

**Statistics department**

**Introduction to data science (SE305)**

**English section**

**Final project**

**Moscow Housing Price Dataset**

**Presented to:**

**Dr. Sara Osama**

**Submitted by:**

|  |  |
| --- | --- |
| **Nada Hamouda Mohamed Jamiel** | **5210898** |

**Introduction**

The property market can be unpredictable. Our analysis aims to study how various variables affect apartment prices and how they interact with each other. Using a dataset, retrieved from [here](https://www.kaggle.com/datasets/egorkainov/moscow-housing-price-dataset), on housing prices and apartment characteristics collected from Moscow and the Moscow Oblast region in November 2023, we examined how features like apartment type, floor count, proximity to metro stations, and other factors impact prices.

Our analysis focuses on three research questions:

1. What factors significantly influence whether the living area is large or small? How accurately can Area, Kitchen area, and Number of rooms predict living area size?
2. What are the variables that significantly affect apartment prices?
3. What variables can classify apartments as high or low value?

By applying statistical methods, we aim to state clear answers to the raised questions.

Data description:

Our dataset contains 22767 observations and 12 variables describing different aspects of the apartment as follows:

Variables:

1. Quantitative

* Price: The price of the apartment (in country’s currency).
* Minutes to metro: the walking time needed from the apartment to the nearest metro station (in minutes).
* Number of rooms: The number of rooms in the apartment.
* Area: The total area of the apartment (in square meters).
* Living area: the area of the living room/area in the apartment (in square meters).
* Kitchen area: The area of the kitchen (in square meters).
* Floor: the apartment’s floor number.
* Number of floors: The number of floors in the building of the apartment.

1. Categorical

* Apartment type: The type of apartment (secondary or new building).
* Metro station: The name of the nearest metro station.
* Region: The region where the apartment is located (Moscow or Moscow Oblast).
* Renovation: The level of renovation of the apartment (Designer, Cosmetic, European-style, without renovation).

Data preparation & manipulation

* Started by exploring the variables and taking a sample of 70% train data (and 30% to be tested).
* Correctly assigning variables to their classes.
* Exploring the existence of any missing values and none was found.
* Variable Region was removed, we didn’t use it in our analysis since we were not interested in the effect of the region on the price (if excited).
* Variable Number of rooms where categorized into: no rooms, single, two, three and more, for easier use in the analysis.
* The Price where changed to be in thousands to make it easier to read.
* The Metro station variable where changed to be in only four categories (north stations, south stations, east stations and west stations) using this [metro stations map](https://news.metro.ru/sc_lat.html), for easier interpretation.
* We checked for the existence of outliers, and a function was created to replace any outlier smaller than the lower limit with the 1st quantile and any outlier larger than the higher limit with the 3rd quantile.
* A subset from the data was created for the quantitative variables only called (quan\_data).
* We converted the living area into 2 categories (binary variable) for the logistic model.
* For the decision tree, we converted the price into a categorical variable.

**Descriptive analysis**

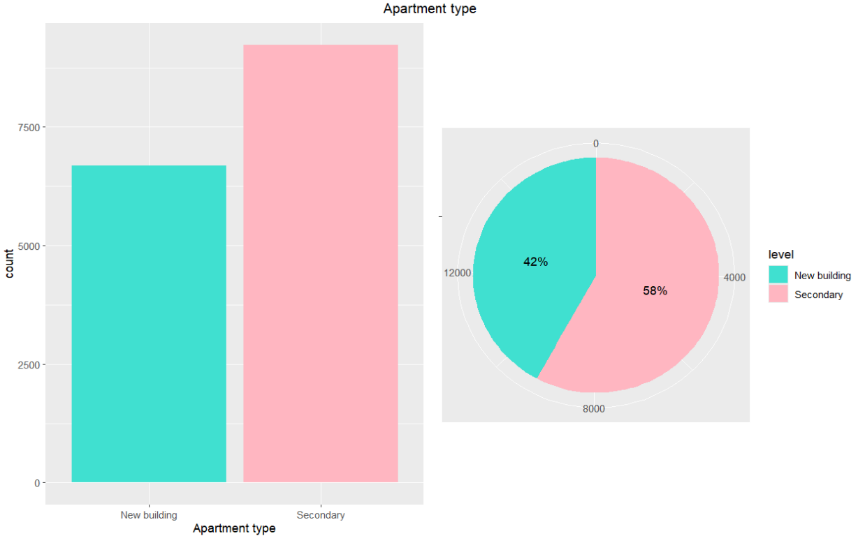
For categorical variables:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable | Apartment type |  | Metro station |  | Number of rooms |  | Renovation |
| New building: | 6682 | East : | 2869 | No rooms: | 2600 | Cosmetic: | 9006 |
| Secondary : | 9233 | North: | 3672 | single: | 3628 | Designer: | 2128 |
|  |  | South: | 6502 | two: | 4461 | European-style: | 2577 |
|  |  | West : | 2872 | three: | 2685 | Without renovation: | 2204 |
|  |  |  |  | more: | 2541 |  |  |

Table 1: summary statistic

The dataset reveals that the majority of apartments are secondary, with a significant proportion located near metro stations in the South region, followed by the North. Two-room apartments are the most common, although single-room and three-room options are also notable. Renovations are predominantly cosmetic, while designer and European-style renovations are less common, and a smaller portion of apartments remain without renovations.

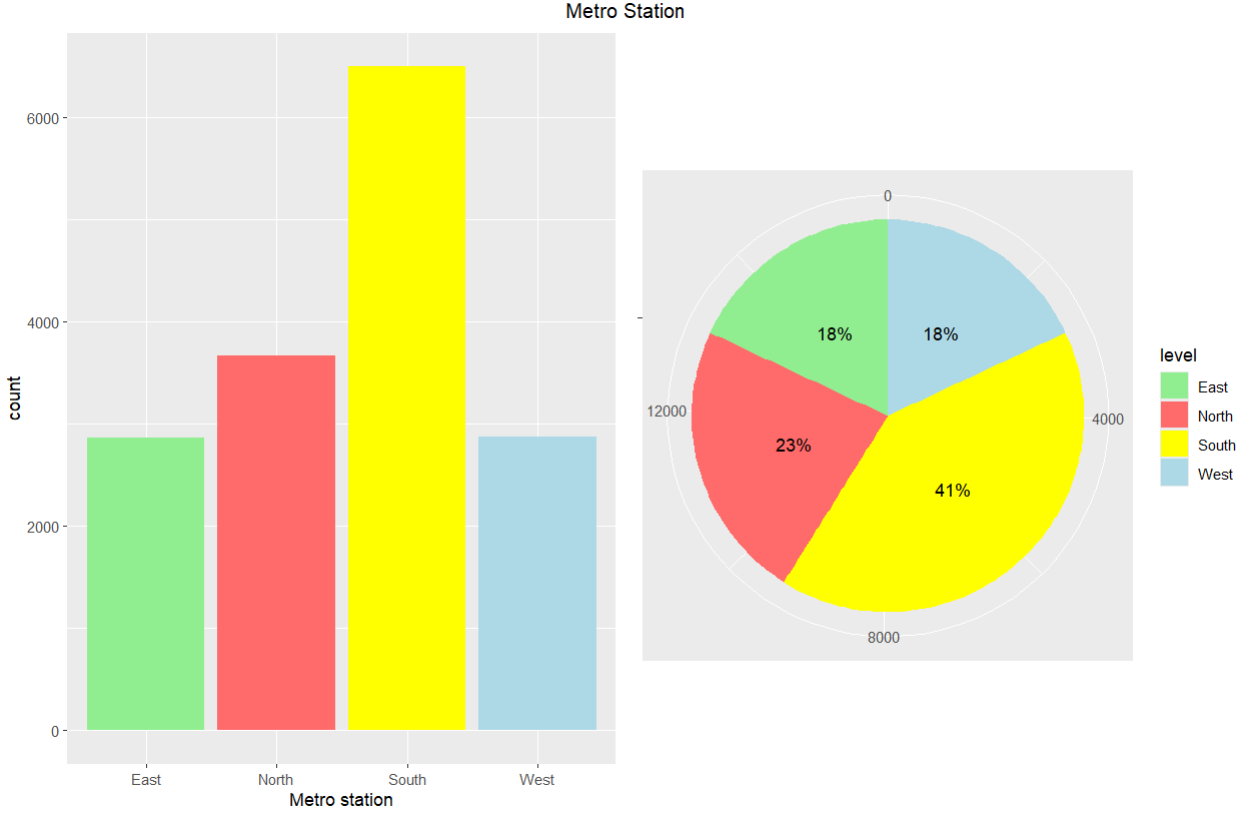
Figure 1

****

Apartment type:

The pie chart displays the percentage distribution, indicating that 58% of the apartments are Secondary, while 42% are new building. The bar chart shows that Secondary apartments have a higher count compared to new building apartments.

Figure 2

****

Metro station:

The bar chart shows that the South region has the highest number of metro stations, followed by the North, while the East and West have fewer stations. The pie chart is showing that 41% of stations are in the South, which is the majority of stations, 23% in the North, and both the East and West have 18% each.

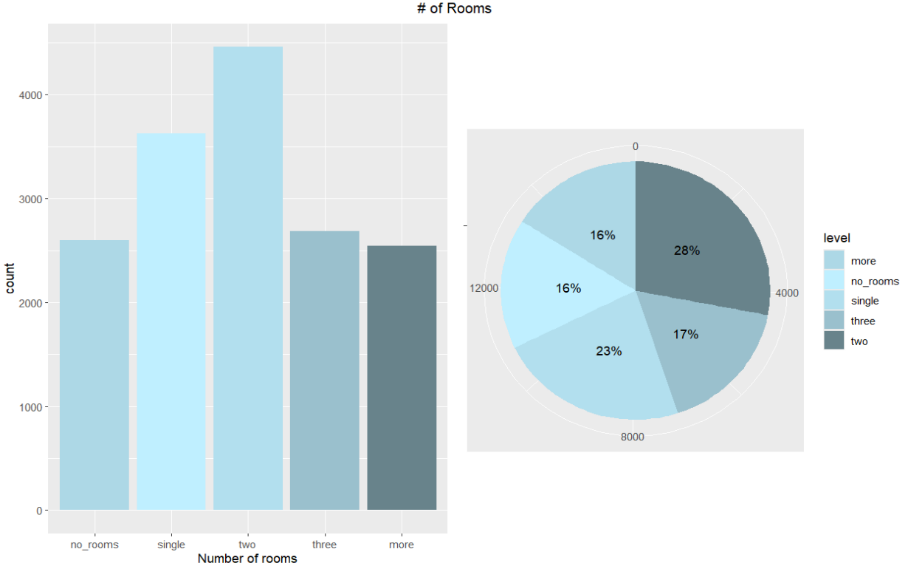
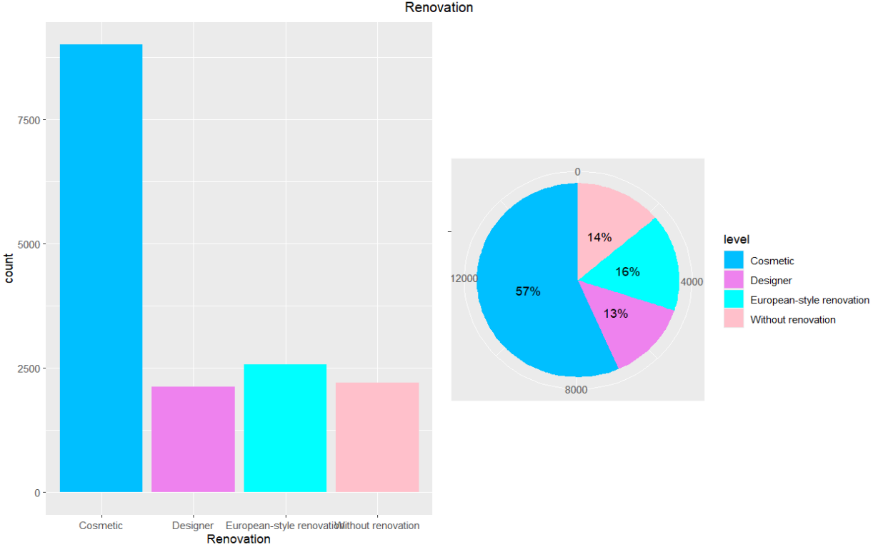
****

Figure 3

Number of rooms:

The bar chart reveals that apartments with two rooms are the most common, followed by single room. The other categories, including three rooms, no rooms, and more than three rooms, have smaller counts. The pie chart shows that: the majority of apartments have two rooms with 28% of apartments, 23% have single room, 17% have three rooms, and apartments with no rooms or more than three rooms are 16% each.

