Apartments for rent

Contents

1 Introduction

Domain: The project aims to determine how prices of US rental apartments are set. The dataset is available in 10K and 100K versions of observations.

Target and business utility: Results from this study have utility for both tenants and landlords. Tenants would benefit from unbiased pricing information when signing or renewing leases. Landlords would be able to secure adequate rent. In addition, landlords could determine which services would have the highest return on investment. From a business perspective, developing a good algorithm with predictive capabilities would be one possible service which a generic real estate agency could leverage to increase profits to address above mentioned needs.

The chosen dataset consists of 10000 instances and 22 attributes. It is the result of sampling obtained from ads on real estate websites in the United States.

Below, attributes' meaning for each and every variable is reported:

1. **id** = unique identifier of apartment;
2. **category** = category of classified;
3. **title** = title text of apartment;
4. **body** = body text of apartment;
5. **amenities** = like AC, basketball,cable, gym, internet access, pool, refrigerator etc;
6. **bathrooms** = number of bathrooms;
7. **bedrooms** = number of bedrooms;
8. **currency** = price in current;
9. **fee** = fee;
10. **has_photo** = photo of apartment;
11. **pets_allowed** = what pets are allowed dogs/cats etc;
12. **price** = rental price of apartment;
13. **price_display** = price converted into display for reader;
14. **price_type** = price in USD;
15. **square_feet** = size of the apartment;
16. **address** = where the apartment is located;
17. **cityname** = where the apartment is located;
18. **state** = where the apartment is located;
19. **latitude** = where the apartment is located;
20. **longitude** = where the apartment is located;
21. **source** = origin of classified;
22. **time** = when classified was created.

2 Preprocessing

This phase's goal is to obtain a complete dataset without null values, missing values, non-significant values and outliers. In addition, new versions of some existing variables were created in order not to lose the information content for the Data Mining phase.

In the following lines, implemented actions are reported to remark the adopted rationale (*for further details, refer to the script of the code*):

- **id** variables is dropped bacaused it is not significant;
- **title** and **body** variables are dropped because their textual content is fully derivable from other attributes;
- **category**: only apartment records are kept. Afterwords, this variables is dropped having only one categorical value;
- **amenities** is converted into a numeric variable (**amenities2**): this new attribute computes the number of amenities in the apartment;
- **bathrooms**: null and non-significant values are replaced with the number of bedrooms (*correlation = 0.71*). Additional null and non-significant values are replaced with 1, It is assumed that each apartment has at leat 1 bathroom;
- **bedrooms**:null and non-significant values are replaced with the number of bathrooms (*correlation = 0.71*). Additional null and non-significant values are replaced with 1, It is assumed that each apartment has at leat 1 bedroom;;
- **currency** is dropped because it has only one categorical value;
- **has_photo**: a new numeric version is created (**has_photo2**);
- **fee** is dropped because it has only one categorical value;
- **pets_allowed**: null and non-significant records are replaced with the categorical value "None". A new numeric version is created (**pets_allowed2**) representing the number of pet categories the apartment welcomes;
- **price**: outliers are dropped (IQR Methodology). A new ordinal version **price_category** is created;
- **cityname** e **state**: missing and non-significant values are replaced with real values (*Google Maps is used*). These 2 variables are deploye in the visualization phase the thery are dropped being derivable from coordinates (*latitude and longitude*);
- **source**: a new numeric version is created (**source2**);
- **time** is dropped because records are not interpretable.

At the end of this phase, the dataset consists of 9367 records (*-6.33%*) and 15 attributes (*-31.82%*).

3 Descriptive analysis and data visualization

Characteristic	**N = 9,367**
source	
GoSection8	30 / 9,367 (0.3%)
Home Rentals	1 / 9,367 (<0.1%)
Listanza	18 / 9,367 (0.2%)
ListedBuy	170 / 9,367 (1.8%)
Real Estate Agent	1 / 9,367 (<0.1%)
RealRentals	66 / 9,367 (0.7%)
rentbits	2 / 9,367 (<0.1%)
RENTCafé	1 / 9,367 (<0.1%)
RentDigs.com	2,624 / 9,367 (28%)
RentLingo	6,439 / 9,367 (69%)
RENTOCULAR	14 / 9,367 (0.1%)
tenantcloud	1 / 9,367 (<0.1%)
amenities	3.55 (3.16) (1.00 (18.00))
bathrooms	
1	6,486 / 9,367 (69%)
1.5	261 / 9,367 (2.8%)
2	2,192 / 9,367 (23%)
2.5	241 / 9,367 (2.6%)
3	116 / 9,367 (1.2%)
3.5	42 / 9,367 (0.4%)
4	23 / 9,367 (0.2%)
4.5	5 / 9,367 (<0.1%)
5	1 / 9,367 (<0.1%)
bedrooms	
1	4,671 / 9,367 (50%)
1.5	1 / 9,367 (<0.1%)
2	3,180 / 9,367 (34%)
2.5	1 / 9,367 (<0.1%)
3	1,148 / 9,367 (12%)
4	318 / 9,367 (3.4%)
5	42 / 9,367 (0.4%)
6	6 / 9,367 (<0.1%)
has_photo	
No	170 / 9,367 (1.8%)
Thumbnail	8,329 / 9,367 (89%)
Yes	868 / 9,367 (9.3%)
pets_allowed	
Cats	458 / 9,367 (4.9%)
Cats,Dogs	4,944 / 9,367 (53%)
Dogs	104 / 9,367 (1.1%)
None	3,861 / 9,367 (41%)
price	1,309.89 (499.64) (200.00 (2,810.00))
price_category	
High	1,873 / 9,367 (20%)
Low	2,500 / 9,367 (27%)
Medium	4,994 / 9,367 (53%)
square_feet	896.41 (428.84) (101.00 (4,614.00))

In this section, a dataset description is provided evaluating how some variables of interest are distributed. Most significant plots are also attached (*for further details, refer to the script of the implemented code or to the Plots folder*).

The following tables and plots give a general overview main variables' statistics and distribution.

