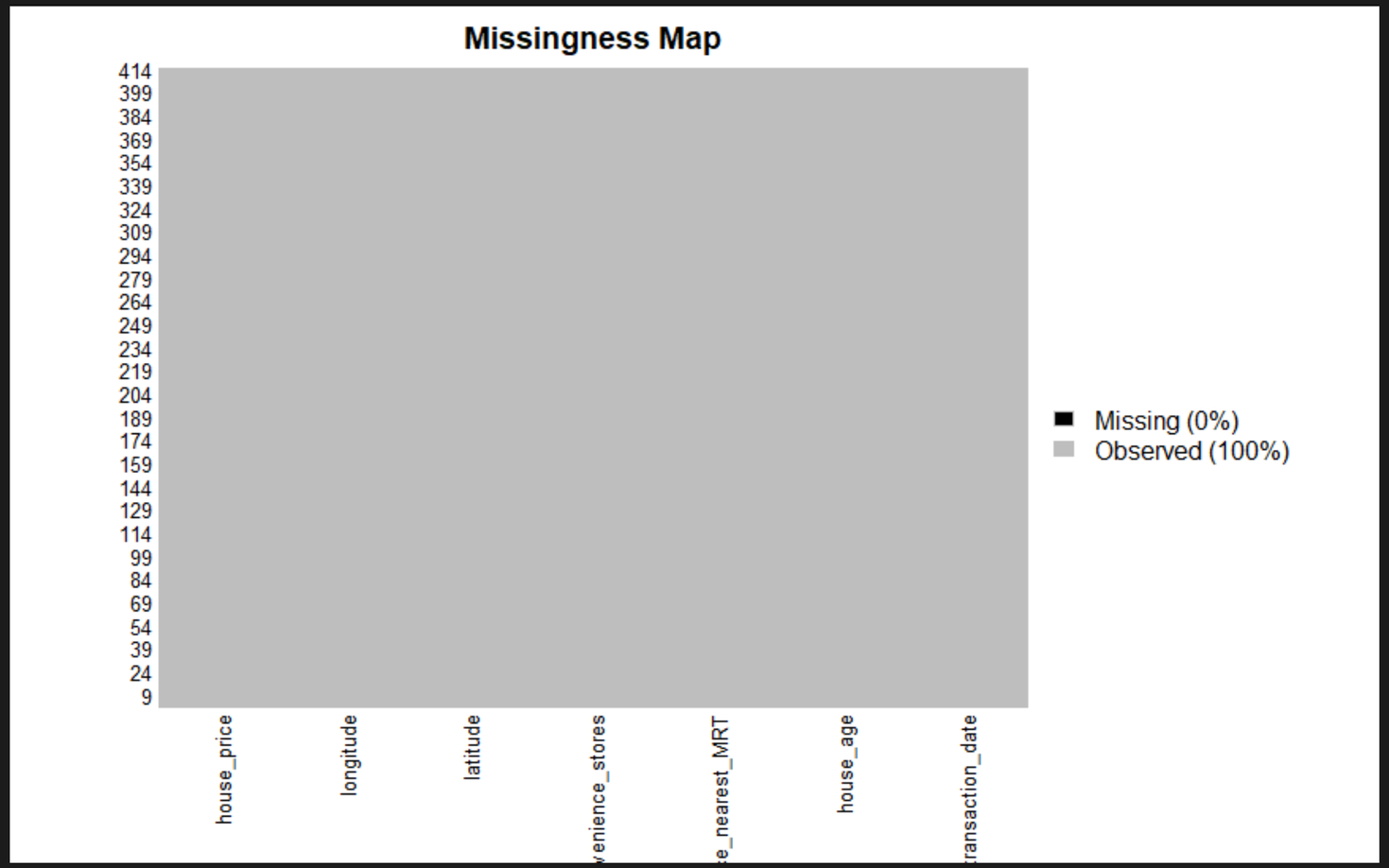
```
ggplot(realestate, aes(x=house_price)) +
geom_boxplot(color = "#1b1b1b")
```

Real Estate Price Prediction

Preprocessing

Real Estate
Price
Prediction

Missing Values



#Missing values

```
install.packages("Amelia")
library(Amelia)
missmap(realestate, col=c("black", "grey"))
```

Delete unnecessary columns

Before

```
> str (realestate)
'data.frame':   414 obs. of  8 variables:
 $ No                        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ transaction_date          : num  2013 2013 2014 2014 2013 ...
 $ house_age                 : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
 $ distance_nearest_MRT      : num  84.9 306.6 562 562 390.6 ...
 $ number_convenience_stores: int  10 9 5 5 5 3 7 6 1 3 ...
 $ latitude                  : num  25 25 25 25 25 ...
 $ longitude                 : num  122 122 122 122 122 ...
 $ house_price               : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ...
```

After

```
> realestate <- select(realestate, -No)
> str (realestate)
'data.frame':   414 obs. of  7 variables:
 $ transaction_date          : num  2013 2013 2014 2014 2013 ...
 $ house_age                 : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
 $ distance_nearest_MRT      : num  84.9 306.6 562 562 390.6 ...
 $ number_convenience_stores: int  10 9 5 5 5 3 7 6 1 3 ...
 $ latitude                  : num  25 25 25 25 25 ...
 $ longitude                 : num  122 122 122 122 122 ...
 $ house_price               : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ...
```

#Deleting columns

```
install.packages("dplyr")
library(dplyr)
realestate <- select(realestate, -No)
str (realestate)
```

Real Estate Price Prediction

Normalization

NN works best when the input data are scaled to a narrow range around zero.