Figure 4

****

Renovation:

The bar chart shows that apartments with Cosmetic renovations are the most common, with significantly higher percentages than the other types. The pie chart provides that 57% of apartments have Cosmetic renovations, 16% have European-style renovations, 14% are without renovation, and only 13% of apartments have Designer renovations.

For quantitative variables:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable | Price | Minutes to metro | Area | Living area | Kitchen area | Floor | Number of floors |
| Min. | 1150 | 0.00 | 6.00 | 2.00 | 1.00 | 1.000 | 1.00 |
| 1st Qu. | 7100 | 7.00 | 37.40 | 17.50 | 8.60 | 4.000 | 11.00 |
| Median | 11393 | 11.00 | 53.00 | 28.50 | 10.60 | 8.000 | 16.00 |
| Mean | 14868 | 11.53 | 56.04 | 30.42 | 11.31 | 8.586 | 15.21 |
| 3rd Qu. | 24800 | 15.00 | 77.00 | 43.10 | 14.50 | 13.000 | 20.00 |
| Max. | 51131 | 27.00 | 136.40 | 81.50 | 23.30 | 26.000 | 33.00 |
| Standard deviation | 9939.276096 | 5.279249429 | 25.41261742 | 15.60241717 | 4.333909266 | 5.869655423 | 6.200591611 |
| trimmed | 13633.56754 | 11.31249509 | 54.35539307 | 28.94592005 | 11.09421189 | 8.041388518 | 15.09754182 |
| mad | 8139.210097 | 5.9304 | 26.286498 | 17.7912 | 3.85476 | 7.413 | 5.9304 |
| range | 49981 | 27 | 130.4 | 79.5 | 22.3 | 25 | 32 |
| skew | 1.170094119 | 0.385206441 | 0.636228819 | 0.791885883 | 0.427744452 | 0.684751709 | 0.195375758 |
| kurtosis | 1.091545883 | -0.37754402 | 0.232881948 | 0.304153059 | -0.028020879 | -0.259801088 | -0.21352873 |
| se | 78.78643203 | 0.041847437 | 0.201440169 | 0.123676892 | 0.034353935 | 0.046527454 | 0.049150711 |
| unique | 6663 | 28 | 2031 | 721 | 218 | 26 | 33 |
| Mean CI | 14713.964  15022.824 | 11.45  11.62 | 55.65  56.43 | 30.18  30.66 | 11.24  11.38 | 8.49  8.68 | 15.11  15.30 |
| IQR | 17699.85 | 8 | 39.6 | 25.6 | 5.9 | 9 | 9 |

Table 2: Summary statistic

The table summarizes key statistics for quantitative housing variables, highlighting variability and distribution patterns. Apartment’s prices range from 1150 thousands to 51131 thousands, with a median of 11393 thousands and an average price of 14868 thousands, showing positive skewness due to high-priced outliers. Areas and living spaces show similar trends, with areas ranging from 6 to 136 square meters and an average of approximately 56 square meters, while living spaces range from 2 to 81.5 square meters with a mean living area of 30.42 square meters. Both variables have moderate variability and positive skewness, indicating smaller apartments. Kitchen areas are ranging from 1 to 23 square meters with a mean of 11.31 square meters, and are slightly positively skewed. The number of floors in a building is ranging from 1 to 33 floors. However, the observed apartments are located from 1st to 26th floor. The waking time needed from the apartment to the nearest metro station ranges from 0 to 27 minutes, with a mean of 11.53 minutes and slight left skew, suggesting most apartments are close to metro stations.

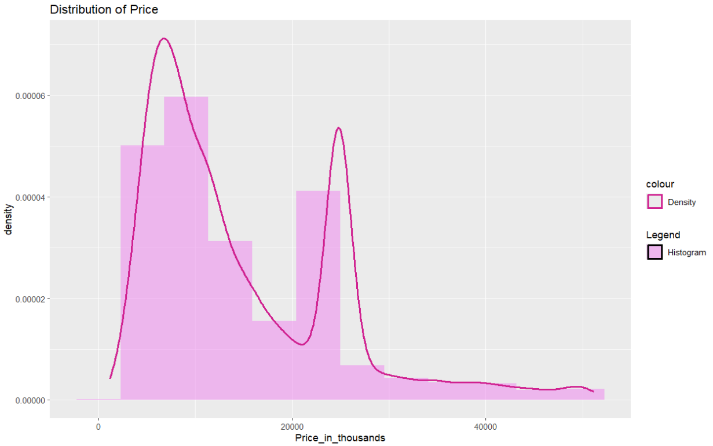
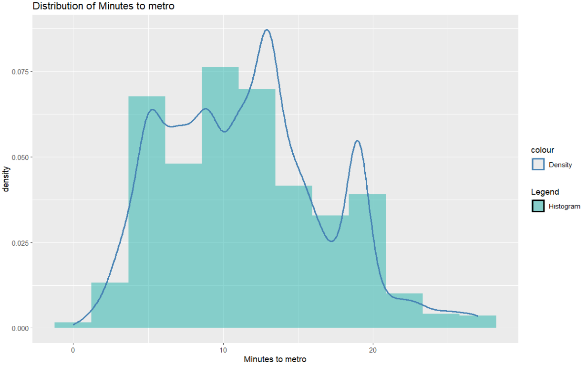


Figure 5

Price:

The chart combines a histogram that shows the frequency of prices across different ranges and a density curve provides a smoothed representation of the price distribution of apartments prices (in thousands). The data shows two peaks, one for apartments priced lower (around 10000) and another for apartments priced higher (around 25000), where the higher peak is for the higher priced apartments.

Figure 6



Minutes to metro:

The histogram and the density curve displays the frequency of different walking time intervals needed to reach the nearest metro station, showing there distribution. The data shows a peak around 10 minutes, indicating that most apartments are 10 minutes away from the nearest metro. The peak is not that sharp showing a higher variability in the data.

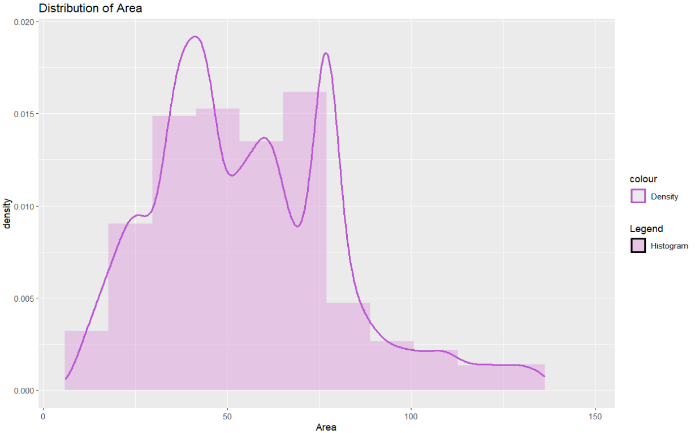
****

Figure 7

Area:

The histogram shows the frequency of different area ranges, while the density curve provides a smoothed distribution. The data displays two peaks: one around 50 square meters and another slightly above 75 square meters, this suggests two common size categories for apartments. The distribution shows that apartments of areas larger than 100 square meters are less common.

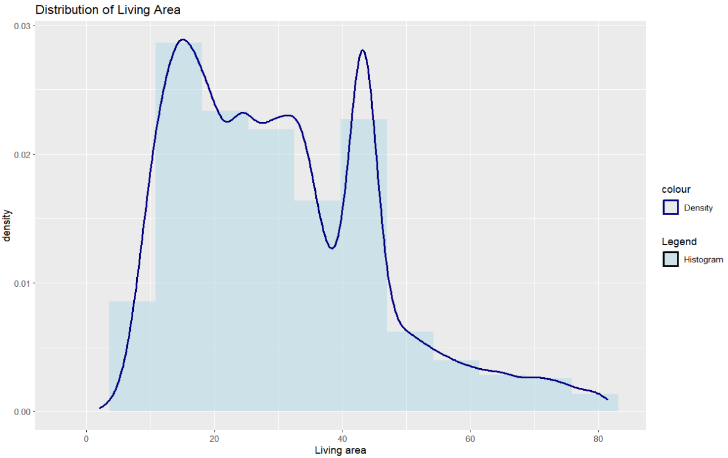
****

Figure 8

Living area:

The histogram and the density curve represents the frequency of different ranges of living areas, showing there distribution. The distribution is peaking around 25 square meters and near 40 square meters, this suggests two common size categories for living areas. Generally, the Living area distribution is very close in shape to that of the total area of the apartment.

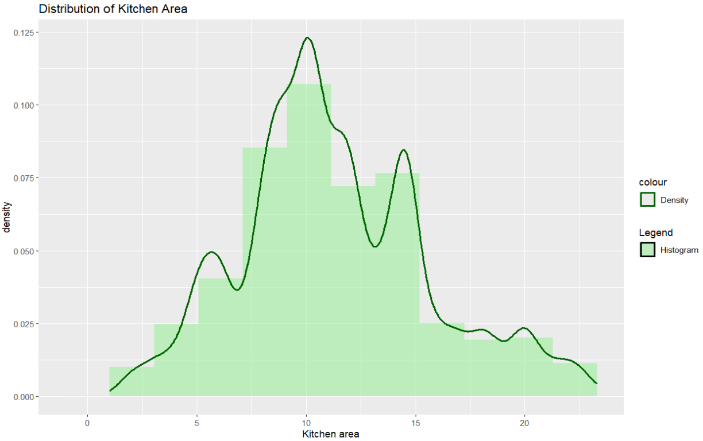
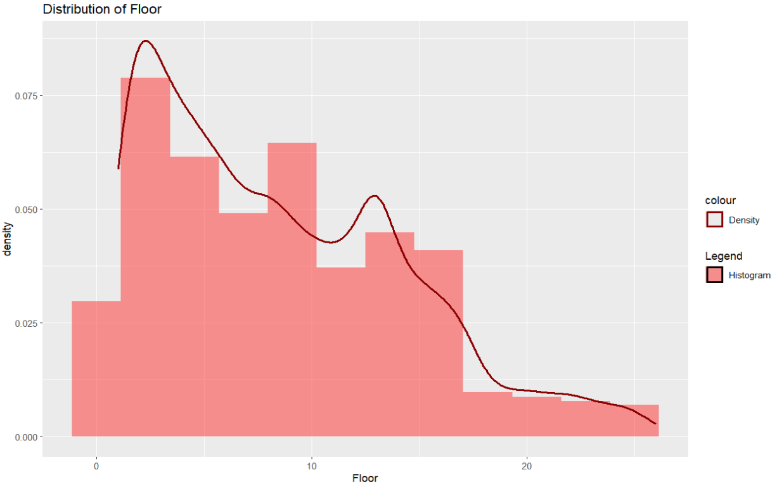


Figure 9

Kitchen area:

The histogram and the density curve shows the frequency and the smoothed distribution of different kitchen area ranges. The distribution is showing a peak around 10 square meters, indicating this is the most common kitchen size. The frequency declines for kitchen areas larger than 15 square meters and less than 5 square meters.