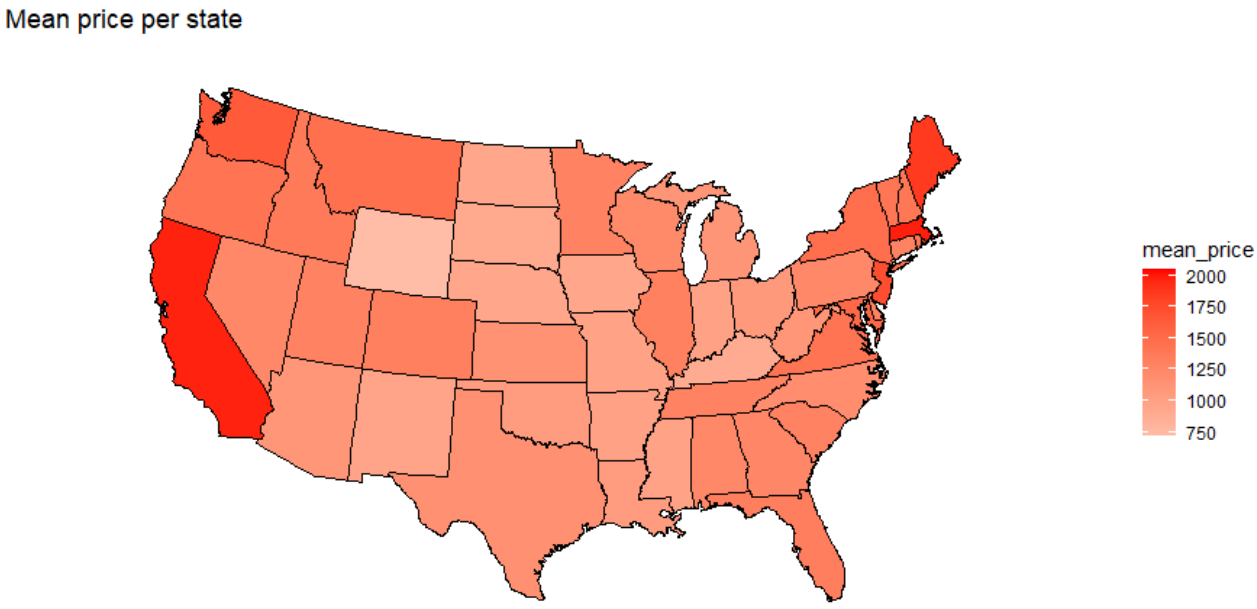


Figure 1: Mean Price per State

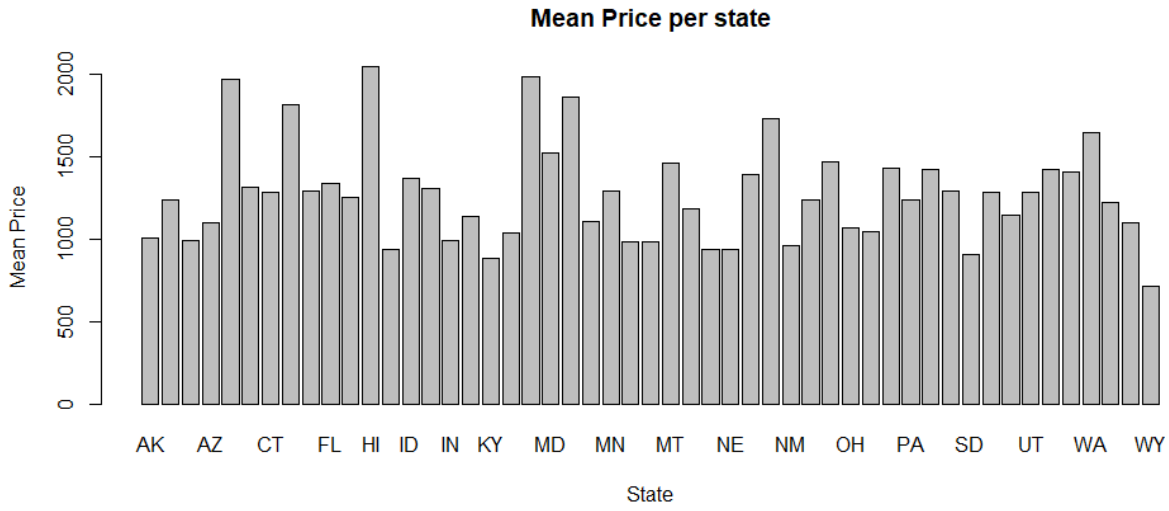


Figure 2: Mean Price per State (Histogram)

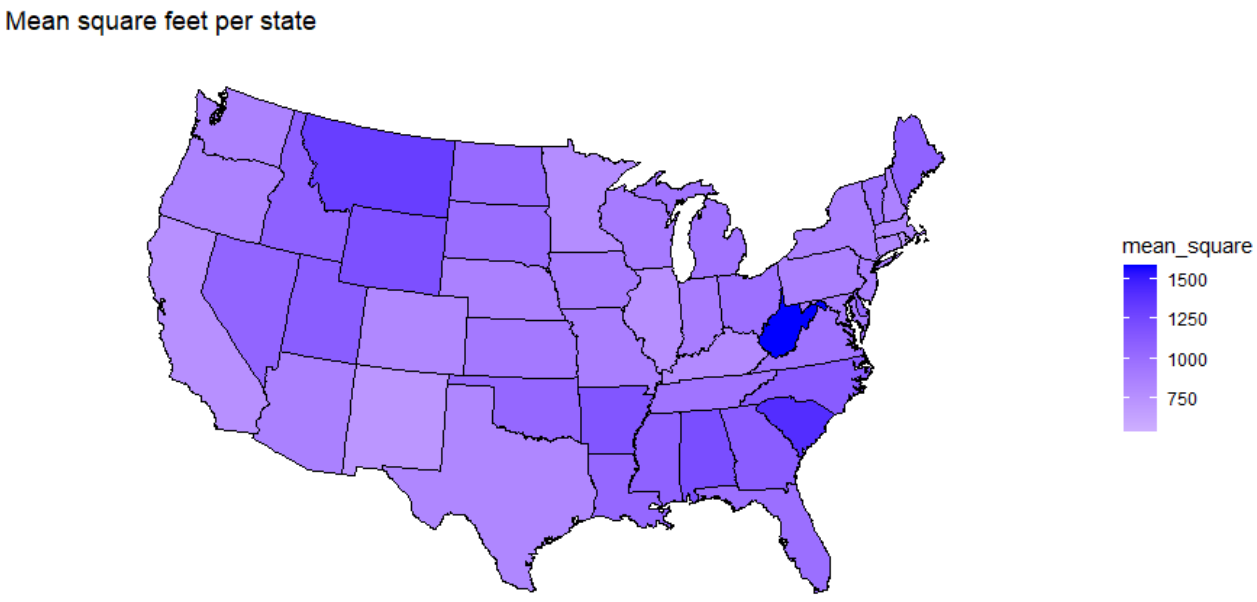


Figure 3: Mean Square Feet per State

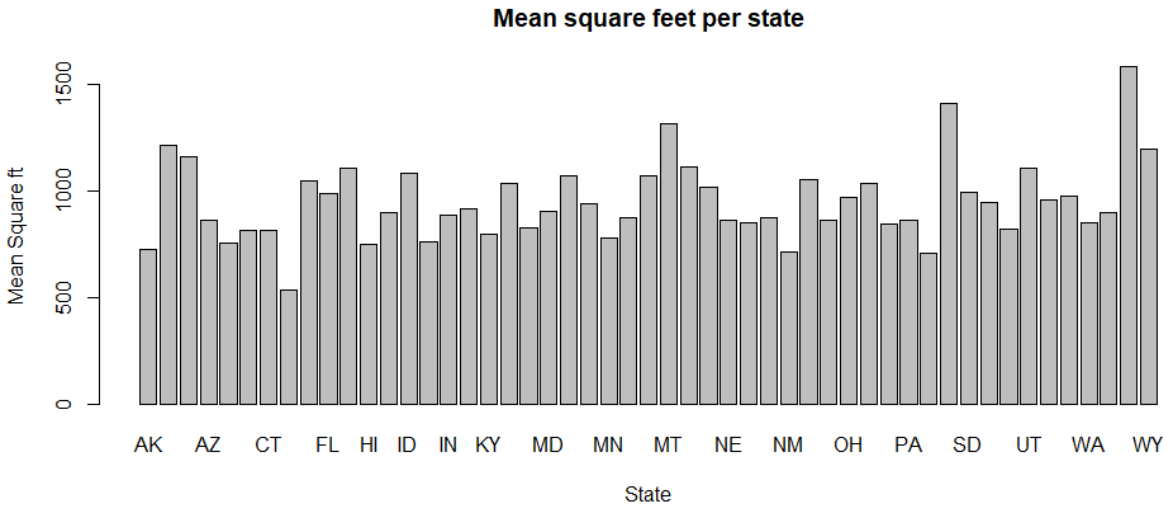


Figure 4: Mean square feet per state (Histogram)