#Normalization
normalize <- function(x) { return((x - min(x)) / (max(x)- min (x)))}

realestate_norm <- as.data.frame (lapply(realestate, normalize))

#Results of normalization
summary(realestate_norm $ house_price)

#In comparison of the original min / max
summary(realestate $ house_price)

```
> #Results of normalization
> summary(realestate_norm $ house_price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.1829  0.2807  0.2764  0.3549  1.0000
> #In comparison of the original minimum and maximum
> summary(realestate $ house_price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   7.60   27.70   38.45   37.98   46.60  117.50
>
```

Real Estate Price Prediction

Training and testing sets

We will divide the data into 75 percent for the training set and 25 percent for the testing set (Based in the observations)

```
> head(realestate_train)
  transaction_date house_age distance_nearest_MRT number_convenience_stores  latitude longitude house_price
1        0.2729258 0.7305936          0.009512672                       1.0 0.6169413 0.7193228   0.2757052
2        0.2729258 0.4452055          0.043809391                       0.9 0.5849491 0.7114514   0.3148317
3        1.0000000 0.3036530          0.083315051                       0.5 0.6712312 0.7588958   0.3612375
4        0.9093886 0.3036530          0.083315051                       0.5 0.6712312 0.7588958   0.4294813
5        0.1812227 0.1141553          0.056799089                       0.5 0.5731944 0.7431529   0.3230209
6        0.0000000 0.1621005          0.332833348                       0.3 0.3754241 0.4206383   0.2229299
```

```
> head(realestate_test)
    transaction_date  house_age distance_nearest_MRT number_convenience_stores  latitude longitude house_price
312        0.5458515 0.48630137          0.07957356                       0.4 0.5111488 0.6966789   0.3148317
313        1.0000000 0.80821918          0.04565551                       0.9 0.4682501 0.7241751   0.6405823
314        0.7270742 0.18949772          0.01259580                       0.5 0.4201406 0.7239595   0.3202912
315        0.6364629 0.08447489          0.08578650                       0.6 0.4840039 0.7945870   0.3093722
316        0.4541485 0.35616438          0.26807814                       0.2 0.6196074 0.4808066   0.1792539
317        0.6364629 0.30365297          0.03515249                       0.7 0.4119001 0.7487600   0.3130118
```

realestate_train <- realestate_norm[1:311,]
realestate_test <- realestate_norm[312:414,]

Processing and results

Real Estate
Price
Prediction

Real Estate
Price
Prediction

Neural networks

Neural Network is an information processing paradigm that is inspired by the human nervous system. As in the Human Nervous system, we have Biological neurons in the same way in Neural networks we have Artificial Neurons which is a Mathematical Function that originates from biological neurons.