Figure 10

****

Floor:

The distribution of apartments across floors. The figure shows that most apartments are located on lower floors, with a sharp peak around the 1st or 2nd floor, and a gradual decrease in frequency as the floor number increases. The right tale of the distribution indicates that apartments on higher floors are less common.

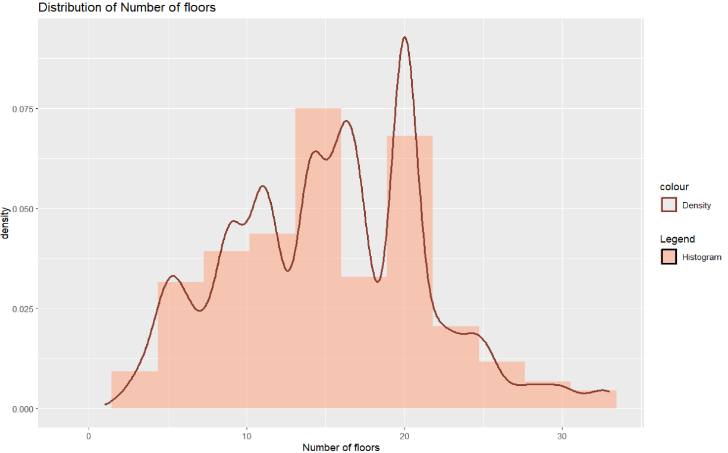
****

Figure 11

Number of floors:

The distribution of apartments based on the building’s number of floors shows peaks around 15 and 20 floors buildings, indicating that these buildings are the most common, with the peak around 15 floors being higher. There is a sharp decline starts from 20 floors to buildings with more than 30 floors, suggesting a scarcity of taller buildings.

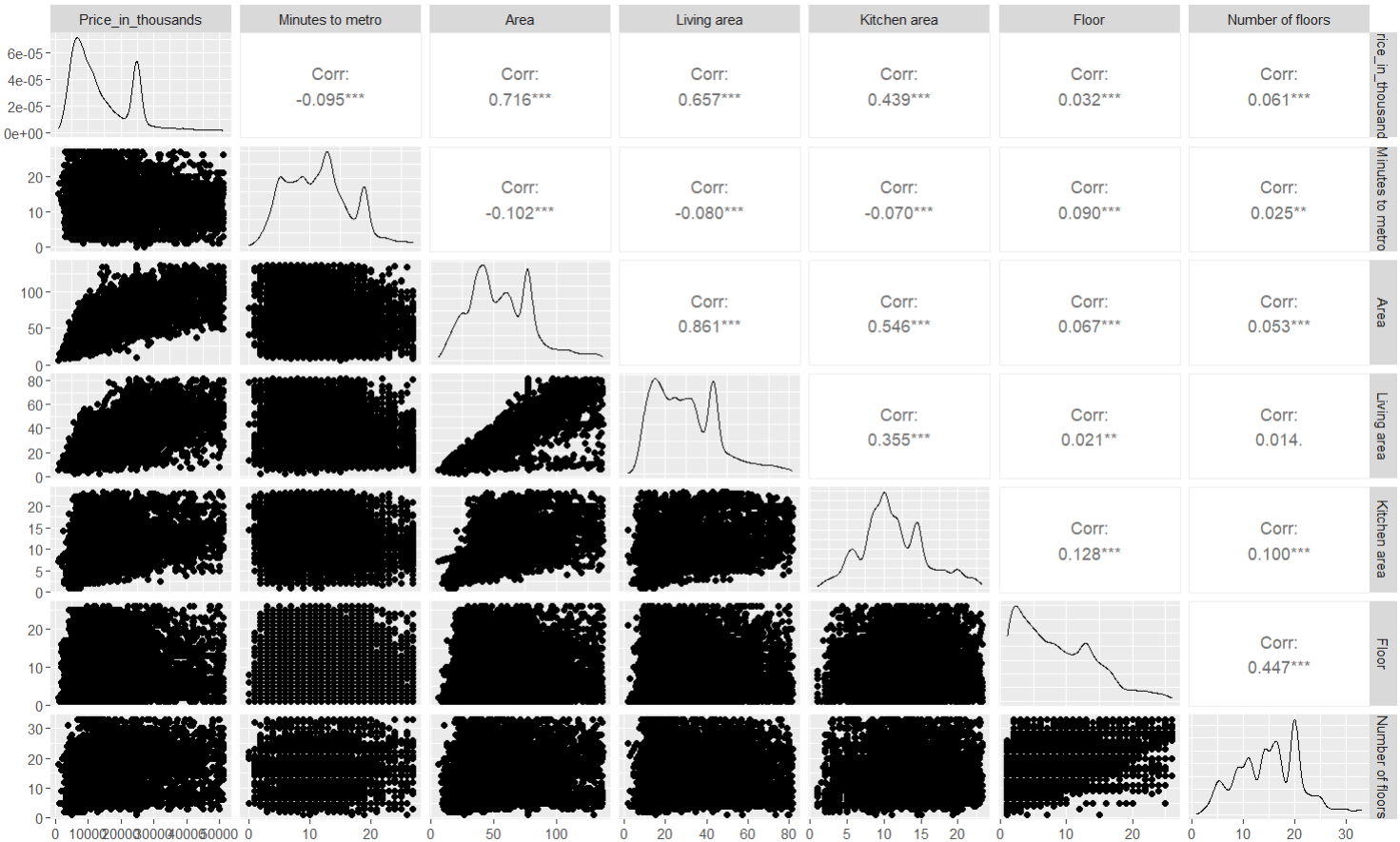
****

Figure 12

Figure 12 presents the pair plot showing the relationships and correlations between quantitative variables, such as price, distance to the metro, area, and other apartment’s characteristics. There is a strong positive correlation between each of, price and area, price and living area, and living area and total area. There is a weak negative correlations between few of the variables, such as between price and minutes to the metro suggesting that proximity to the metro increases property value, and between area and living area with minutes to metro suggesting that the apartments closer to metro stations may be characterized by their larger size . The diagonal includes density plots for individual variables. While scatter plots illustrate pairwise relationships, showing that only area and living area are somewhat linearly correlated.

**Statistical analysis**

We performed some modeling techniques and machine learning algorithms to get answers for our research questions.

Logistic Model

Research question: What factors significantly influence whether the living area is large or small? And how accurately can Area, Kitchen area, and Number of rooms predict whether the living area is above or below the threshold?

The response for the logistic model is the Living Area which is converted into binary (1 or 0) (greater or less than the median), the categorical variable number of rooms were represented as dummies its base category is no rooms

**The 1st model:**

Logit (P) = −7.455 + 0.1805(Area) − 0.2511(Kitchen area) − 2.038(Number of rooms: single) + 0.9495(Number of rooms: two) + 1.325(Number of rooms: three) + 0.3815(Number of rooms: more) + 3.529(Price in thousands)

The Price and the Area were highly correlated and by looking at the coefficients and their p-values and using step wise method (both forward and backward), we found that the price was insignificant so we removed it and reran the model.

**The 2nd and final model:**

Logit (P) = −7.4439 + 0.1816(Area) − 0.2512(Kitchen area) − 2.0480(Number of rooms: single) + 0.9247(Number of rooms: two) + 1.3074(Number of rooms: three) + 0.3785(Number of rooms: more)

The models accuracy = 94%, stating that the model correctly predicts most outcomes. Sensitivity = 95.3% which is a very good percentage of identifying true positive cases. Specificity = 93.2%, showing effectiveness at identifying true negative cases.

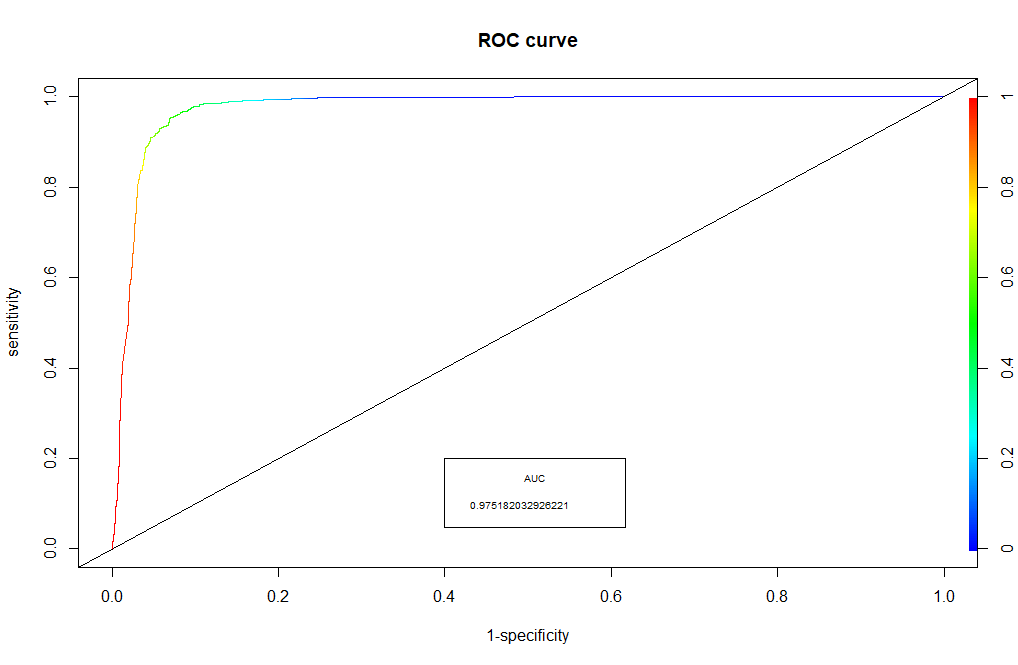
The Kappa measure = 0.885 which is showing a very high agreement between predicted and actual values. McNemar's Test p-value: < 1.423e-10 Indicating an imbalance in false positives and false negatives. McFadden's R-squared = 0.7023 – Good fit, which is explaining 70% of variability in the data. Also the best cutoff point was shown at 0.548.