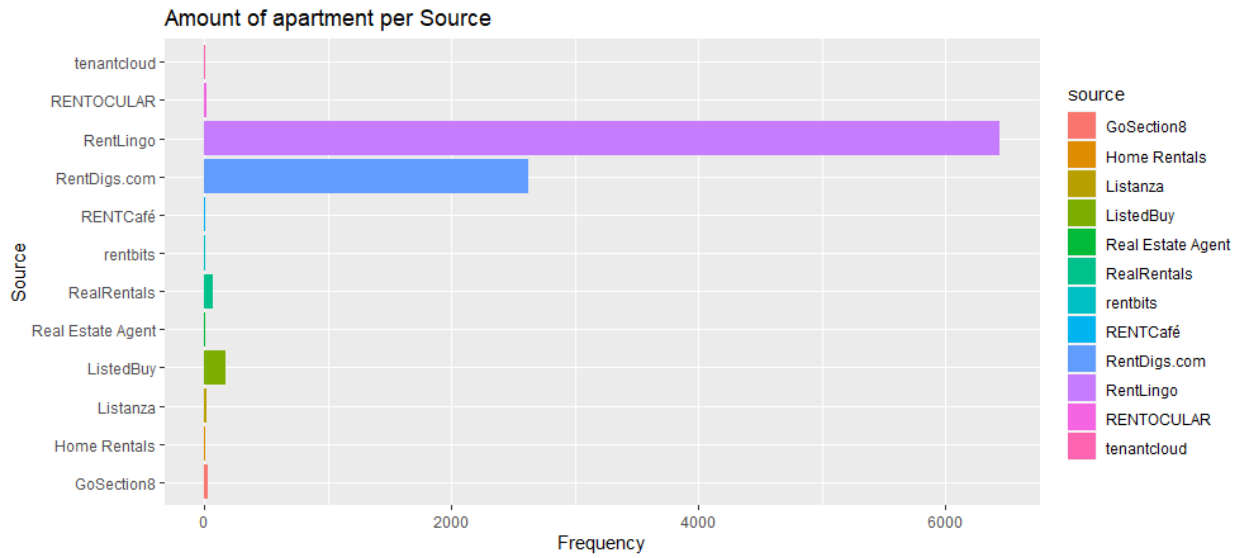


Figure 5: Amount of apartment per Source

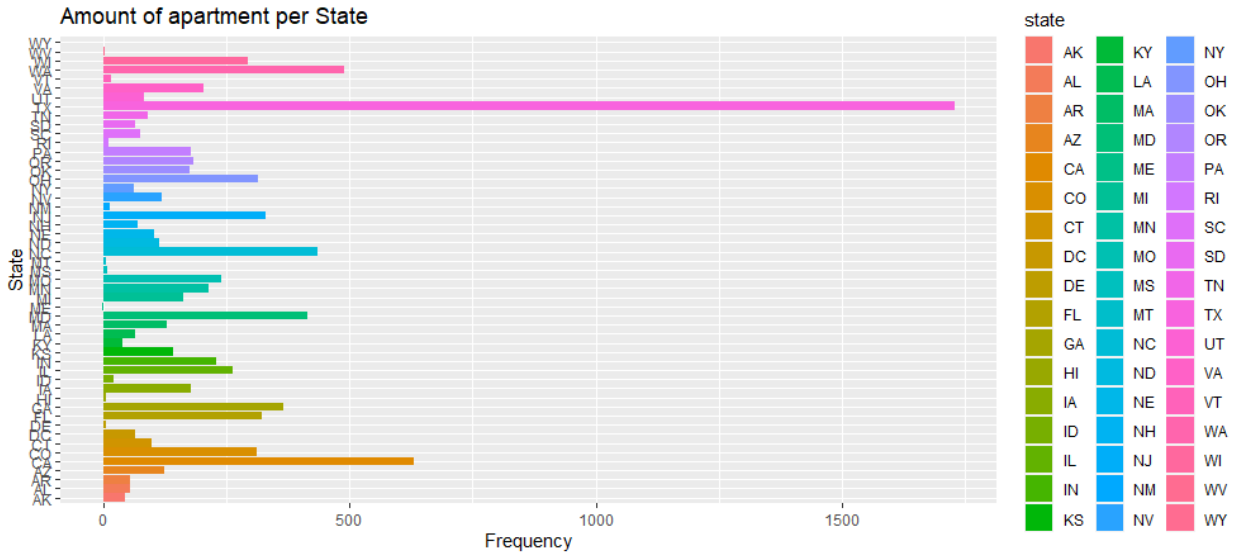


Figure 6: Amount of apartment per State

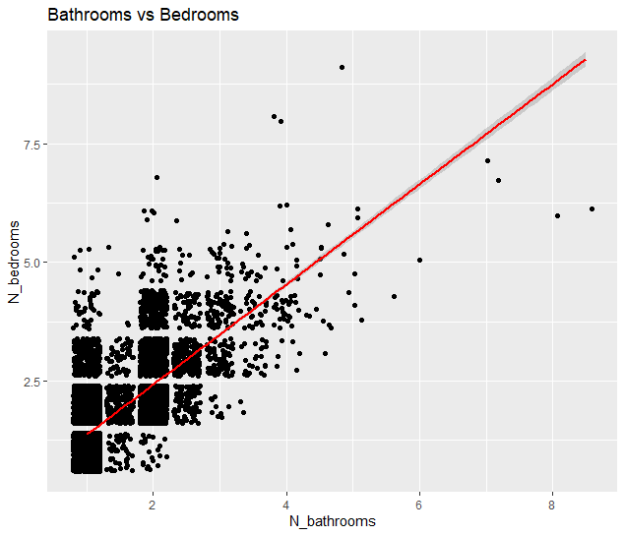


Figure 7: Bathrooms vs Bedrooms

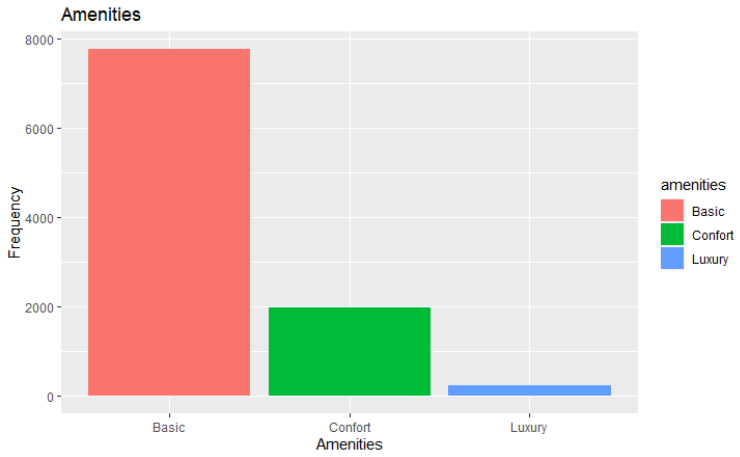


Figure 8: Histogram of Amenities

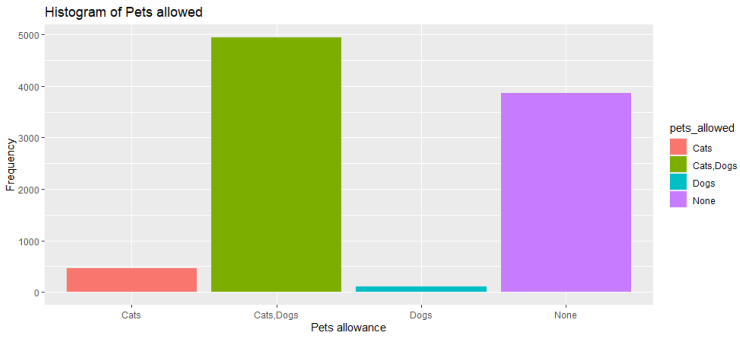


Figure 9: Histogram of pets allowed

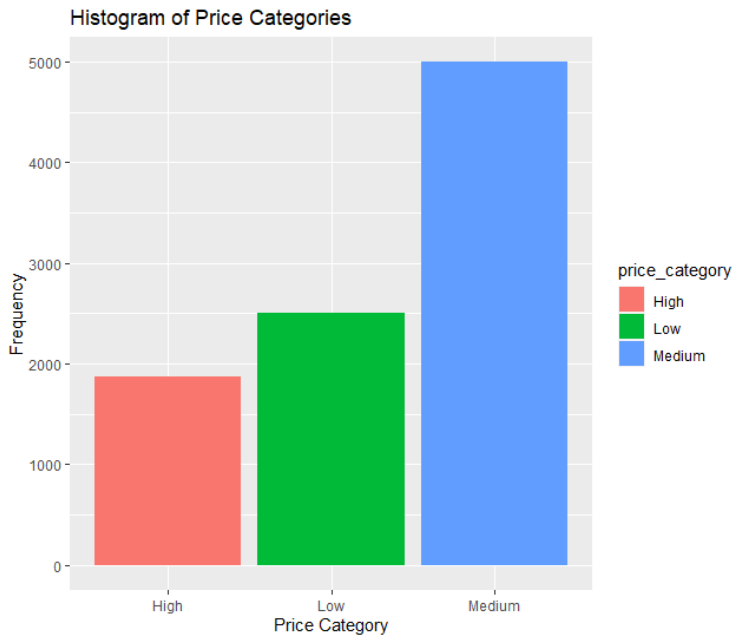


Figure 10: Histogram of Price Categories

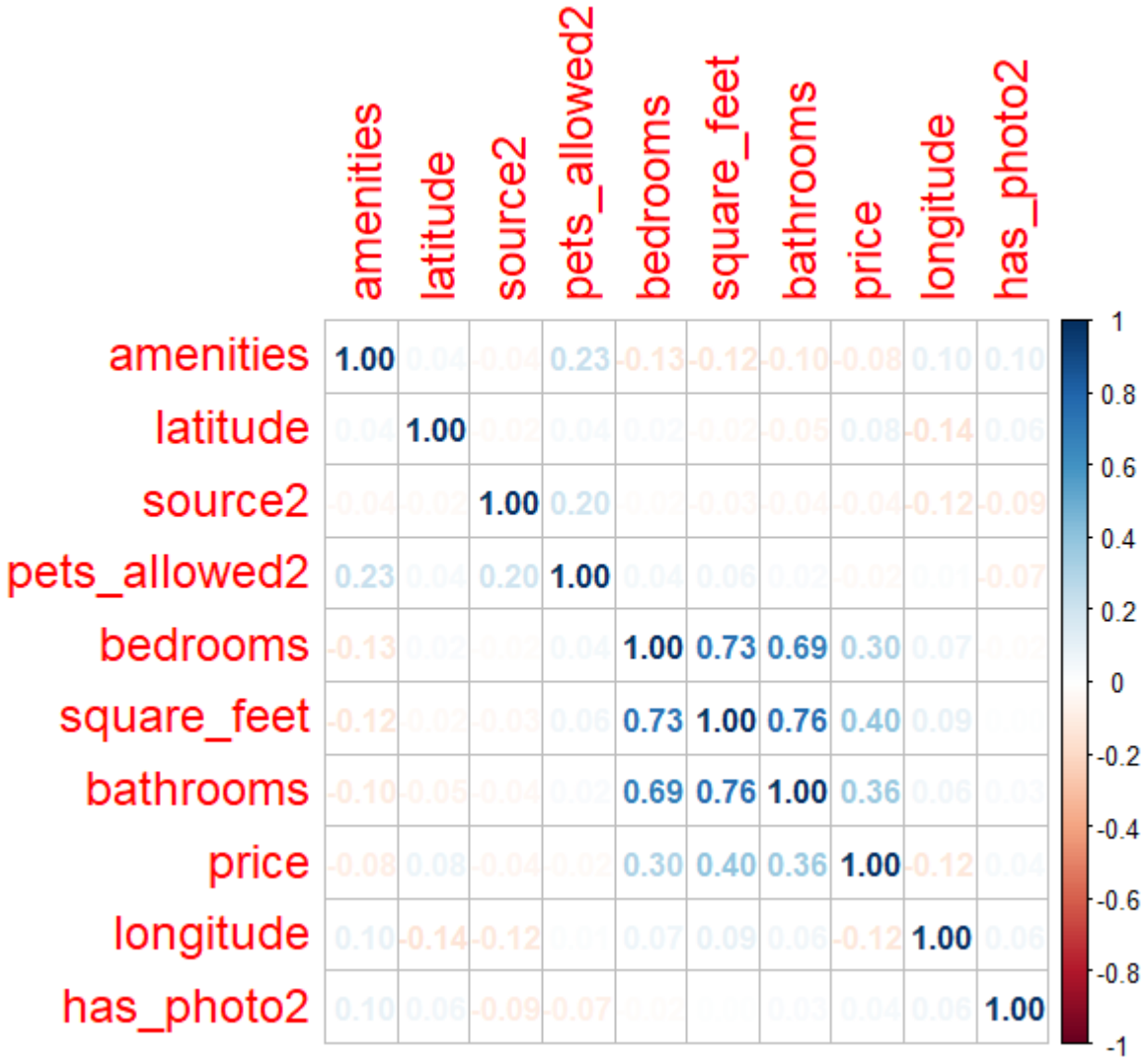


Figure 11: Correlation Plot

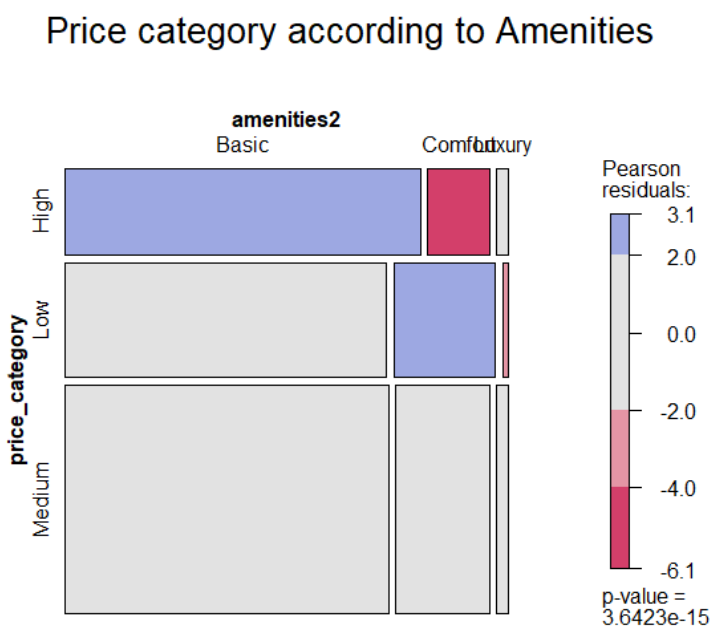


Figure 12: Price category according to Amenities

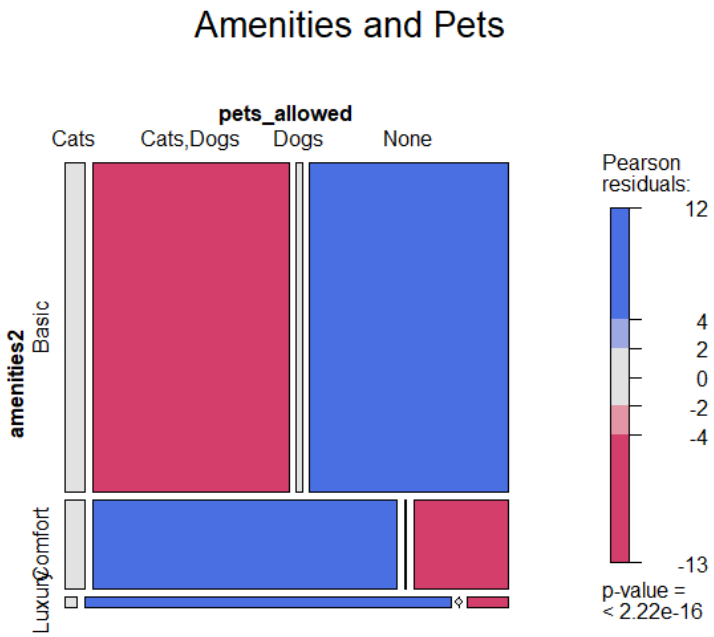


Figure 13: Amenities and Pets

4 Data Mining

After pre-processing and analysing the dataset, data mining algorithms are implemented in this section. Regression and classification tasks are performed using the Caret package. For each applied algorithm, its predictive goodness is assessed according to the target variable (*price in regression, price_category in classification*). In the final section, a comparison is proposed to discriminate which methodology performs best. In order to avoid *NON-ZERO-VARIANCE* problems, 4 distinct sources (*for a total of 6 cases out of 9367*) were dropped from the source variable because they occur only once.