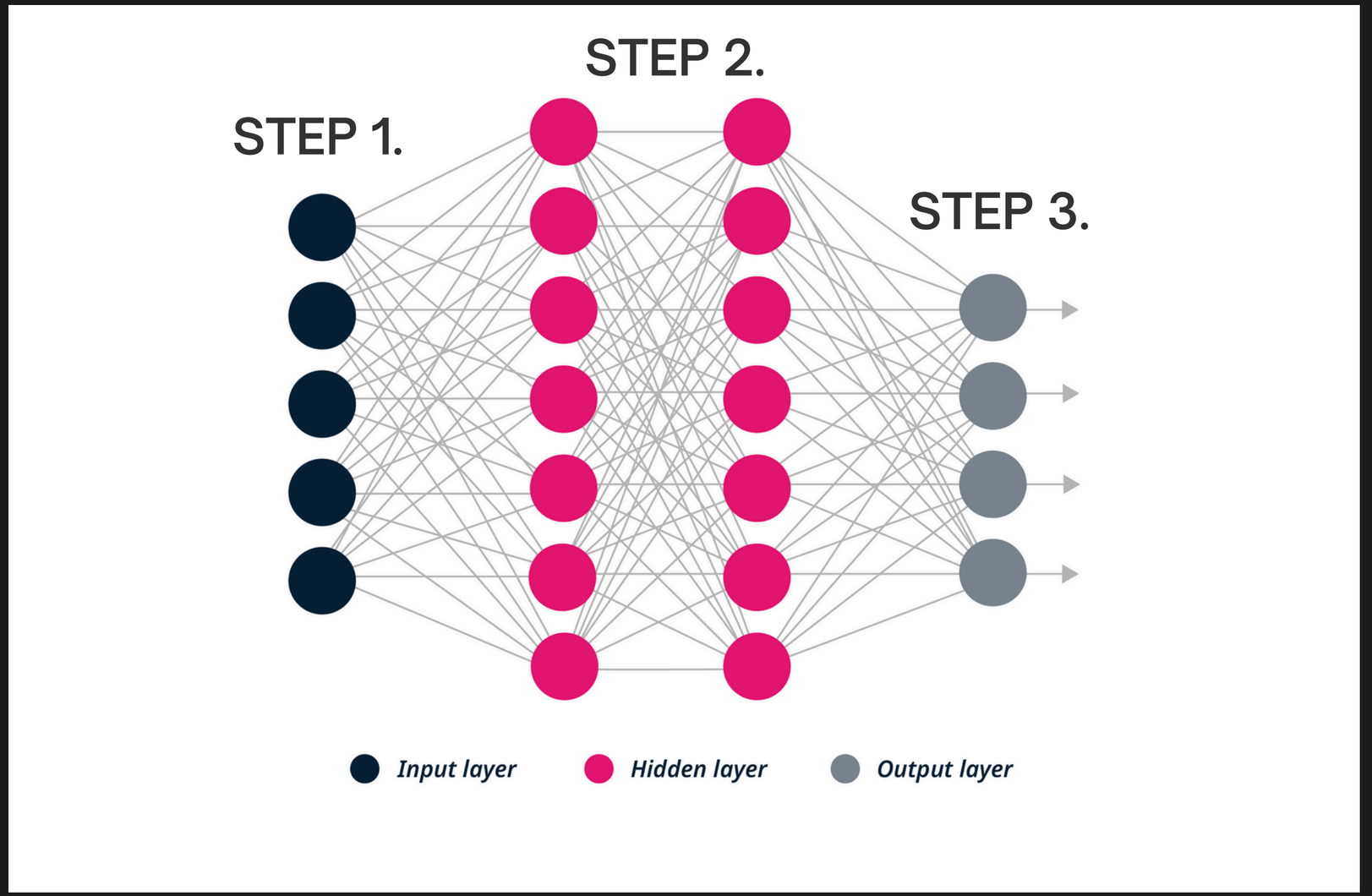
The artificial neural network (ANN) assimilates data in the same way the human brain processes information. The brain’s neurons process information in the form of electric signals. External information, or stimuli, is received and processed, and the brain then produces an output.

Similarly, neural networks reflect the behavior of the human brain, allowing computer programs to recognize patterns and solve common problems in the fields of artificial intelligence (AI), machine learning, and deep learning.

This process mimicry is achieved in three steps:

- **Step 1**: ANNs receive input through several processors that operate simultaneously and are arranged in tiers
- **Step 2**: The first tier receives the raw input data, which it then processes through interconnected nodes that have their own sets of knowledge and rules
- **Step 3**: The processor then passes it on to the next tier as output. Each successive tier of processors and nodes receives the output from the tier preceding it and processes it further. This refines the data incrementally rather than having to process the raw data anew every time.