Figure 13

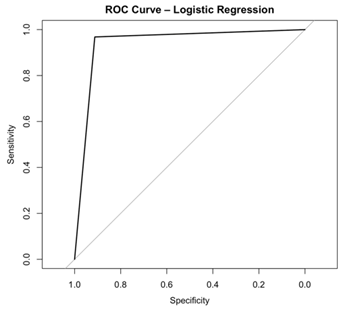


Figure 14

From figures 13 and 14, the ROC curve is close to the top-left corner, which indicates that the model performs well in distinguishing between the two classes and the area under the curve = 0.975.

The logistic regression model achieved an accuracy of 94.23% on the test dataset. This indicates that the model correctly classified approximately 94 out of 100 cases. Such a high accuracy suggests that the model performs well in distinguishing between the target categories.

**Interpreting the coefficients ():**

* Intercept = 0.000585: the estimated odds of having a living area above the median when all predictor variables are zero is 0.0585% of the base categories odds. It indicates that in the absence of any predictors, the odds of having a living area above the median are extremely low.
* Area = 1.199: For each additional square meter of area, the estimated odds of having a living area above the median increase by 19.9%. This means that larger apartments are more likely to have a living area above the median.
* Kitchen area = 0.778: For each additional square meter of kitchen area, the estimated odds of having a living area above the median decrease by 22.2%. This indicates that larger kitchens are associated with a lower odds of having a living area above the median.

Number of rooms (relative to no rooms, which is the reference category):

* Number of rooms (single) = 0.129: Apartments with a single room have an 87.1% decrease in the estimated odds of having a living area above the median compared to the estimated odds of those with no rooms, implying that single-room apartments are less likely to have a larger living area.
* Number of rooms (two) = 2.521: Apartments with two rooms have a 152.1% increase in the estimated odds of having a living area above the median compared to the estimated odds of those with no rooms, meaning that they are more likely to have a living area above the median.
* Number of rooms (three) = 3.696: Apartments with three rooms have a 269.6% increase in the estimated odds of having a living area above the median compared to the estimated odds of those with no rooms.
* Number of rooms (more) = 1.460: Apartments with more than three rooms have a 46% increase in the estimated odds of having a living area above the median compared to the estimated odds of those with no rooms, suggesting that apartments with more than three rooms are more likely to have a larger living area.

Regression Model

Research question: what are the variables that have significant effect on the price of the apartments?

The response for the regression model is the price (in thousands)(y), the categorical variables were represented as dummies where the base category for the apartment type is new building, for the metro station is east, for number of rooms is no rooms, and for the renovation is cosmetic

**The 1st model:**

Y= - 4899.554 + 6154.399(Apartment type: secondary) - 1441.328(Metro station: north) - 585.129(Metro station: south) +130.788(Metro station: west) - 55.448(Minutes to metro) + 1052.747(Number of rooms: single) + 922.698(Number of rooms: two) + 1310.574(Number of rooms: three) + 414.055(Number of rooms: more) + 162.863(Area) + 69.439(Living area) + 150.288(Kitchen area) + 62.262(Floor) + 79.683(Number of floors) + 4364.667(Renovation: Designer) + 3853.540(Renovation: European-style) + 5098.843(Renovation: Without renovation)

Adjusted R-squared: 0.7411

Checking the assumptions:

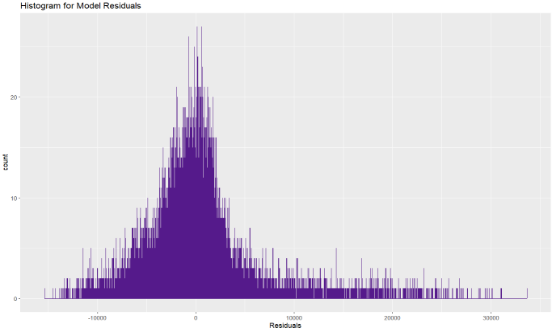
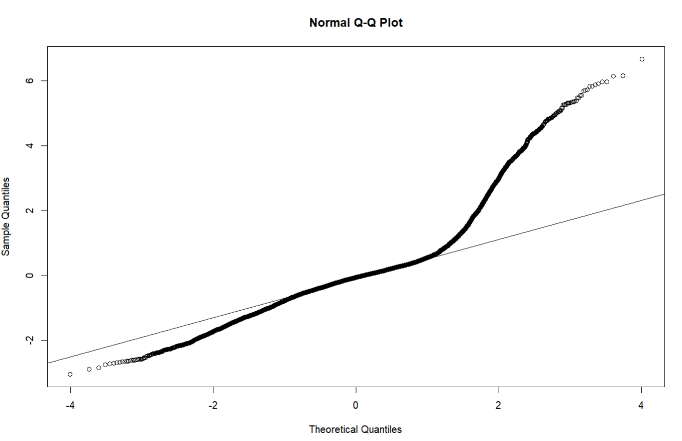


Figure 13: normal Q-Q plot

Figure 14: residuals histogram

Residuals normality: The Q-Q plot reveals some deviations from normality in the tails, suggesting heavier distributions at the extremes. Also, the histogram of residuals shows an almost symmetric distribution but with heavier right tail. However, given the large sample size used in the analysis, we can use the Central Limit Theorem, which justifies assuming normality for inferential purposes.

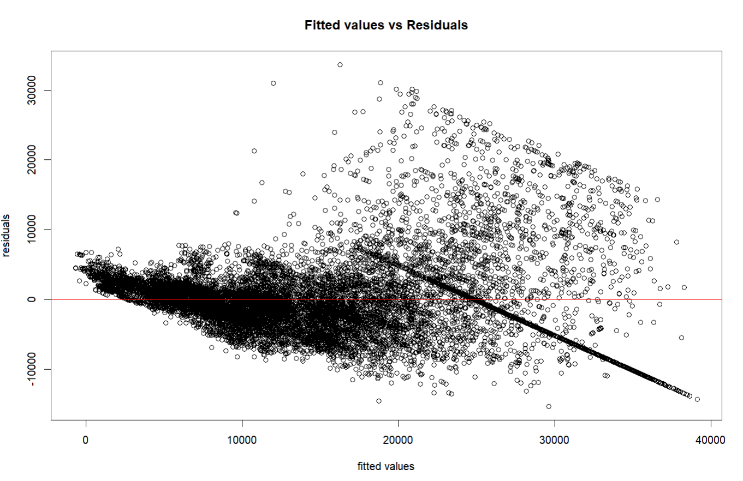


Figure 15

Homoscedasticity: figures 17 presents the residuals versus fitted values scatter plot which shows that some parts of the residuals are randomly scattered around the zero line, almost satisfying the homoscedasticity assumption. However, there is a noticeable downward trend in the residuals at higher fitted values, suggesting potential violation of the assumption at large values.

Table 3: VIF table

|  |  |
| --- | --- |
| variables | VIF |
| Apartment type: Secondary | 2.2315 |
| Metro station: North | 2.0779 |
| Metro station: South | 2.1373 |
| Metro station: West | 1.9854 |
| Minutes to metro | 1.0974 |
| Number of rooms: single | 2.2289 |
| Number of rooms: two | 2.8524 |
| Number of rooms: three | 3.2820 |
| Number of rooms: more | 3.5588 |
| Area | 6.1273 |
| Living area | 4.6250 |
| Kitchen area | 1.7577 |
| Floor | 1.3212 |
| Number of floors | 1.4189 |
| Renovation: Designer | 1.8582 |
| Renovation: European-style | 1.6307 |
| Renovation: Without renovation | 1.8509 |

Multicollinyarity: Since generally a VIF value below 10 suggest insignificance of multicollinearity and in this model all variables have VIF values less than 10, so we can say that multicollinearity is not significant in this model. The highest VIF is for the Area at 6.1273 which is still acceptable. This indicates that while some correlation exists between these variables, it is not that severe.

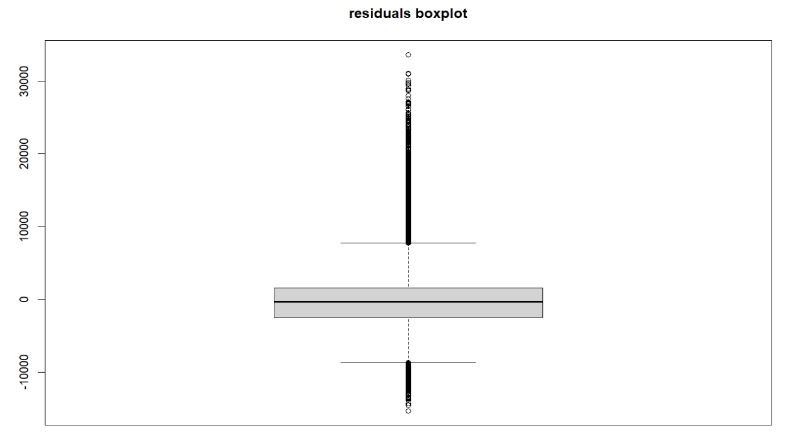


Figure 16: residuals boxplot

Outliers: figure 18 shows that there are quite a lot of outliers present on both ends, while the majority of residuals are fall around the zero and have a relatively narrow interquartile range. This suggests that the model may struggle to accurately predict extreme values.

**The 2nd model:**

We tried taking the log of the response variable, aiming that it may help with normality of residuals, losing the trend in the higher values of the residuals, reduce the influence of extrema values. Also we removed “living area” and “kitchen area” because they are highly correlated with “area” and removing them didn’t affect the goodness of the model, and overall it’s a better fitting model.

Log (Y) = 7.9878227 + 0.5040360 (Apartment type: secondary) - 0.1811199(Metro station: north) - 0.0592974(Metro station: south) - 0.0334464(Metro station: west) - 0.0033993(Minutes to metro) + 0.327735(Number of rooms: single) + 0.3558686(Number of rooms: two) + 0.3553278(Number of rooms: three) + 0.2645714(Number of rooms: more) + 0.0133134(Area) + 0.0054765(Floor) + 0.0022617(Number of floors) + 0.2864544(Renovation: Designer) +0.2541517(Renovation: European-style) + 0.2758101(Renovation: Without renovation)

Adjusted R-squared: 0.8175

Checking the assumptions: After applying the log transformation to the dependent variable:

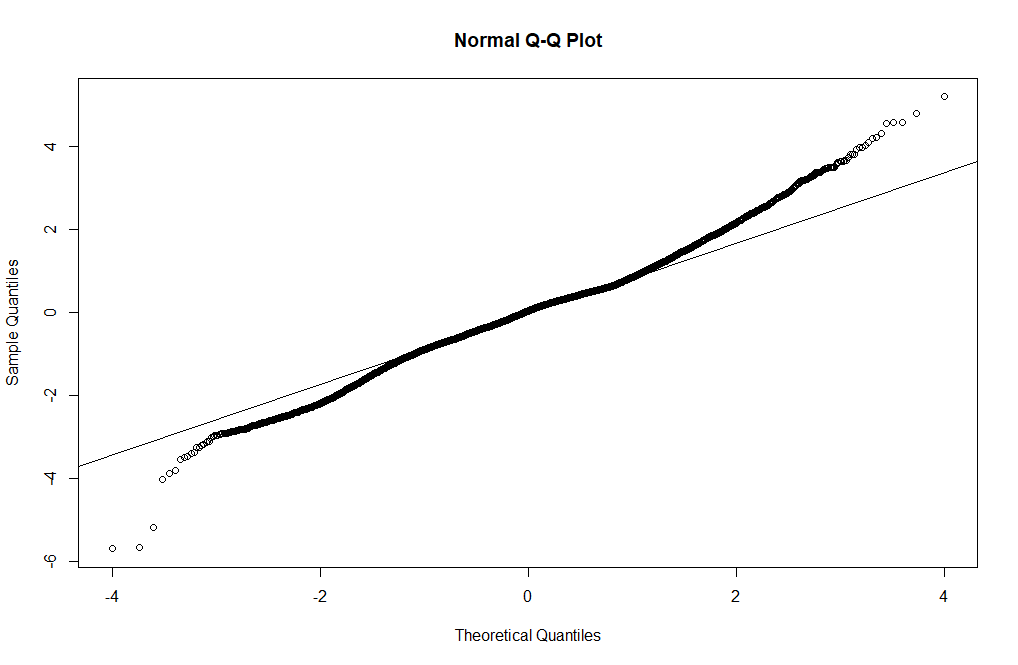


Figure 17: normal Q-Q plot

The Q-Q plot indicates a slight improvement in normality. The residuals align more closely to the diagonal line, suggesting that the assumption of normality is better satisfied. So as mentioned before, using the Central Limit Theorem we assume normality since we are working with a large sample size.