Before implementing algorithms, the initial dataset was split into two datasets: *training* and *testing*, used respectively to train the model and assess its predictive performance. 5 folders Cross Validation is adopted to derive results.

```
AFR <- read.csv("./Files/AFR_10K_for_Data_Mining.txt", sep="")
library(caret)

AFR[,c(2,3,4,7,8,9,10,12,13,15)] <- sapply(AFR[c(2,3,4,7,8,9,10,12,13,15)],as.numeric)
AFR[,c(1,5,6,11,14)] <- lapply(AFR[,c(1,5,6,11,14)], factor)

#remove correlated and derivable variables
AFR <- AFR[,c(1,3,4,5,6,8,9,10,11,14)]

#create variable "Services" as mean of bathrooms and bedrooms because they have high correlation
Services <- as.data.frame(rowMeans(AFR[,c('bathrooms', 'bedrooms')], na.rm = TRUE))
colnames(Services)<- "Services"
AFR <- cbind(AFR,Services)
AFR <- AFR[,-c(2,3)]

#Data Splitting
set.seed(54321)
inTrain <- createDataPartition(y = AFR$price_category, p = 0.75, list = FALSE)
training <- AFR[inTrain,]
testing <- AFR[-inTrain,]

#setup trainControl
fitControl <- trainControl(method = 'cv',
                           number = 5,
                           verboseIter = FALSE,
                           classProbs =  TRUE)
```

For each classification algorithm, the following rational is adopted:

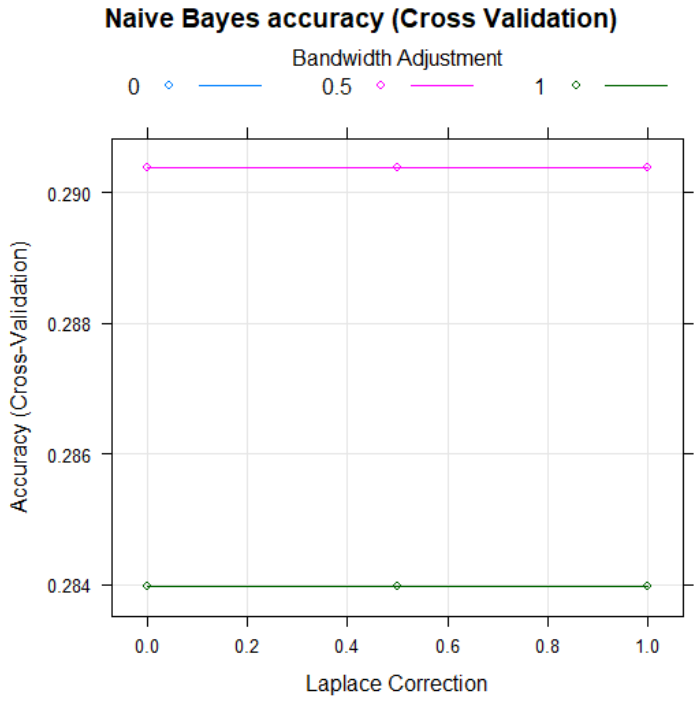
1. **tuning parameters definition:** a small set of tuning parameters is defined;
2. **training model:** the type of algorithm to be used on the training dataset, the target variable and the performance metrics to be adopted are chosen;
3. **accuracy plot:** accuracy is plotted according to tuning parameters;
4. **variables' importance plot** variables' importance is plotted according to output model results;
5. **prediction** model's predictive capacity is assessed on the testing dataset;

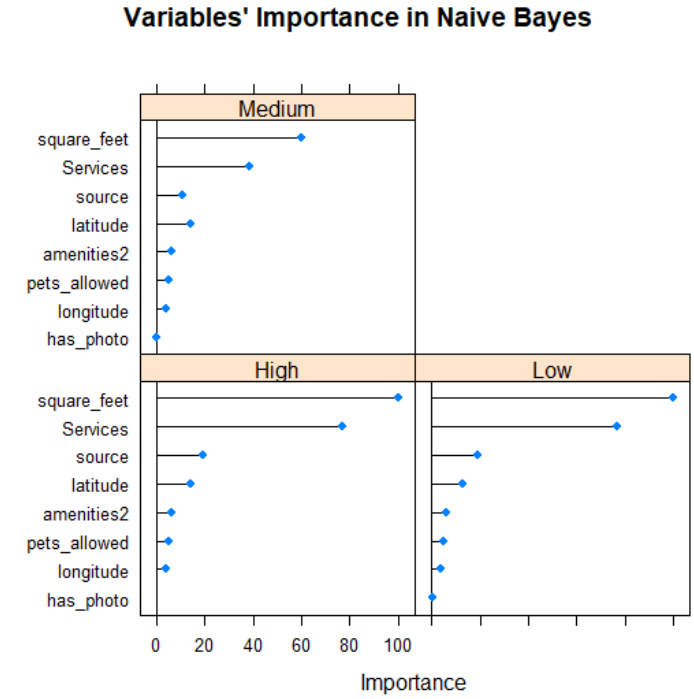
A sample code for Naive Bayes classification is provided in the following section. As far as regression tasks are concerned, a similar rational is adopted even though tuning parameters and performance measures are not explictly defined (*default algorithms setting are preferred*).

4.1 Classification

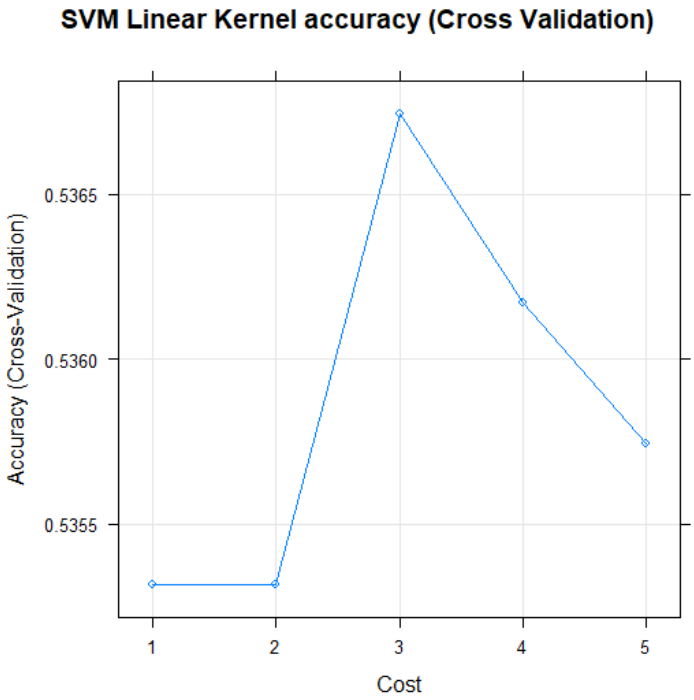
4.1.1 Naive Bayes

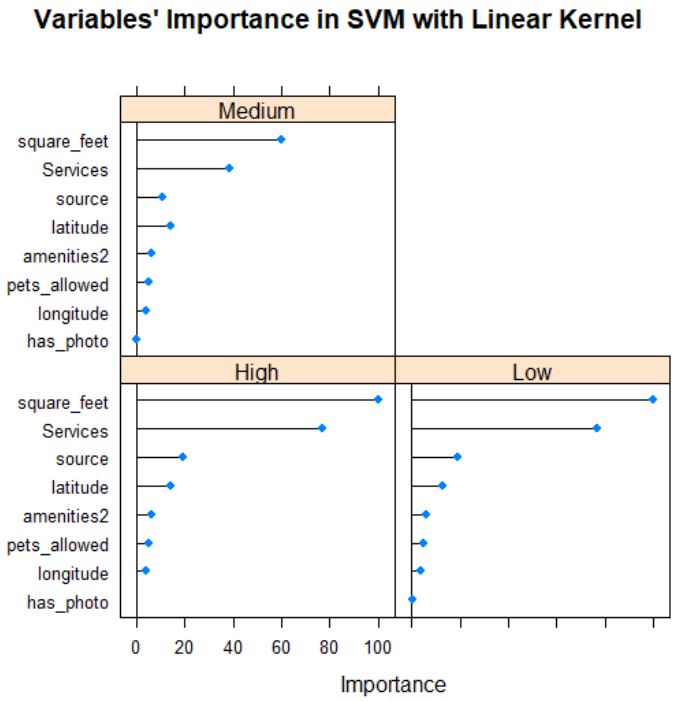
```
library(naivebayes)
#setup tuneGrid
grid <-  expand.grid( laplace = c(0,0.5,1.0),
                      usekernel = TRUE,
                      adjust = c(0,0.5,1.0))

#setup trainControl
NB <- train(price_category ~ .,
            data = training,
            method = "naive_bayes",
            trControl= fitControl,
            tuneGrid = grid,
            preProcess = c("center", "scale"),
            metric = "Accuracy")

#plot of accuracy
plot(NB, main = "Naive Bayes accuracy (Cross Validation)")

#Variables' Importance
VarImp_NB <- varImp(NB, scale = TRUE)
plot(VarImp_NB, main = "Variables' Importance in Naive Bayes")

#prediction
prediction_NB  <- predict(NB,testing)
```