Real Estate
Price
Prediction



Real Estate
Price
Prediction

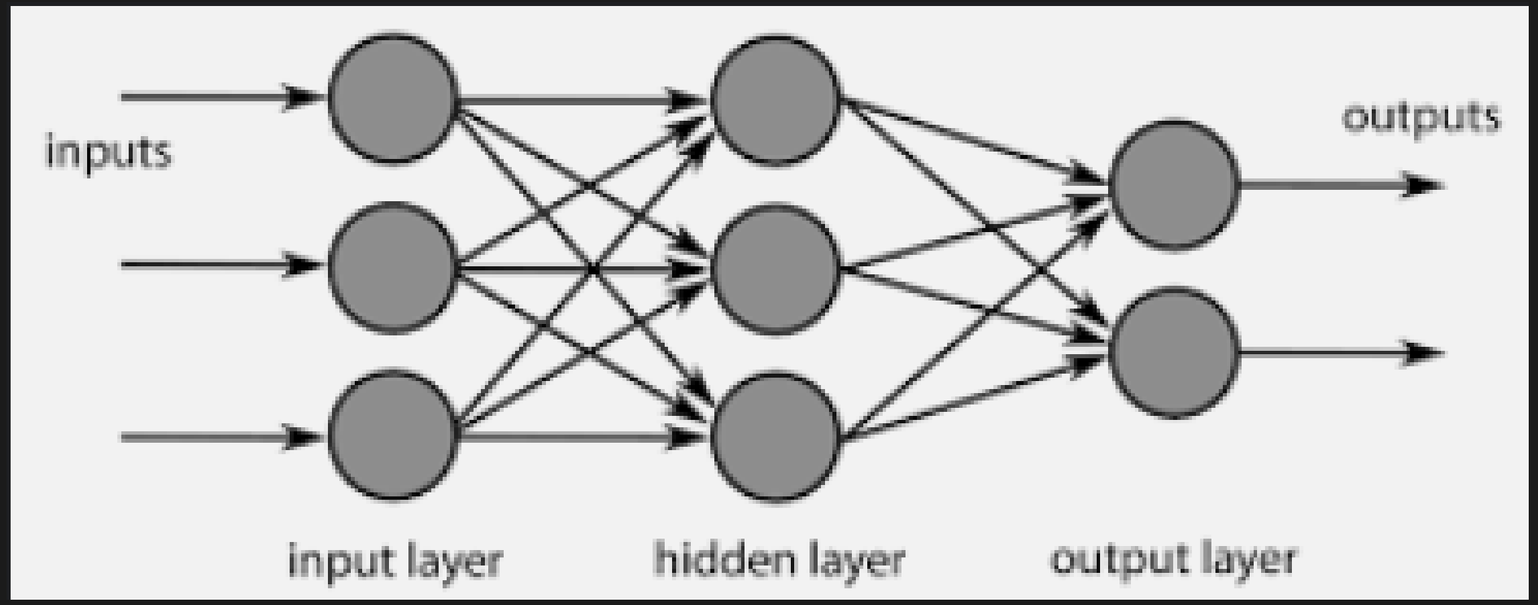
In Data Mining

Neural Network Method is used For Classification, Clustering, Feature mining, prediction, and pattern recognition. McCulloch-Pitts model is considered to be the first neural network and the Hebbian learning rule is one of the earliest and simplest learning rules for the neural network. The neural network model can be broadly divided into the following three types:

- Feed-Forward Neural Networks
  - Information moves in only one direction (forward).
- Feedback Neural Network
  - Information can travel in both directions in a feedback network.
- Self Organization Neural Network
  - It is used to produce a low-dimensional (typically two-dimensional) representation of a higher-dimensional data set while preserving the topological structure of the data.