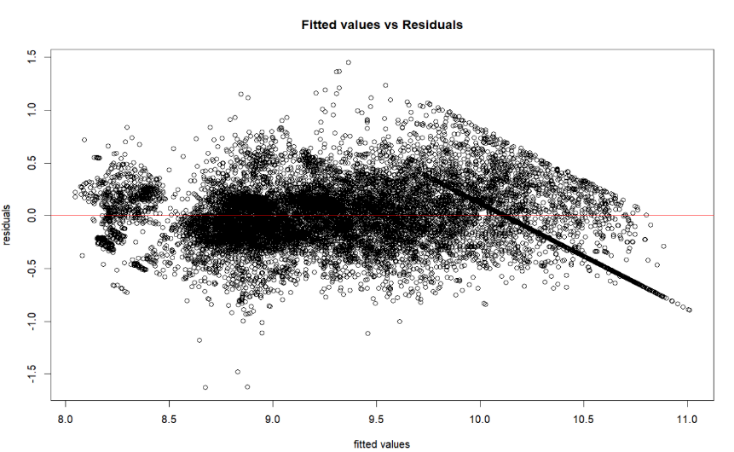


Figure 18: fitted values vs residuals

The residuals versus fitted values plot shows somewhat a downward pattern that remains at higher fitted values. However, there is a notable improvement with residuals appearing more randomly scattered around the zero line, indicating better satisfaction of the homoscedasticity assumption.

So overall there was a slight improvement in the assumptions satisfaction. Additionally, the adjusted R-squared of the 2nd model explains approximately 82% of the variability in log of the apartment’s prices, so it’s a better fitting model for the data.

Interpreting the coefficients:

* Intercept (7.9878227): This is the log of the price (in thousands) for a reference apartment (a new building, in the East metro station, with no rooms, 0 minutes to the metro, 0 area, no renovation, etc.).
* Apartment type: Secondary (0.5040360): Apartments in secondary buildings are associated with 0.5 increase in log the price compared to new buildings, holding other variables constant.
* Metro stations relative to the east stations:

Metro station: North (- 0.1811199): Apartments near the North metro stations have a 0.181 decrease in log price compared to apartments near the East metro stations, holding other variables constant.

Metro station: South (- 0.0592974): Apartments near the South metro stations show a decrease in log price by 0.059 unit, compared to those near the East metro stations, holding other variables constant.

Metro station: West (-0.0334464): Apartments near the West metro stations are associated with a 0.033 decrease in log price compared to those near the East metro stations, holding other variables constant.

* Minutes to metro (- 0.0033993): For every additional minute to the nearest metro station, the log price decreases by 0.003, holding other variables constant.
* Number of rooms relative to no rooms:

Number of rooms: Single (0.327735): Apartments with one room have a 0.33 increase in log price compared to apartments with no rooms, holding other variables constant.

Number of rooms: Two (0.3558686): Apartments with two rooms have a 0.355 increase in log price compared to those with no rooms, holding other variables constant.

Number of rooms: Three (0.3553278): Apartments with three rooms have a 0.355 increase in log price compared to the base category, holding other variables constant.

Number of rooms: More (0.2645714): Apartments with more than three rooms have a 0.26 increase in log price compared to those with no rooms, holding other variables constant.

* Area (0.0133134): For every additional unit increase in the area, the log price increases by 0.013, reflecting the positive impact of larger areas on the price, holding other variables constant.
* Floor (0.0054765): For each floor higher, the log price increases by 0.0054, holding other variables constant. Indicating higher floors are slightly more expensive.
* Number of floors (0.0022617): An increase in the total number of floors adds 0.0022 to the log price, holding other variables constant.
* Renovation types relative to cosmetic renovation:

Designer (0. 2864544): Apartments with designer renovations have a 0.286 increase in log price compared to those with cosmetic renovations, holding other variables constant.

European-style (0. 2541517): European-style renovations result in a 0.254 increase in log price compared to those with cosmetic renovations, holding other variables constant.

Without renovation (0. 2758101): Apartments without renovation have a 0.276 increase in log price compared to those with cosmetic renovations, holding other variables constant.

Machine learning algorithm “decision tree”

Research question: What variables can classify apartments as high or low value?

* We started by factoring the price variable into two categories (high if price > mean) and (low if price < mean).
* We applied decision tree on the variables and we reached the following :

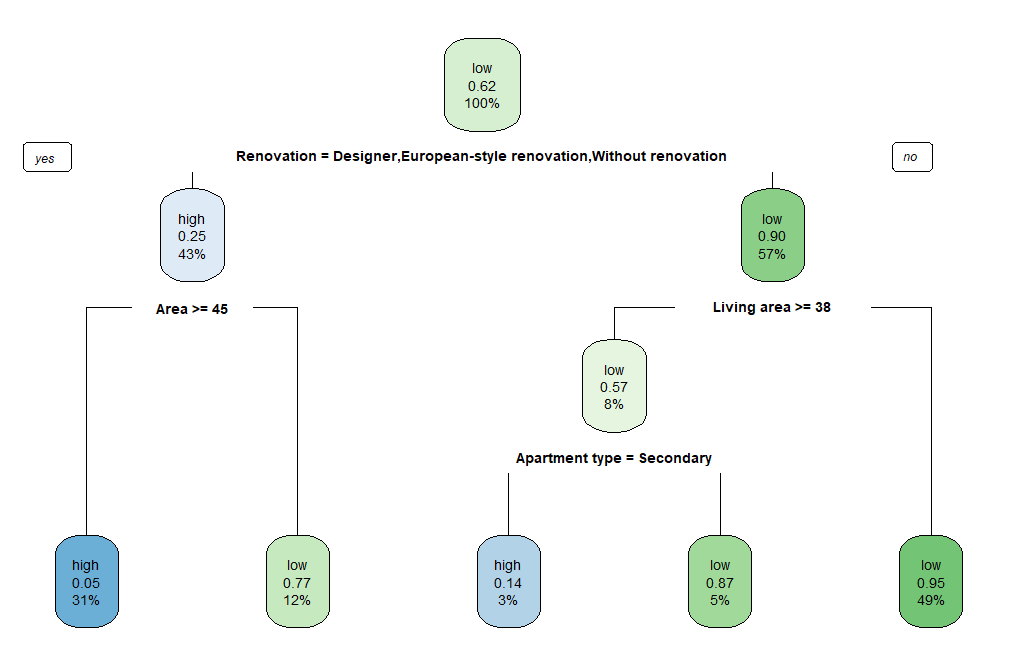


Figure 19: decision tree

From figure 21:

* We started with that 62% of apartments are classified as "low" price overall.
* Variables were arranged based on the maximum information gain, starting with “renovation”, then “area” and “living area”, and finally “apartment type”.
* If an apartment has cosmetic Renovation: high price probability of 43%, otherwise low price probability of 57%.
* The next important feature for splitting is Area: Area ≥ 45: then the probability of low price becomes 77%, While Area < 45: has a probability of high price equals 31%.
* On the other hand, if living Area < 38: leads to a probability of low price = 95%, otherwise we have an 8% probability of low price.
* Finally we divide the 8% based on the apartment type, so if it’s a secondary apartment then the probability of high price = 3%, otherwise, the probability of low price = 5%.
* So we can say that :

Apartments with renovation type not cosmetic and with area >= 45 meter square, and Apartments with renovation type cosmetic, living area >=38 and apartment type secondary, have probabilities of being a high priced apartments 31% and 3% respectively. On the other hand, Apartments with renovation type not cosmetic and with area <45 meter square, Apartments with renovation type cosmetic, living area >=38, and apartment type new building and Apartments with renovation type cosmetic, living area <38, have probabilities of being a low priced apartments 12%,5% and 49% respectively.

Table 4: confusion matrix

|  |  |  |
| --- | --- | --- |
| Reference  Prediction | High | Low |
| High | 2215 | 368 |
| Low | 128 | 4050 |

Table 4 shows that:

* The percentages of the correctly specified:

High 75%

Low 97%.

* The overall accuracy of classification = 0.9266 which is very high accuracy.
* 95% Confidence Interval (CI): (0.9202, 0.9327)

**Conclusion**

So we can conclude that:

* From the logistic model: larger apartments (in terms of total area) are more likely to have a living area above the median, however, Larger kitchen areas decrease the likelihood of having a living area above the median and Apartments with more rooms (especially two or three rooms) have significantly higher odds of having a living area above the median, while single-room apartments are much less likely to have a larger living area
* From the regression model: the model with log the dependent variable is the better fitting model in our analysis since it’s explaining approximately 82% of the variability in log of the apartment’s prices. Additionally, all of the variables are of significance to the price. Where all of the variables have a positive relationship with the price except for the time needed to reach the metro station walking and the metro station.
* From the machine learning algorithm: we can classify apartments as high or low priced using the type of the renovation done to the apartment, apartment’s total area, living area and the apartment’s type with 93% accuracy.

**My function**

We created 2 functions the 1st one is our main function, the 2nd is an extra idea and both of them are between lines 430:458 in our R script.

1. The function called “my-func” calculates the price of the apartment in case paid as installments not in cash.

The function takes the original price of the apartment and the interest rate, and it calculates the new price iterating over 3 different installment periods (in months) to give the buyer an idea on the different apartment prices in case of different periods of payment.

1. This function is designed to predict the price in thousands of an apartment based on the results of the regression model.

The function takes variables “or their categories” that was kept in the regression model, which represent the characteristics of an apartment, calculates the predicted log price using the given coefficients from the regression model.