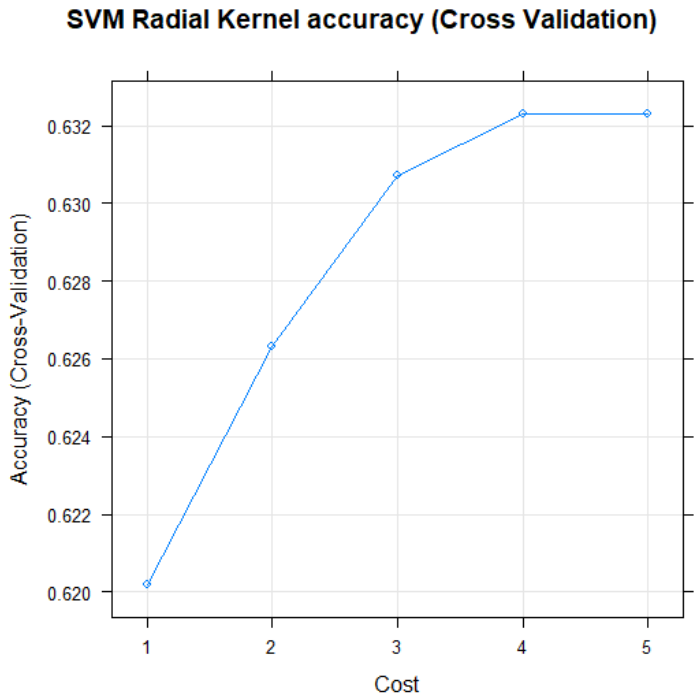


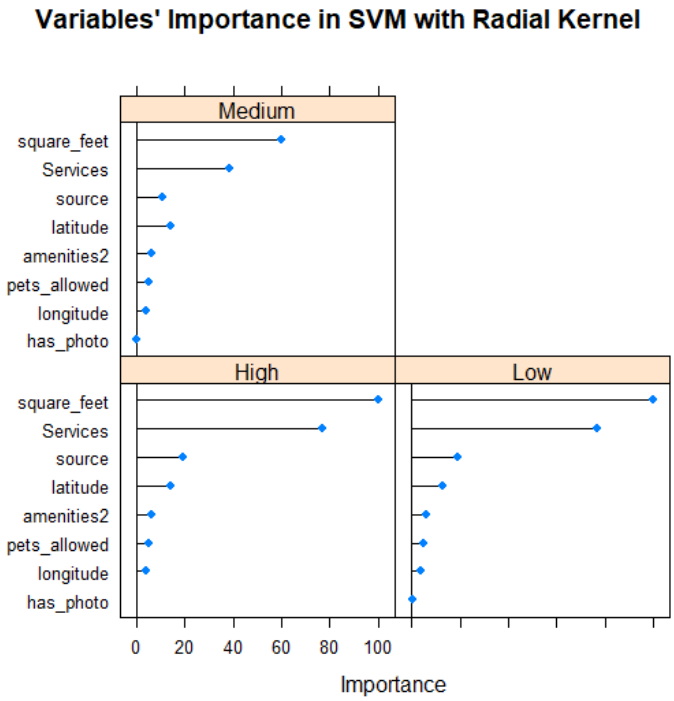
4.1.2 Support Vector Machines with Linear Kernel



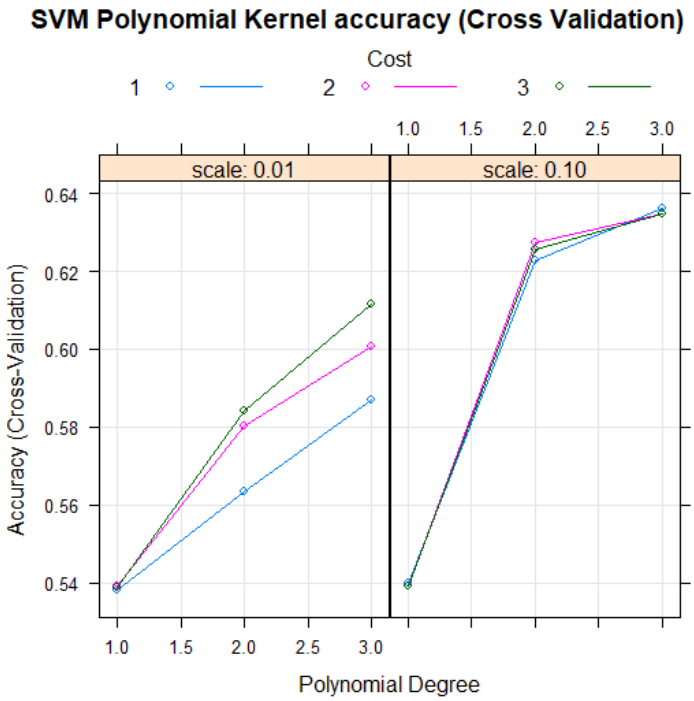


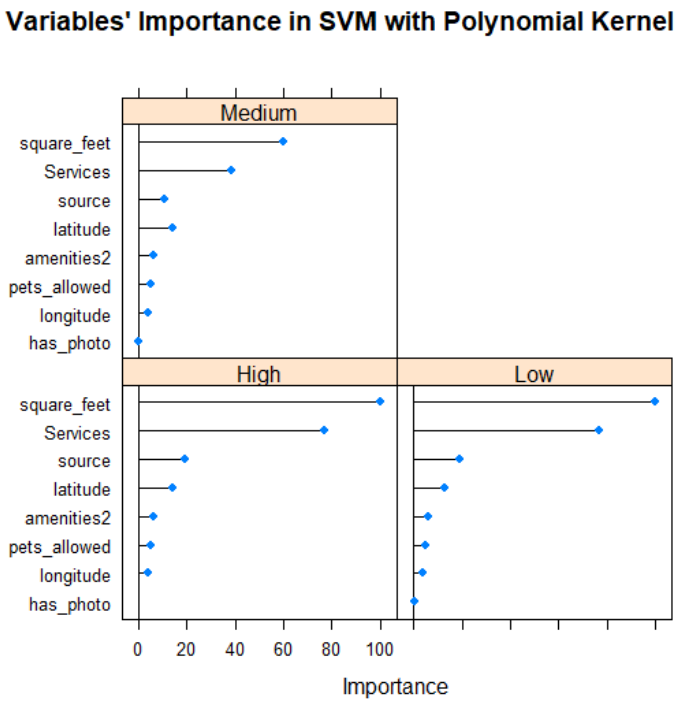
4.1.3 Support Vector Machines with Radial Kernel