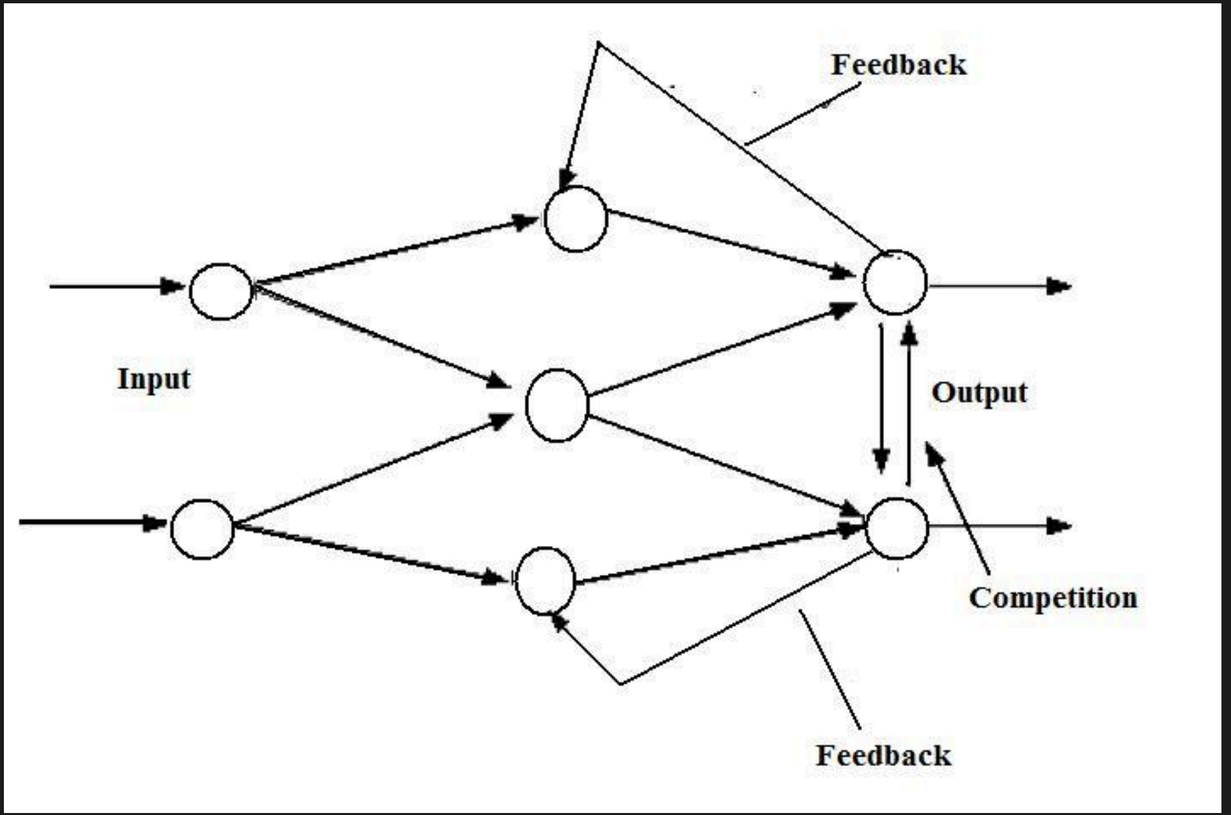
Real Estate
Price
Prediction

Feed-Forward Neural Networks



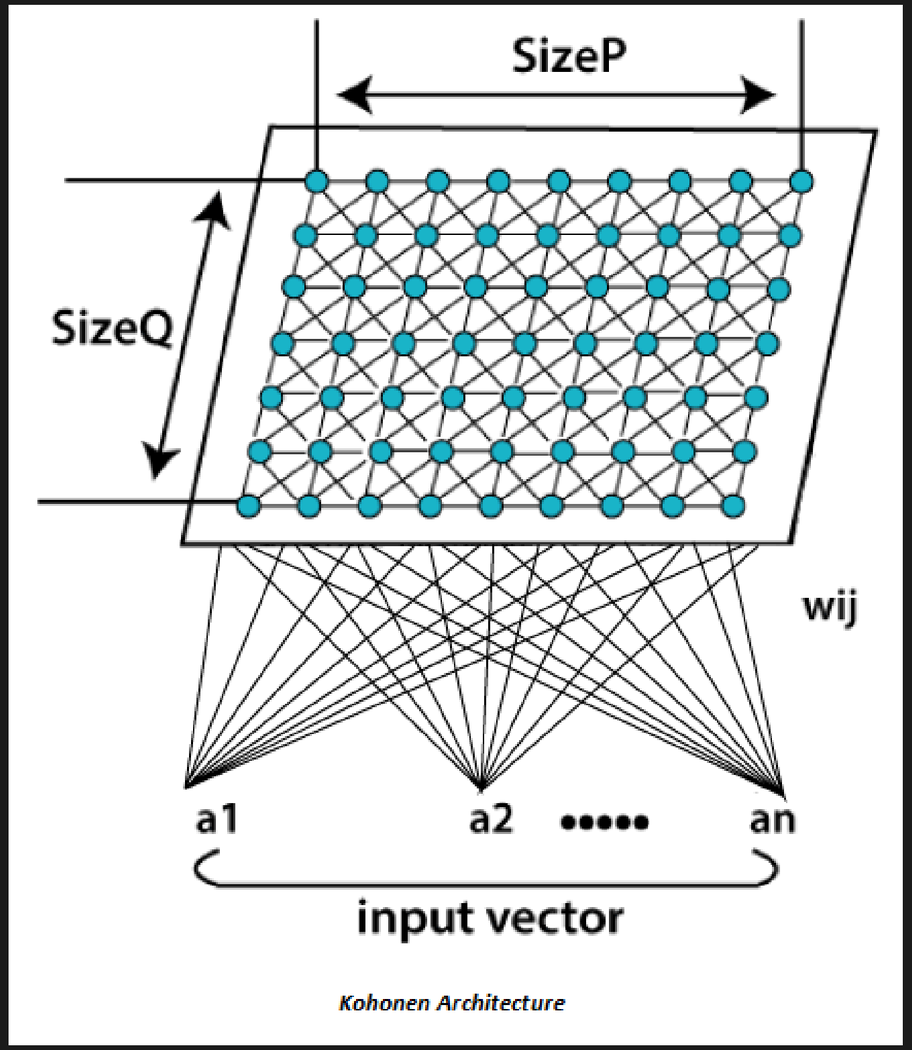
Real Estate
Price
Prediction

Feedback Neural Network



Real Estate
Price
Prediction

Self Organization Neural Network



Kohonen Architecture

Real Estate
Price
Prediction

Some uses

Neural networks help in mining large amounts of data in various sectors such as retail, banking (Fraud detection), bioinformatics(genome sequencing), etc. Data Mining uses Neural networks to harvest information from large datasets from data warehousing organizations. Which helps the user in decision making.