**Appendix**

library(readxl)

Housing\_Price\_price\_data <- read\_excel("Housing Price dataset.xlsx")

View(Housing\_Price\_price\_data)

set.seed(123)

sample<-sample(c(TRUE,FALSE),nrow(Housing\_Price\_price\_data),replace=TRUE , prob=c(0.7,0.3))

price\_data<-Housing\_Price\_price\_data[sample,]

test<- Housing\_Price\_price\_data[!sample,]

View(price\_data)

summary(price\_data)

# data prep

price\_data[] <- lapply(price\_data, function(x) {

if (is.character(x)) {

as.factor(x)

} else {

x

}

})

# missing values

p\_na<-function(x){sum(is.na(x))/length(x)\*100}

apply(price\_data,2,p\_na)

na.omit(price\_data)

# remove non-used var

library(dplyr)

price\_data<- price\_data %>% select(-Region)

# change number of rooms into categorical

table(price\_data$`Number of rooms`)

breaks<-c(min(price\_data$`Number of rooms`)-1,0,1,2,3,max(price\_data$`Number of rooms`))

price\_data$`Number of rooms`<- cut(price\_data$`Number of rooms`,breaks,labels = c("no\_rooms","single","two","three","more"))

summary(price\_data$`Number of rooms`)

# change price structure to be in thousands (easier to read)

price\_data$Price<- price\_data$Price/1000

price\_data <- price\_data %>% rename(Price\_in\_thousands = Price)

summary(price\_data$Price\_in\_thousands)

# group metro stations

table(price\_data$`Metro station`)

north\_stations <- c("Алтуфьево","Санино","Долгопрудная","Улица Академика Королёва","Аникеевка ","Бутырская","ВДНХ","Верхние Котлы","Бескудниково" ,"Бабушкинская", "Белокаменная", "Беломорская","Бибирево","Верхние Лихоборы","Пятницкое шоссе", "Аникеевка", "Динамо", "Водный стадион", "Владыкино"," Водный стадион", "Войковская", "Волоколамская", "Дегунино", "Деловой центр", "Депо, Динамо", "Дмитровская","Зеленоград — Крюково", "Зорге", "Карамышевская","Коптево", "Красногвардейская","Красносельская", "Красные ворота", "Курская","Лианозово", "Лихоборы", "Марьина Роща", "Марьина Роща (Шереметьевская)","Медведково", "Митино", "Народное Ополчение","Новоподрезково", "Окружная", "Отрадное", "Петровско-Разумовская", "Покровское", "Планерная", "Полежаевская", "Речной вокзал", "Рижская", "Ростокино","Савеловская", "Савёловская", "Селигерская", "Сокол", "Стахановская", "Стрешнево", "Сходненская", "Суликатная", "Тимирязевская", "Трикотажная","Тушинская", "Физтех", "Хлебниково", "Ховрино", "Хорошево", "Хорошёво", "Хорошёвская", "Щукинская", "Яхромская")

south\_stations <- c("Аннино","Академическая" ,"Аэропорт Внуково","Варшавская (Коломенское)","Бульвар Адмирала Ушакова","Братиславская","Битца", "Битцевский Парк","Борисово","Верхние котлы","Пенягино", "Пражская", "Беляево", "Раменки", "Выхино", "Красный Строитель", "Улица Дмитриевского","Верхние Лихоборы","Вешняки","Боровское шоссе", "Беляево ","Бунинская аллея","Бунинская Аллея","Бутово","Бульвар Дмитрия Донского", "Варшавская","Верхние котлы", "Вешняки", "Волгоградский проспект", "Воробьёвы горы", "Воронцовская", "Выставочная", "Говорово", "Гражданская", "Давыдково", "Добрынинская", "Домодедовская", "Дубровка", "Жулебино", "ЗИЛ", "Зюзино", "Зябликово", "Калитники", "Калужская", "Кантемировская", "Каховская", "Каширская", "Китай-город","Достоевская","Кожуховская", "Коломенская", "Коммунарка", "Коньково", "Косино", "Котельники","Крестьянская застава", "Крымская", "Кузьминки","Курьяново", "Ленинский проспект", "Лермонтовский проспект", "Лесопарковая", "Лухмановская", "Люблино", "Марксистская", "Марьино", "Матвеевская","Мичуринский проспект", "Москворечье", "Нагатинская", "Нагатинский Затон", "Нагорная", "Нахимовский проспект", "Новопеределкино", "Новоясеневская", "Новые Черемушки", "Новые Черёмушки", "Озёрная", "Ольховая", "Орехово", "Остафьево", "Парк Победы", "Перерва", "Печатники", "Подольск", "Прокшино", "Пролетарская", "Проспект Вернадского", "Профсоюзная", "Пыхтино", "Рабочий посёлок", "Рабочий Посёлок", "Рассказовка","Румянцево", "Саларьево", "Севастопольская", "Семеновская", "Семёновская", "Серпуховская", "Силикатная", "Сетунь", "Студенческая", "Текстильщики", "Теплый Стан", "Тёплый Стан", "Терехово", "Тропарево", "Тропарёво","Тульская", "Угрешская", "Улица Академика Янгеля", "Улица Горчакова", "Улица Скобелевская", "Улица Старокачаловская", "Университет", "Филатов Луг", "Царицыно", "Чертановская", "Шаболовская", "Шипиловская", "Щербинка", "Юго-Восточная", "Юго-Западная", "Южная", "Ясенево")

east\_stations <- c("Авиамоторная", "Автозаводская","Соколиная гора","Библиотека им. Ленина", "Алексеевская","Проспект Мира", "Шоссе Энтузиастов","Лубянка", "Алма-Атинская", "Андроновка","Бауманская","Ботанический сад", "Бульвар Рокоссовского", "ВДНХ","Волжская", "Измайлово", "Измайловская","Косино", "Котельники","Кузнецкий мост","Лубянка ","Депо","Комсомольская","Лефортово", "Локомотив","Маяковская", "Менделеевская", "Москва-Товарная","Некрасовка","Нижегородская", "Новаторская", "Новогиреево", "Новодачная", "Новокосино", "Новохохловская", "Окская","Партизанская", "Первомайская", "Перово", "Преображенская площадь","Площадь Гагарина", "Площадь Ильича", "Площадь Революции", "Полянка", "Пушкинская", "Римская","Рязанский проспект", "Свиблово", "Соколиная Гора", "Сокольники", "Таганская","Смоленская", "Сретенский бульвар", "Сухаревская", "Тверская", "Театральная", "Третьяковская","Фонвизинская", "Черкизовская", "Шоссе энтузиастов", "Щелковская", "Щёлковская", "Электрозаводская")

west\_stations <- c("Арбатская", "Аэропорт","Воробьевы горы","Терехово (Мнёвники)","Александровский сад","Аминьевская","Библиотека и Ленина","Выставочный центр", "Петровский Парк", "Багратионовская", "Беговая", "Белорусская", "Боровицкая","Балтийская", "Баррикадная","Внуково", "Волоколамская","Пионерская", "Раменки", "Петровский парк", "Выставочная", "Киевская","Кленовый бульвар", "Красногорская", "Краснопресненская","Красный Балтиец", "Кропоткинская", "Крылатское", "Кунцевская","Кутузовская", "Ломоносовский проспект", "Лужники", "Марк","Международная", "Минская", "Мнёвники", "Молодежная", "Молодёжная", "Мякинино", "Нахабино","Немчиновка", "Озёрная", "Октябрьское поле", "Опалиха", "Очаково", "Павшино", "Панфиловская","Новокузнецкая", "Новослободская", "Октябрьская", "Охотный ряд", "Павелецкая", "Парк культуры", "Парк Культуры","Площадь Гагарина", "Площадь Ильича", "Площадь Революции", "Полянка", "Пушкинская", "Римская","Сколково", "Славянский бульвар", "Солнцево", "Спартак", "Спортивная", "Строгино", "Сухаревская", "Тестовская", "Технопарк", "Трубная","Улица 1905 года", "Филевский парк", "Филёвский парк", "Фили", "ЦСКА", "Шелепиха","Тургеневская", "Фрунзенская", "Цветной бульвар", "Чеховская", "Чистые пруды", "Чкаловская")

price\_data$`Metro station` <- ifelse(price\_data$`Metro station` %in% north\_stations, "North",

ifelse(price\_data$`Metro station` %in% south\_stations, "South",

ifelse(price\_data$`Metro station` %in% east\_stations, "East",

ifelse(price\_data$`Metro station` %in% west\_stations, "West", NA))))

price\_data$`Metro station` <- as.factor(price\_data$`Metro station`)

summary(price\_data$`Metro station`)

# check for outliers

library(psych)

library(DescTools)

quan<- price\_data %>%select(Price\_in\_thousands,`Minutes to metro`,Area,`Living area`,`Kitchen area`,Floor,`Number of floors`)

boxplot(price\_data$Price\_in\_thousands)

boxplot(price\_data$`Minutes to metro`)

boxplot(price\_data$Area)

boxplot(price\_data$`Living area`)

boxplot(price\_data$`Kitchen area`)

boxplot(price\_data$Floor)

boxplot(price\_data$`Number of floors`)

# num of outliers

count\_outliers <- function(x) {

return(length(boxplot.stats(x)$out))

}

outlier\_counts <- sapply(quan, count\_outliers)

outlier\_counts

# remove outliers

replace\_outliers <- function(k) {

Q1 <- quantile(k, .25)

Q3 <- quantile(k, .75)

IQR\_value <- IQR(k)

lower\_bound <- Q1 - 1.5 \* IQR\_value

upper\_bound <- Q3 + 1.5 \* IQR\_value

k[k < lower\_bound] <- Q1

k[k > upper\_bound] <- Q3

return(k)

}

price\_data[] <- lapply(price\_data[], function(col) {

if (is.numeric(col)) replace\_outliers(col) else col

})

summary(price\_data)

write.csv(summary(price\_data), "summary.csv")

# data visualization

library(ggplot2)

library(gridExtra)

b1<-ggplot(price\_data,aes(`Apartment type`))+geom\_bar(fill= c("turquoise","lightpink"))

b2<-ggplot(price\_data,aes(`Metro station`))+geom\_bar(fill= c("lightgreen","indianred1","yellow1","lightblue"))

b3<-ggplot(price\_data,aes(`Number of rooms`))+geom\_bar(fill= c("lightblue","lightblue1","lightblue2","lightblue3","lightblue4"))

b4<-ggplot(price\_data,aes(Renovation))+geom\_bar(fill= c("deepskyblue","violet","cyan","pink"))

f1<-Freq(price\_data$`Apartment type`)

p1<-ggplot(f1, aes(x="",y=freq, fill=level)) + geom\_bar(stat="identity", width=1)+

coord\_polar("y", start=0) +

geom\_text(aes(label = paste0(round(perc\*100), "%")), position = position\_stack(vjust=0.5)) +

labs(x = NULL, y = NULL)+

scale\_fill\_manual(values = c("turquoise", "lightpink"))

f2<-Freq(price\_data$`Metro station`)

p2<-ggplot(f2, aes(x="",y=freq, fill=level)) + geom\_bar(stat="identity", width=1)+

coord\_polar("y", start=0) +

geom\_text(aes(label = paste0(round(perc\*100), "%")), position = position\_stack(vjust=0.5)) +

labs(x = NULL, y = NULL)+

scale\_fill\_manual(values =c("lightgreen","indianred1","yellow1","lightblue"))

f3<-Freq(price\_data$`Number of rooms`)

p3<-ggplot(f3, aes(x="",y=freq, fill=level)) + geom\_bar(stat="identity", width=1)+

coord\_polar("y", start=0) +

geom\_text(aes(label = paste0(round(perc\*100), "%")), position = position\_stack(vjust=0.5)) +

labs(x = NULL, y = NULL)+

scale\_fill\_manual(values = c("lightblue","lightblue1","lightblue2","lightblue3","lightblue4"))

f4<-Freq(price\_data$Renovation)

p4<-ggplot(f4, aes(x="",y=freq, fill=level)) + geom\_bar(stat="identity", width=1)+

coord\_polar("y", start=0) +

geom\_text(aes(label = paste0(round(perc\*100), "%")), position = position\_stack(vjust=0.5)) +

labs(x = NULL, y = NULL)+

scale\_fill\_manual(values = c("deepskyblue","violet","cyan","pink"))

grid.arrange(b1,p1,nrow=1 , ncol= 2 , top= "Apartment type" )

grid.arrange(b2,p2,nrow=1 , ncol= 2 , top= "Metro Station" )

grid.arrange(b3,p3,nrow=1 , ncol= 2 , top= "# of Rooms" )

grid.arrange(b4,p4,nrow=1 , ncol= 2 , top= "Renovation" )