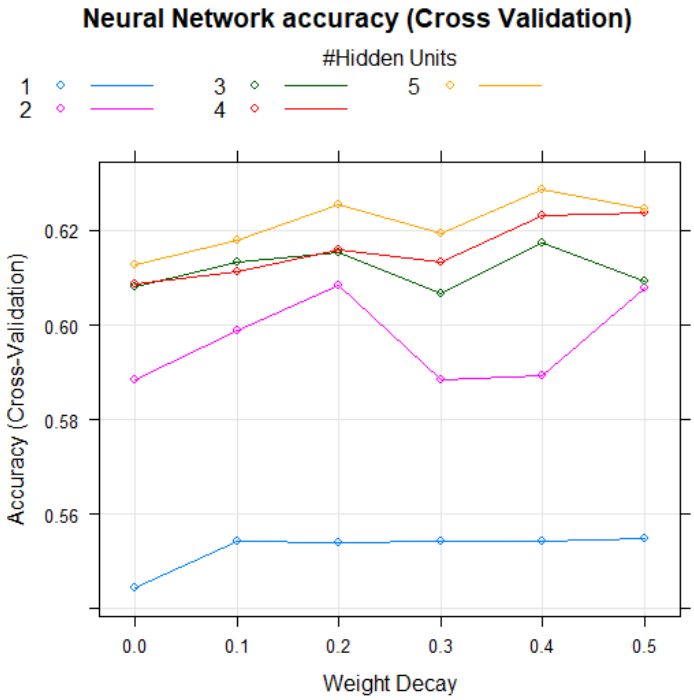


4.1.4 Support Vector Machines with Polynomial Kernel

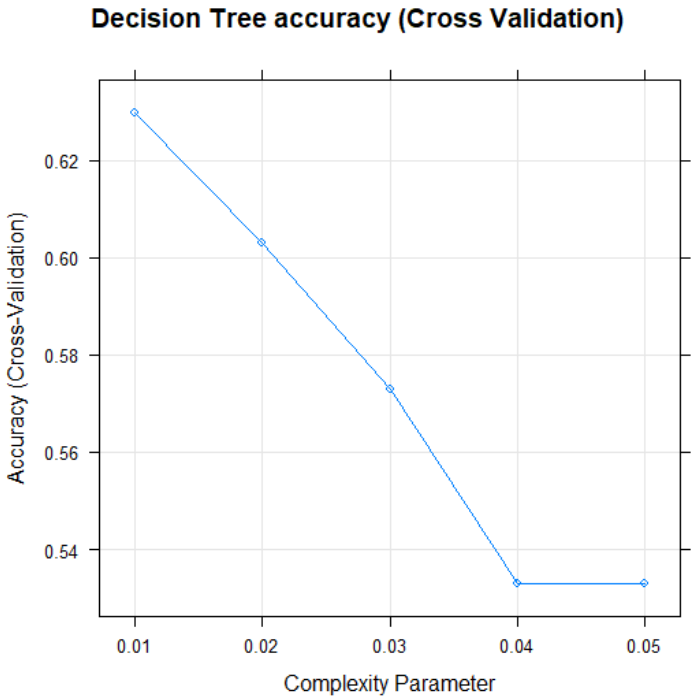


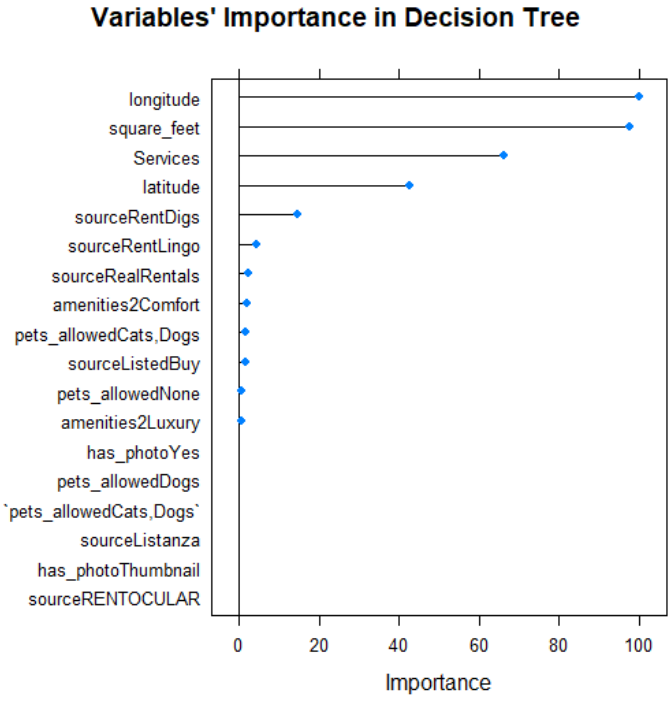


4.1.5 Neural Network

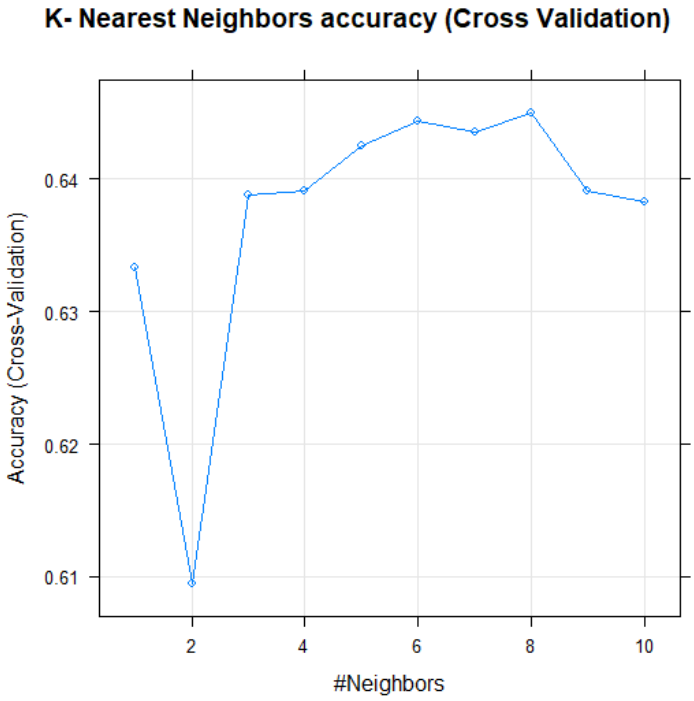


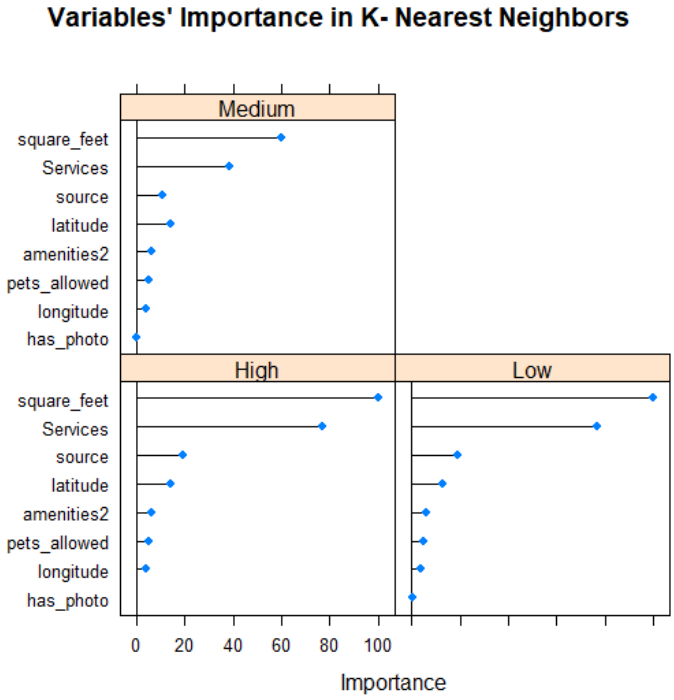
4.1.6 Decision Tree



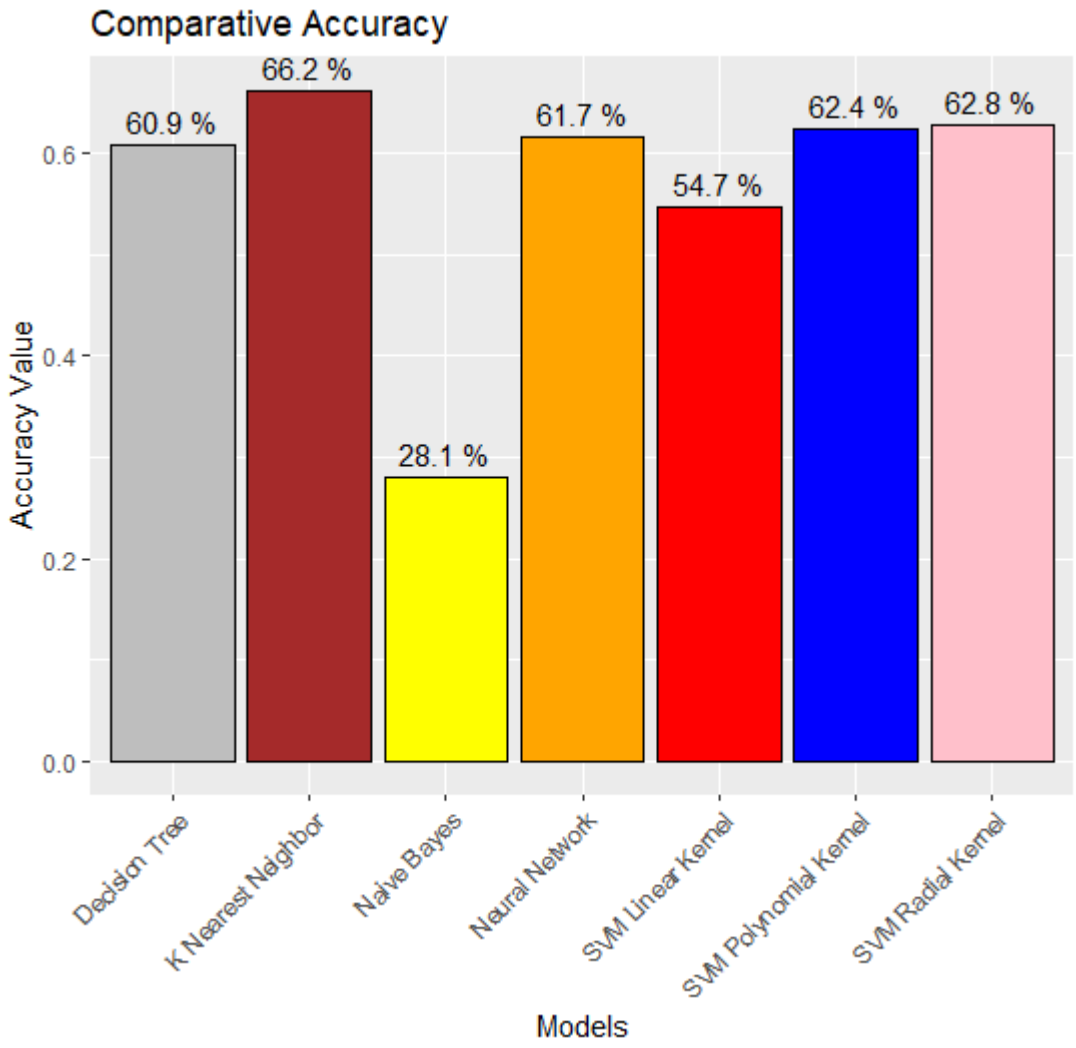


4.1.7 K- Nearest Neighbors



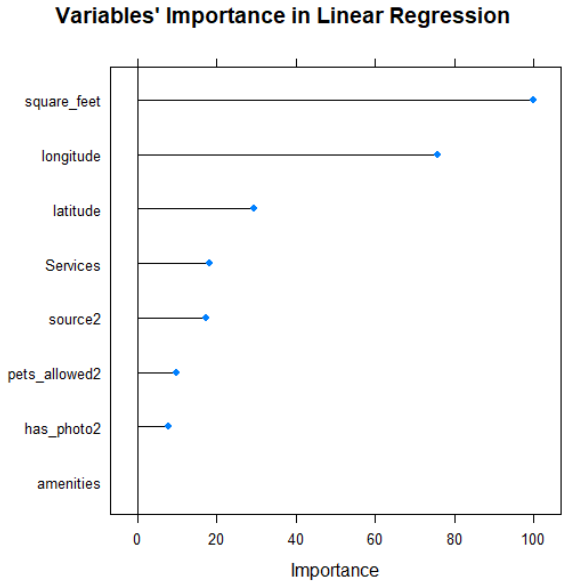