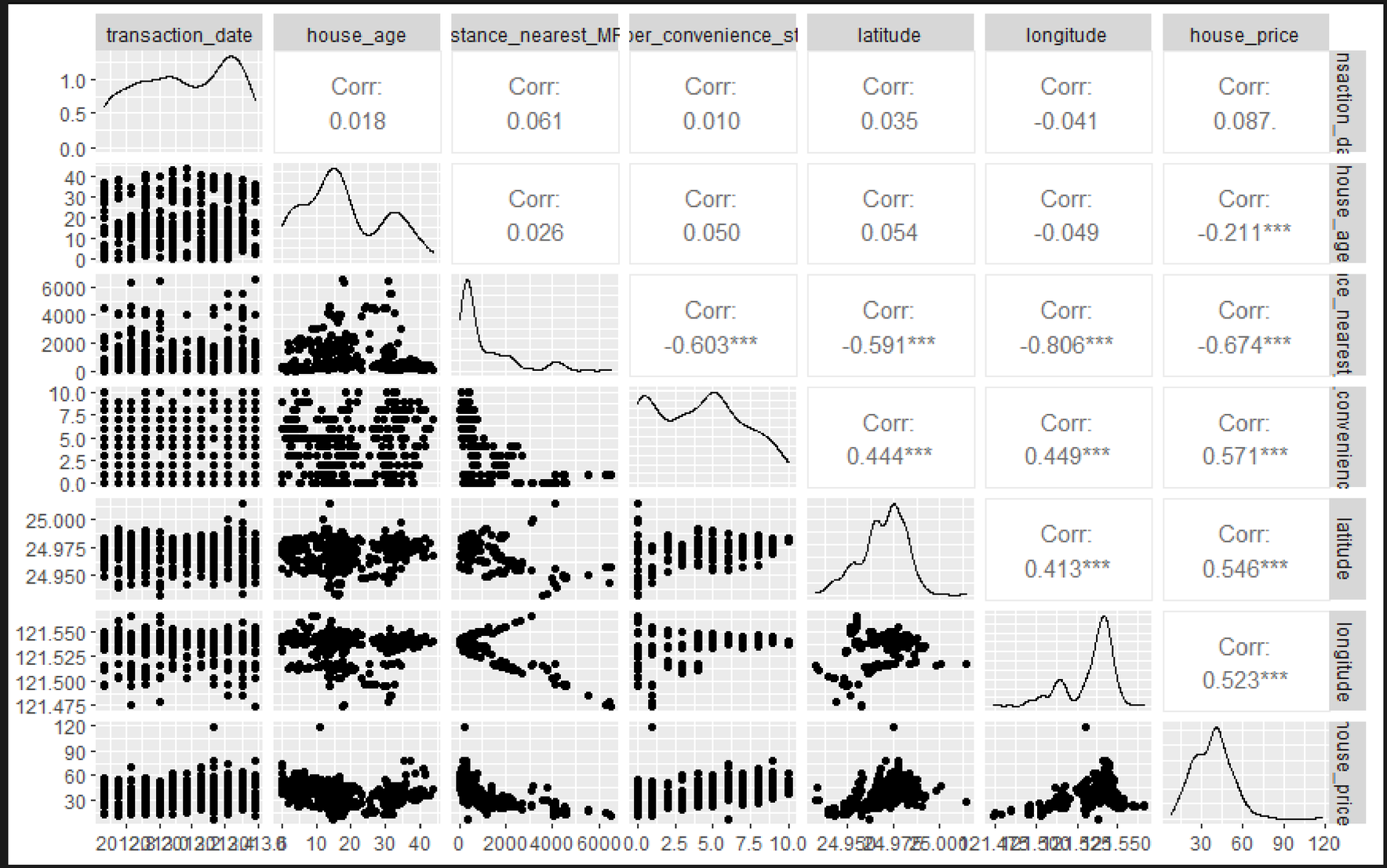
Some of the Applications of Neural Network In Data Mining are given below:

- **Fraud Detection**: The problem is going to increase in today's modern world because of the advancement of technology, which makes fraud relatively easy to commit but on the other hand technology also helps is fraud detection and in this neural network help us a lot in detecting fraud.
- **Healthcare**: In healthcare, Neural Network helps us in Diagnosing diseases, as we know that there are many diseases and there are large datasets having records of these diseases. With neural networks and these records, we diagnosed these diseases in the early stage as soon as possible.

Pairs plot

Correlation between the variables in the dataset.

ggpairs(realestate)