# histograms

library(scales)

ggplot(price\_data, aes(`Price\_in\_thousands`)) +

geom\_histogram(aes(y=after\_stat(density), fill="Histogram"), bins=12, position="identity", alpha=0.5) +

geom\_density(aes(color="Density"), size = 1, alpha = 0) +scale\_fill\_manual(values=c("Histogram"="violet")) +

scale\_color\_manual(values=c("Density"="violetred")) +labs(fill="Legend",title="Distribution of Price")+scale\_y\_continuous(labels = comma)

ggplot(price\_data, aes(`Minutes to metro`)) +

geom\_histogram(aes(y=after\_stat(density), fill="Histogram"), bins=12, position="identity", alpha=0.5) +

geom\_density(aes(color="Density"), size = 1, alpha = 0) +scale\_fill\_manual(values=c("Histogram"="lightseagreen")) +

scale\_color\_manual(values=c("Density"="steelblue")) +labs(fill="Legend",title="Distribution of Minutes to metro")

ggplot(price\_data, aes(`Area`)) +

geom\_histogram(aes(y=after\_stat(density), fill="Histogram"), bins=12, position="identity", alpha=0.5) +

geom\_density(aes(color="Density"), size = 1, alpha = 0) +scale\_fill\_manual(values=c("Histogram"="plum")) +

scale\_color\_manual(values=c("Density"="mediumorchid")) +labs(fill="Legend",title="Distribution of Area")

ggplot(price\_data, aes(`Living area`)) +

geom\_histogram(aes(y=after\_stat(density), fill="Histogram"), bins=12, position="identity", alpha=0.5) +

geom\_density(aes(color="Density"), size = 1, alpha = 0) +scale\_fill\_manual(values=c("Histogram"="lightblue")) +

scale\_color\_manual(values=c("Density"="navy")) +labs(fill="Legend",title="Distribution of Living Area")

ggplot(price\_data, aes(`Kitchen area`)) +

geom\_histogram(aes(y=after\_stat(density), fill="Histogram"), bins=12, position="identity", alpha=0.5) +

geom\_density(aes(color="Density"), size = 1, alpha = 0) +scale\_fill\_manual(values=c("Histogram"="lightgreen")) +

scale\_color\_manual(values=c("Density"="darkgreen")) +labs(fill="Legend",title="Distribution of Kitchen Area")

ggplot(price\_data, aes(`Floor`)) +

geom\_histogram(aes(y=after\_stat(density), fill="Histogram"), bins=12, position="identity", alpha=0.5) +

geom\_density(aes(color="Density"), size = 1, alpha = 0) +scale\_fill\_manual(values=c("Histogram"="firebrick1")) +

scale\_color\_manual(values=c("Density"="darkred")) +labs(fill="Legend",title="Distribution of Floor")

ggplot(price\_data, aes(`Number of floors`)) +

geom\_histogram(aes(y=after\_stat(density), fill="Histogram"), bins=12, position="identity", alpha=0.5) +

geom\_density(aes(color="Density"),size = 1, alpha = 0) +scale\_fill\_manual(values=c("Histogram"="lightsalmon")) +

scale\_color\_manual(values=c("Density"="coral4")) +labs(fill="Legend",title="Distribution of Number of floors")

# descriptives

library(vcdExtra)

qual\_rel <- function(p, q) {

tbl <- table(p, q)

print("Contingency Table:")

print(tbl)

print("Row Proportions:")

print(proportions(tbl, 1))

print("Table with Margins:")

print(addmargins(tbl))

print("Chi-square Test:")

print(chisq.test(tbl, correct = FALSE))

print("Gamma:")

print(GKgamma(tbl,level =0.95 ))

}

qual\_rel(price\_data$`Apartment type`,price\_data$Renovation)

qual\_rel(price\_data$`Apartment type`,price\_data$`Metro station`)

qual\_rel(price\_data$`Apartment type`,price\_data$`Number of rooms`)

qual\_rel(price\_data$Renovation,price\_data$`Metro station`)

qual\_rel(price\_data$Renovation,price\_data$`Number of rooms`)

qual\_rel(price\_data$`Metro station`,price\_data$`Number of rooms`)

# for quantitative

quan\_data<- price\_data %>%select(Price\_in\_thousands,`Minutes to metro`,Area,`Living area`,`Kitchen area`,Floor,`Number of floors`)

View(quan\_data)

describe(quan\_data)

Desc(quan\_data)

library("Hmisc")

r <- rcorr(as.matrix(quan\_data))

r

write.csv(describe(quan\_data), "describe.csv")

library("GGally")

ggpairs(quan\_data)

ggplot(price\_data, aes(x=`Minutes to metro`, fill=`Metro station`)) +

geom\_bar(position="dodge") +

labs(title="distance for metro station")

# logistic analysis

# creating binary var

price\_data$living\_bin <- ifelse( price\_data$`Living area`>= median(price\_data$`Living area`), 1, 0)

# full model

attach(price\_data)

logistic\_model1 <- glm(living\_bin ~ Area + `Kitchen area`+ `Number of rooms`+ Price\_in\_thousands,

family = binomial(link = "logit"),data = price\_data)

logistic\_model1

summary(logistic\_model1)

selection<-step(logistic\_model1,direction = "both")

summary(selection)

# reduced model(remove price as it is insig)

logistic\_model <- glm(living\_bin ~ Area + `Kitchen area`+ `Number of rooms`,

family = binomial(link = "logit"),data = price\_data)

logistic\_model

summary(logistic\_model)

exp(coef(logistic\_model))

# VIF

car::vif(logistic\_model)

# diviance

anova(logistic\_model)

# McFadden’s R2

library(pscl)

mcfadden\_r2 <-pR2(logistic\_model)["McFadden"]

mcfadden\_r2

# prediction

predictions <- predict(logistic\_model, type = "response")

summary(predictions)

# best cutoff point

library(ROCR)

pred<-prediction(predictions,price\_data$living\_bin)

roc<-performance(pred,"acc")

max<-which.max(slot(roc,'y.values')[[1]])

acc<-slot(roc,'y.values')[[1]][max]

cut<-slot(roc,'x.values')[[1]][max]

print(c(Accuracy = acc , Cutoff= cut))

predicted\_class <- ifelse(predictions > 0.5483376 , 1, 0)

# accuracy

CM <- table(predicted\_class, price\_data$living\_bin)

CM

# error metric

err\_metric <- function(cm) {

accuracy <- sum(diag(cm)) / sum(cm)

print(paste("Accuracy:", accuracy))

}

err\_metric(CM)

# Classification table with useful measures

library(caret)

library(knitr)

confusionMatrix(factor(price\_data$living\_bin), factor(predicted\_class))

CM\_totals<-addmargins(CM, margin = 1:2)

kable(CM\_totals)

# ROC Curve

library(pROC)

roc\_score <- roc(price\_data$living\_bin, predicted\_class)

plot(roc\_score, main = "ROC Curve – Logistic Regression")

#or

roc\_curve<-performance(pred,'tpr','fpr')

plot(roc\_curve,colorize=T,xlab='1-specificity',ylab="sensitivity",main='ROC curve')

abline(0,1)

# area under the curve

auc<-performance(pred,'auc')

auc<-unlist(slot(auc,'y.values'))

legend(0.4,0.2,auc,title= 'AUC',cex = 0.6)

# prepare and apply test data

View(test)

summary(test)

# data prep

test[] <- lapply(test, function(x) {

if (is.character(x)) {

as.factor(x)

} else {

x

}

})

# check missing values

p\_na<-function(x){sum(is.na(x))/length(x)\*100}

apply(test,2,p\_na)

na.omit(test)

# removing non-used var

test<- test %>%select(-Region)

# change number of rooms into factor

table(test$`Number of rooms`)

breaks<-c(min(test$`Number of rooms`)-1,0,1,2,3,max(test$`Number of rooms`))

test$`Number of rooms`<- cut(test$`Number of rooms`,breaks,labels = c("no\_rooms","single","two","three","more"))

summary(test$`Number of rooms`)

# change price structure to be in thousands (easier to read)

test$Price<- test$Price/1000

test <- test %>% rename(Price\_in\_thousands = Price)

summary(test$Price\_in\_thousands)

# group metro stations

table(test$`Metro station`)

north\_stations <- c("Алтуфьево","Санино","Долгопрудная","Улица Академика Королёва","Аникеевка ","Бутырская","ВДНХ","Верхние Котлы","Бескудниково" ,"Бабушкинская", "Белокаменная", "Беломорская","Бибирево","Верхние Лихоборы","Пятницкое шоссе", "Аникеевка", "Динамо", "Водный стадион", "Владыкино"," Водный стадион", "Войковская", "Волоколамская", "Дегунино", "Деловой центр", "Депо, Динамо", "Дмитровская","Зеленоград — Крюково", "Зорге", "Карамышевская","Коптево", "Красногвардейская","Красносельская", "Красные ворота", "Курская","Лианозово", "Лихоборы", "Марьина Роща", "Марьина Роща (Шереметьевская)","Медведково", "Митино", "Народное Ополчение","Новоподрезково", "Окружная", "Отрадное", "Петровско-Разумовская", "Покровское", "Планерная", "Полежаевская", "Речной вокзал", "Рижская", "Ростокино","Савеловская", "Савёловская", "Селигерская", "Сокол", "Стахановская", "Стрешнево", "Сходненская", "Суликатная", "Тимирязевская", "Трикотажная","Тушинская", "Физтех", "Хлебниково", "Ховрино", "Хорошево", "Хорошёво", "Хорошёвская", "Щукинская", "Яхромская")