Real Estate Price Prediction

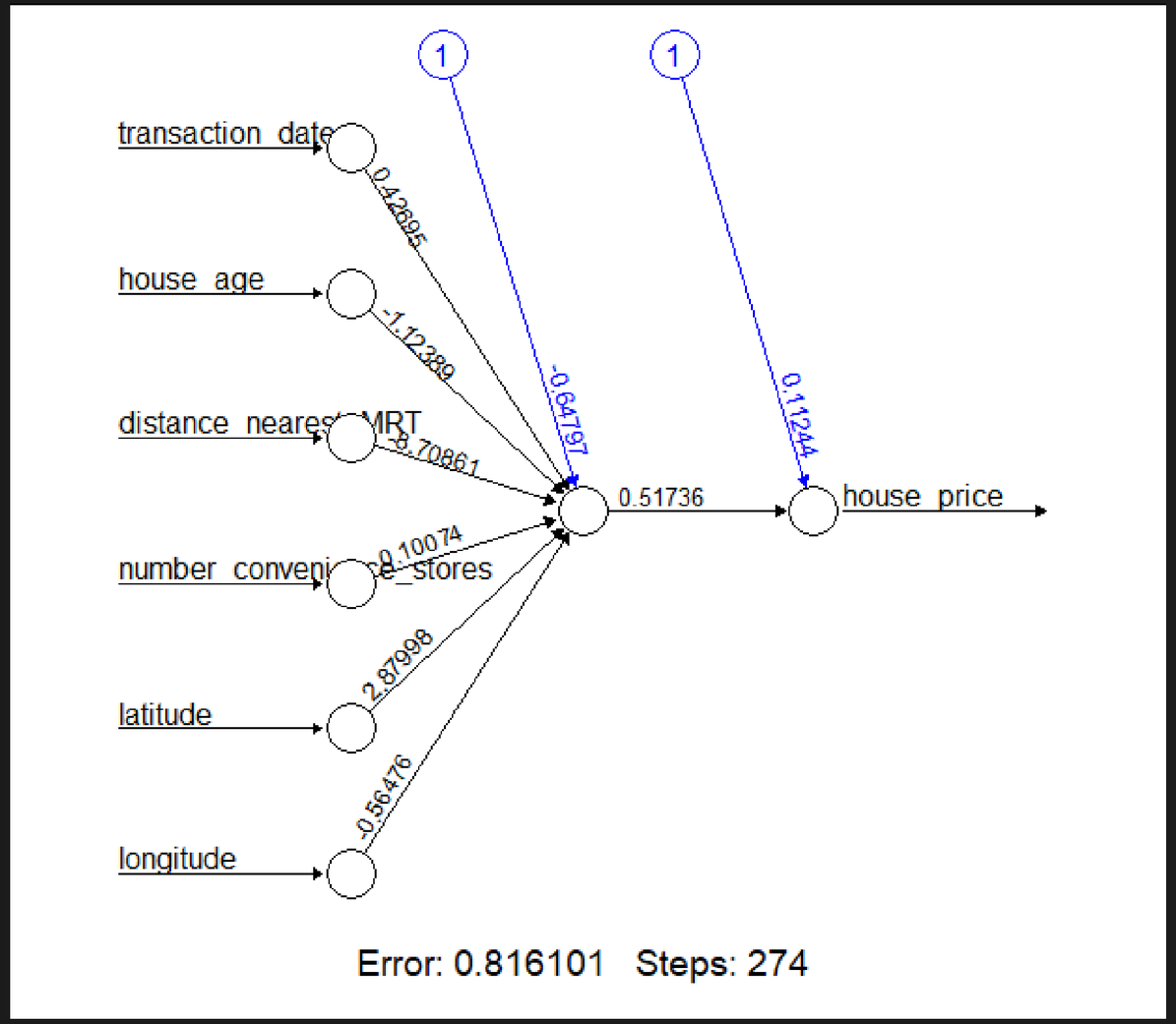
NN In R - Create the model

```r
#Using neuralnet
install.packages("neuralnet")
library(neuralnet)

#Begin by training the simplest multilayer feedforward network with only a single node
set.seed(12345)# To get repeatable results
realestate_model<-neuralnet( house_price ~ transaction_date + house_age + distance_nearest_MRT  +
                             number_convenience_stores + latitude + longitude,
                             data = realestate_train, hidden = 1)
```

Real Estate
Price
Prediction

NN In R - Network topology



Real Estate Price Prediction

NN In R - Evaluating performance

```r
#compute() function generate predictions
#returns two components neurons $net.result which stores the model predicted values.
model_results <- compute (realestate_model, realestate_test[1:6])

predicted_price <- model_results $ net.result

#Correlation between the predicted value and the true value
#HIGHER CORRELATION THE BETTER
cor( predicted_price, realestate_test $ house_price) [,1]
```

Result

```
> cor( predicted_price, realestate_test $ house_price) [,1]
[1] 0.7740833
```