south\_stations <- c("Аннино","Академическая" ,"Аэропорт Внуково","Варшавская (Коломенское)","Бульвар Адмирала Ушакова","Братиславская","Битца", "Битцевский Парк","Борисово","Верхние котлы","Пенягино", "Пражская", "Беляево", "Раменки", "Выхино", "Красный Строитель", "Улица Дмитриевского","Верхние Лихоборы","Вешняки","Боровское шоссе", "Беляево ","Бунинская аллея","Бунинская Аллея","Бутово","Бульвар Дмитрия Донского", "Варшавская","Верхние котлы", "Вешняки", "Волгоградский проспект", "Воробьёвы горы", "Воронцовская", "Выставочная", "Говорово", "Гражданская", "Давыдково", "Добрынинская", "Домодедовская", "Дубровка", "Жулебино", "ЗИЛ", "Зюзино", "Зябликово", "Калитники", "Калужская", "Кантемировская", "Каховская", "Каширская", "Китай-город","Достоевская","Кожуховская", "Коломенская", "Коммунарка", "Коньково", "Косино", "Котельники","Крестьянская застава", "Крымская", "Кузьминки","Курьяново", "Ленинский проспект", "Лермонтовский проспект", "Лесопарковая", "Лухмановская", "Люблино", "Марксистская", "Марьино", "Матвеевская","Мичуринский проспект", "Москворечье", "Нагатинская", "Нагатинский Затон", "Нагорная", "Нахимовский проспект", "Новопеределкино", "Новоясеневская", "Новые Черемушки", "Новые Черёмушки", "Озёрная", "Ольховая", "Орехово", "Остафьево", "Парк Победы", "Перерва", "Печатники", "Подольск", "Прокшино", "Пролетарская", "Проспект Вернадского", "Профсоюзная", "Пыхтино", "Рабочий посёлок", "Рабочий Посёлок", "Рассказовка","Румянцево", "Саларьево", "Севастопольская", "Семеновская", "Семёновская", "Серпуховская", "Силикатная", "Сетунь", "Студенческая", "Текстильщики", "Теплый Стан", "Тёплый Стан", "Терехово", "Тропарево", "Тропарёво","Тульская", "Угрешская", "Улица Академика Янгеля", "Улица Горчакова", "Улица Скобелевская", "Улица Старокачаловская", "Университет", "Филатов Луг", "Царицыно", "Чертановская", "Шаболовская", "Шипиловская", "Щербинка", "Юго-Восточная", "Юго-Западная", "Южная", "Ясенево")

east\_stations <- c("Авиамоторная", "Автозаводская","Соколиная гора","Библиотека им. Ленина", "Алексеевская","Проспект Мира", "Шоссе Энтузиастов","Лубянка", "Алма-Атинская", "Андроновка","Бауманская","Ботанический сад", "Бульвар Рокоссовского", "ВДНХ","Волжская", "Измайлово", "Измайловская","Косино", "Котельники","Кузнецкий мост","Лубянка ","Депо","Комсомольская","Лефортово", "Локомотив","Маяковская", "Менделеевская", "Москва-Товарная","Некрасовка","Нижегородская", "Новаторская", "Новогиреево", "Новодачная", "Новокосино", "Новохохловская", "Окская","Партизанская", "Первомайская", "Перово", "Преображенская площадь","Площадь Гагарина", "Площадь Ильича", "Площадь Революции", "Полянка", "Пушкинская", "Римская","Рязанский проспект", "Свиблово", "Соколиная Гора", "Сокольники", "Таганская","Смоленская", "Сретенский бульвар", "Сухаревская", "Тверская", "Театральная", "Третьяковская","Фонвизинская", "Черкизовская", "Шоссе энтузиастов", "Щелковская", "Щёлковская", "Электрозаводская")

west\_stations <- c("Арбатская", "Аэропорт","Воробьевы горы","Терехово (Мнёвники)","Александровский сад","Аминьевская","Библиотека и Ленина","Выставочный центр", "Петровский Парк", "Багратионовская", "Беговая", "Белорусская", "Боровицкая","Балтийская", "Баррикадная","Внуково", "Волоколамская","Пионерская", "Раменки", "Петровский парк", "Выставочная", "Киевская","Кленовый бульвар", "Красногорская", "Краснопресненская","Красный Балтиец", "Кропоткинская", "Крылатское", "Кунцевская","Кутузовская", "Ломоносовский проспект", "Лужники", "Марк","Международная", "Минская", "Мнёвники", "Молодежная", "Молодёжная", "Мякинино", "Нахабино","Немчиновка", "Озёрная", "Октябрьское поле", "Опалиха", "Очаково", "Павшино", "Панфиловская","Новокузнецкая", "Новослободская", "Октябрьская", "Охотный ряд", "Павелецкая", "Парк культуры", "Парк Культуры","Площадь Гагарина", "Площадь Ильича", "Площадь Революции", "Полянка", "Пушкинская", "Римская","Сколково", "Славянский бульвар", "Солнцево", "Спартак", "Спортивная", "Строгино", "Сухаревская", "Тестовская", "Технопарк", "Трубная","Улица 1905 года", "Филевский парк", "Филёвский парк", "Фили", "ЦСКА", "Шелепиха","Тургеневская", "Фрунзенская", "Цветной бульвар", "Чеховская", "Чистые пруды", "Чкаловская")

test$`Metro station` <- ifelse(test$`Metro station` %in% north\_stations, "North",

ifelse(test$`Metro station` %in% south\_stations, "South",

ifelse(test$`Metro station` %in% east\_stations, "East",

ifelse(test$`Metro station` %in% west\_stations, "West", NA))))

test$`Metro station` <- as.factor(test$`Metro station`)

summary(test$`Metro station`)

# check for outliers

boxplot(test$Price\_in\_thousands)

boxplot(test$`Minutes to metro`)

boxplot(test$Area)

boxplot(test$`Living area`)

boxplot(test$`Kitchen area`)

boxplot(test$Floor)

boxplot(test$`Number of floors`)

#remove outliers

test[] <- lapply(test[], function(col) {

if (is.numeric(col)) replace\_outliers(col) else col

})

# descriptives

qual\_rel(test$`Apartment type`,test$Renovation)

# for quantitative

quan\_data<- test %>%select(Price\_in\_thousands,`Minutes to metro`,Area,`Living area`,`Kitchen area`,Floor,`Number of floors`)

View(quan\_data)

describe(quan\_data)

Desc(quan\_data)

r <- rcorr(as.matrix(quan\_data))

r

ggpairs(quan\_data)

# applying on test data

test$LivingAreaBinarytest <- ifelse( test$`Living area`>= median(test$`Living area`), 1, 0)

predicted\_probs <- predict(logistic\_model, test, type = "response")

# Convert probabilities to binary

predicted\_classes <- ifelse(predicted\_probs > 0.5, 1, 0)

# Create a confusion matrix

actual\_classes <- test$LivingAreaBinarytest

confusion\_matrix\_test <- table(Predicted = predicted\_classes, Actual = actual\_classes)

print(confusion\_matrix\_test)

# Accuracy

accuracy\_test <- sum(diag(confusion\_matrix\_test)) / sum(confusion\_matrix\_test)

print(paste("Accuracy:", accuracy\_test))

# regression analysis

# model 1

fit1 <- lm(Price\_in\_thousands ~ `Apartment type` + `Metro station` + `Minutes to metro` +

`Number of rooms` + Area + `Living area` + `Kitchen area`+

Floor + `Number of floors` + Renovation, data = price\_data)

summary(fit1)

library(ggplot2)

ggplot(data=price\_data, aes(fit1$residuals)) +geom\_histogram(binwidth=6, color="purple4", fill="black") +ggtitle("Histogram for Model Residuals")+labs(x="Residuals")

# normality of residuals

qqnorm(rstandard(fit1))

qqline(rstandard(fit1))

# homoscedasticity

plot(fit1$residuals ~ fit1$fitted.values,xlab="fitted values",ylab = "residuals",main="Fitted values vs Residuals")

abline(h=0, col="red")

# multicollinearity

library(mctest)

mctest(fit1, type="i")

# outliers

plot(cooks.distance(fit1), pch=16, col="blue",main="Cooks Distance")

boxplot(fit1$residuals,main="residuals boxplot")

# stepwise

selection<-step(fit1 ,direction = "both")

summary(selection)

# model 2

fit2 <- lm(log(Price\_in\_thousands) ~ `Apartment type` + `Metro station`+ `Minutes to metro` +

`Number of rooms` + Area + Floor +`Number of floors` + Renovation, data = price\_data)

summary(fit2)

# # checking normality of residuals

qqnorm(rstandard(fit2))

qqline(rstandard(fit2))

# homoscedasticity

plot(fit2$residuals ~ fit2$fitted.values,xlab="fitted values",ylab = "residuals",main="Fitted values vs Residuals")

abline(h=0, col="red")

# multicollinearity

mctest(fit2, type="i")

# outliers

plot(cooks.distance(fit2), pch=16, col="darkgreen",main="Cooks Distance")

boxplot(fit2$residuals,main="residuals boxplot")

# ml

# Decision tree

library(caTools)

library(rpart)

library(rpart.plot)

library(caret)

library(dplyr)

price\_data$prcat <- as.factor(ifelse(price\_data$Price\_in\_thousands >mean(price\_data$Price\_in\_thousands) ,"high","low"))

test$prcat <- as.factor(ifelse(test$Price\_in\_thousands >mean(test$Price\_in\_thousands) ,"high","low"))

model1 <- rpart(prcat ~ ., data = price\_data[,-c(1,12)], method = "class")

model1

rpart.plot(model1)

# feature importance

importance <- varImp(model1)

importance %>%arrange(importance,desc(Overall))

# Making predictions

pred <- predict(model1, newdata = test[,-c(1,12)], type = "class")

pred

# confusion matrix

confusionMatrix(test$prcat, pred)

# my\_func

my\_func <- function(current\_price, interest\_rate) {

final\_prices <-{}

number\_of\_months <- c(24, 60, 120)

for (i in 1:length(number\_of\_months)) {

final\_price <- current\_price + (current\_price \* (number\_of\_months[i] / 12) \* interest\_rate)

final\_prices[i] <- final\_price

}

return(final\_prices)

}

# example

final\_prices<-my\_func(2000000, 0.06)

final\_prices (final\_prices)

}

# another function idea (predict the price)

predicted\_price <- function(apartment\_type = c("New\_bulding", "Secondary"),metro\_station = c("East","North", "South", "West"),

minutes\_to\_metro,number\_of\_rooms = c("no\_rooms","single", "two", "three", "more"),

area,floor,number\_of\_floors,renovation = c("Renovation","Designer", "European-style renovation", "Without renovation"))

{

log\_price <- 7.99 +(ifelse(apartment\_type == "Secondary", 0.5, 0)) +(ifelse(metro\_station == "North", -0.18, 0)) +

(ifelse(metro\_station == "South", -0.06, 0)) +(ifelse(metro\_station == "West", -0.03, 0)) +(minutes\_to\_metro \* -0.003) +

(ifelse(number\_of\_rooms == "single", 0.33, 0)) + (ifelse(number\_of\_rooms == "two", 0.36, 0)) +(ifelse(number\_of\_rooms == "three", 0.36, 0)) +

(ifelse(number\_of\_rooms == "more", 0.26, 0)) +(area \* 0.013) +(floor \* 0.005) +(number\_of\_floors \* 0.002) +

(ifelse(renovation == "Designer", 0.29, 0)) + (ifelse(renovation == "European-style renovation", 0.25, 0)) +(ifelse(renovation == "Without renovation", 0.28, 0))

price <- exp(log\_price)

price\_category <- ifelse(price > mean(price\_data$Price\_in\_thousands), "high", "low")

print(c(price\_in\_thousands= round(price) , apartment\_cat= price\_category))

}