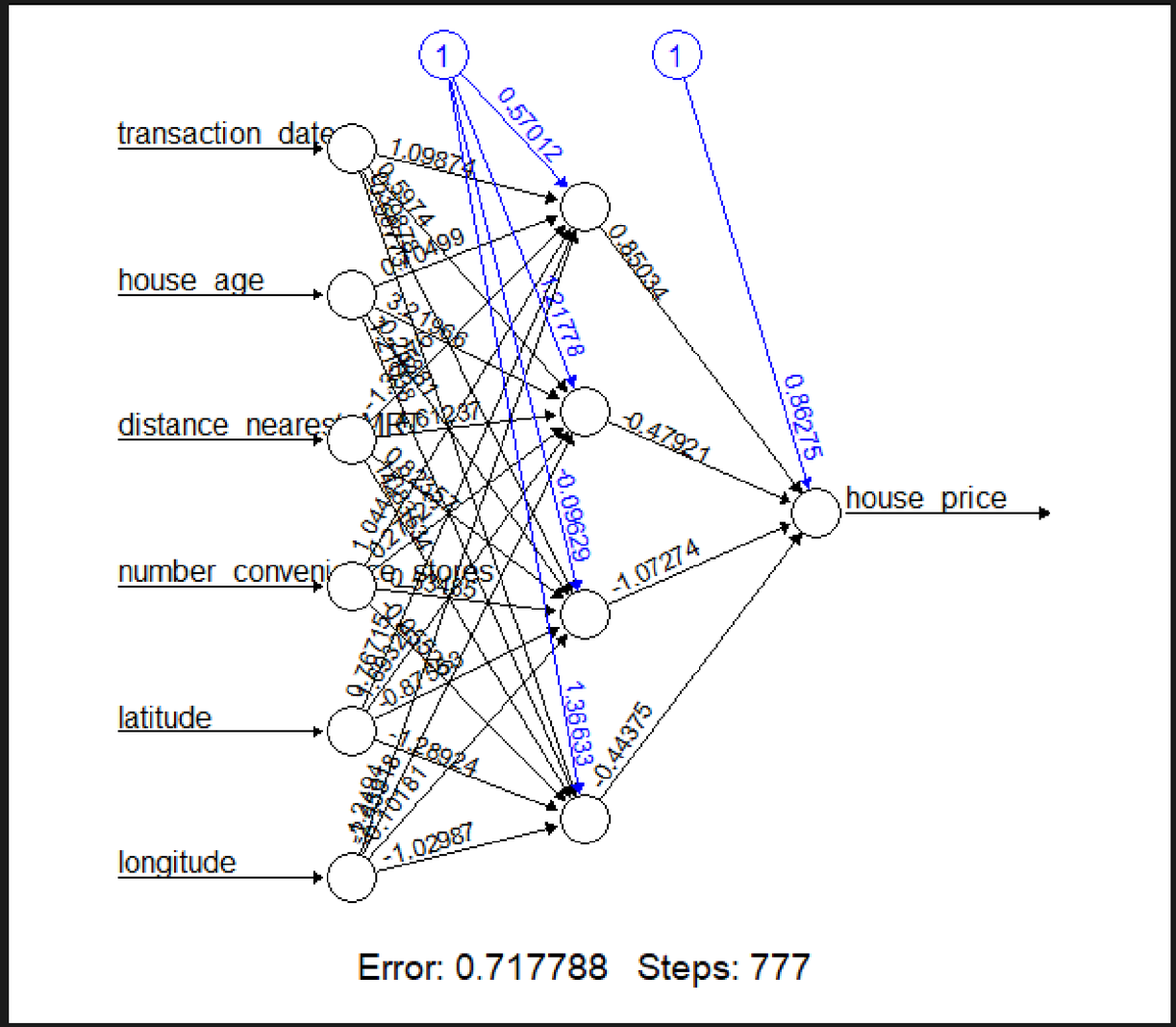
Real Estate Price Prediction

NN In R - Improving the model

```
#Begin by training the simplest multilayer feedforward network with only a four nodes
realestate_model2<-neuralnet( house_price ~ transaction_date + house_age + distance_nearest_MRT  +
                              number_convenience_stores + latitude + longitude,
                              data = realestate_train, hidden = 4)
```

Real Estate
Price
Prediction

NN In R - Network topology



Real Estate Price Prediction

NN In R - Evaluating performance

```r
#compute() function generate predictions
#returns two components neurons $net.result which stores the model predicted values.
model_results2 <- compute (realestate_model2, realestate_test[1:6])

predicted_price2 <- model_results2 $ net.result

#Correlation between the predicted value and the true value
#HIGHER CORRELATION THE BETTER
cor(predicted_price2, realestate_test $ house_price)
```

Result

```
> #Correlation between the predicted value and the true value
> #HIGHER CORRELATION THE BETTER
> cor(predicted_price2, realestate_test $ house_price)
          [,1]
[1,] 0.8034113
```

Real Estate Price Prediction

NN In R - 5 nodes

```r
#Begin by training the simplest multilayer feedforward network with only a five nodes
realestate_model2<-neuralnet( house_price ~ transaction_date + house_age + distance_nearest_MRT  +
                              number_convenience_stores + latitude + longitude,
                              data = realestate_train, hidden = 5)

#Visualize the network topology
plot (realestate_model2, rep="best")

#compute() function generate predictions
#returns two components neurons $net.result which stores the model predicted values.
model_results2 <- compute (realestate_model2, realestate_test[1:6])

predicted_price2 <- model_results2 $ net.result

#Correlation between the predicted value and the true value
#HIGHER CORRELATION THE BETTER
cor(predicted_price2, realestate_test $ house_price)
```

Result

```
> #HIGHER CORRELATION THE BETTER
> cor(predicted_price2, realestate_test $ house_price)
           [,1]
[1,] 0.8109269
```

Real Estate Price Prediction

Classification outputs

In this study, we have a single output which is dependent variable "*house price of unit area* ".

Conclusions and limitations

Real Estate Price Prediction

Real Estate
Price
Prediction

Does the study generalize to other domains?

Absolutely yes, as we saw earlier, there are many applications for ANN in many industries, health, financial, image recognition and so many more that the possibilities are endless.

Limitations

We depend on the hardware of the machine running the ANN, while we were making this study, several times our computers crashed because we selected more layers than we needed and with ANN it's a bit difficult to explain how it managed to give the solution.

Advantages

Being able to try to predict such things it's useful and we can help people decide things or in this case, being able to know the price of a house.

Real Estate
Price
Prediction

What would you do to improve your analysis?

We thing that the data might be small, more data could've improve our analysis and more independent variables.

What is the main weakness of your project?

Probably that its accuracy is around 80%, is not that bad but could be better, and in terms of the dataset, the dataset it's fairly old and might be obsolete.

Bibliography

- https://www.getsmarter.com/blog/career-advice/how-artificial-neural-networks-can-be-used-for-data-mining/
- https://www.geeksforgeeks.org/how-neural-networks-can-be-used-for-data-mining/
- https://rpubs.com/julianhatwell/annr
- https://www.kaggle.com/datasets/quantbruce/real-estate-price-prediction
- https://www.investopedia.com/terms/r/realestate.asp
- https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set
- https://ggobi.github.io/ggally/reference/ggpairs.html