4.1.8 Classification models accuracy comparison

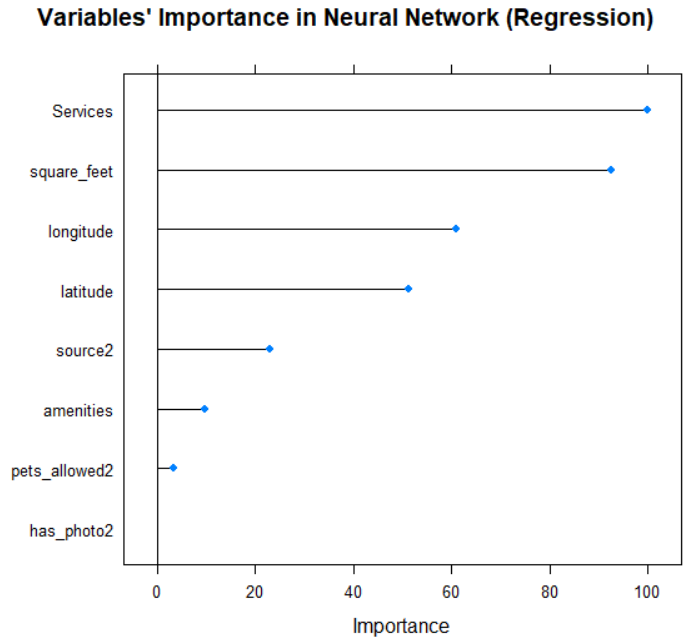


4.2 Regression

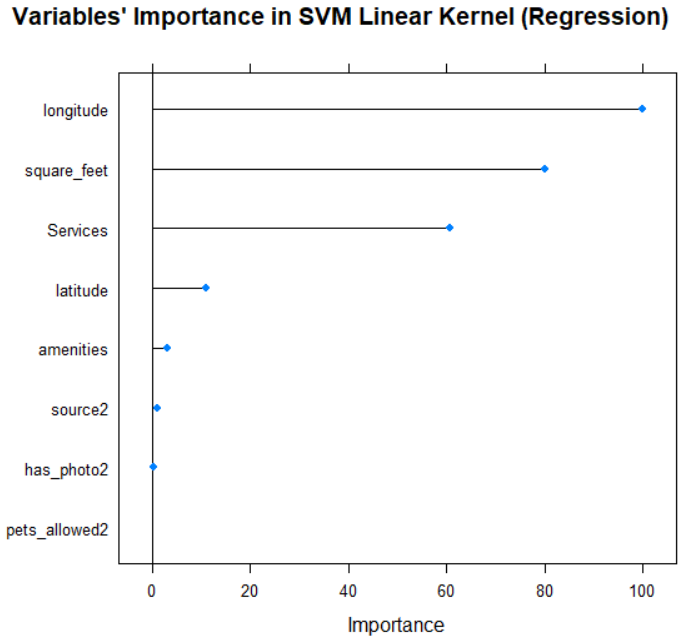
4.2.1 Linear Regression



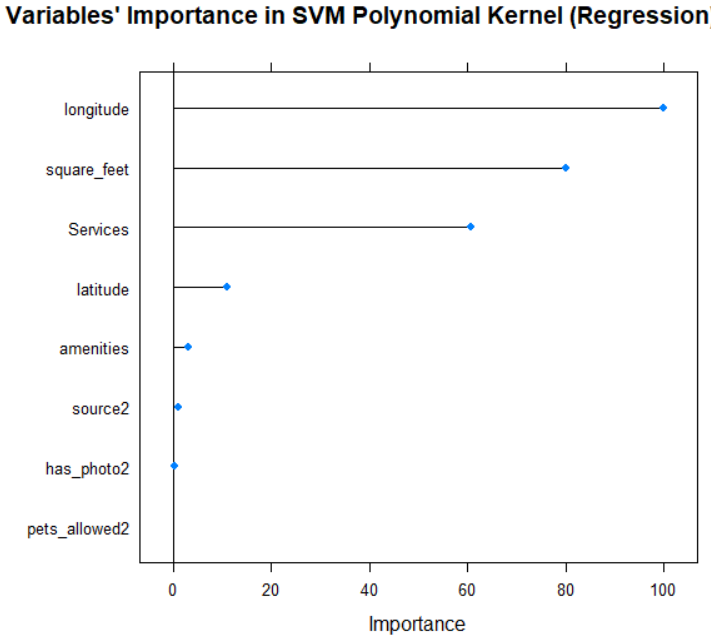
4.2.2 Neural Network



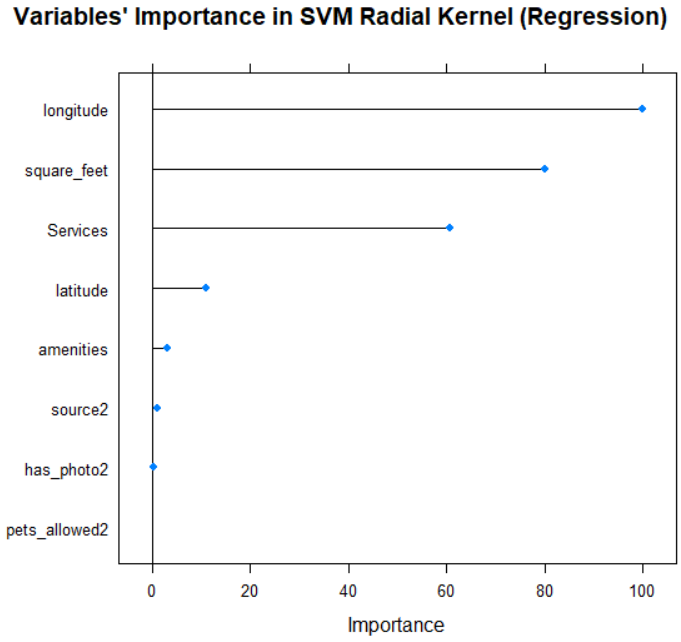
4.2.3 Support Vector Machines with Linear Kernel



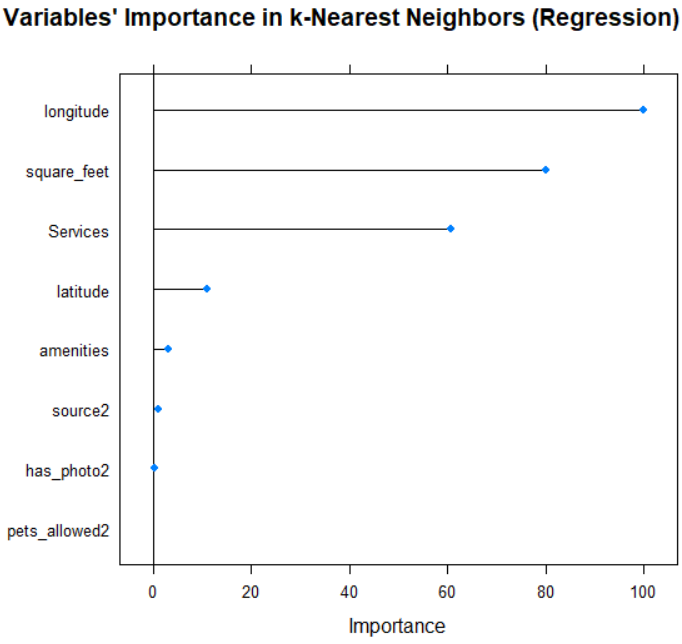
4.2.4 Support Vector Machines with Polynomial Kernel



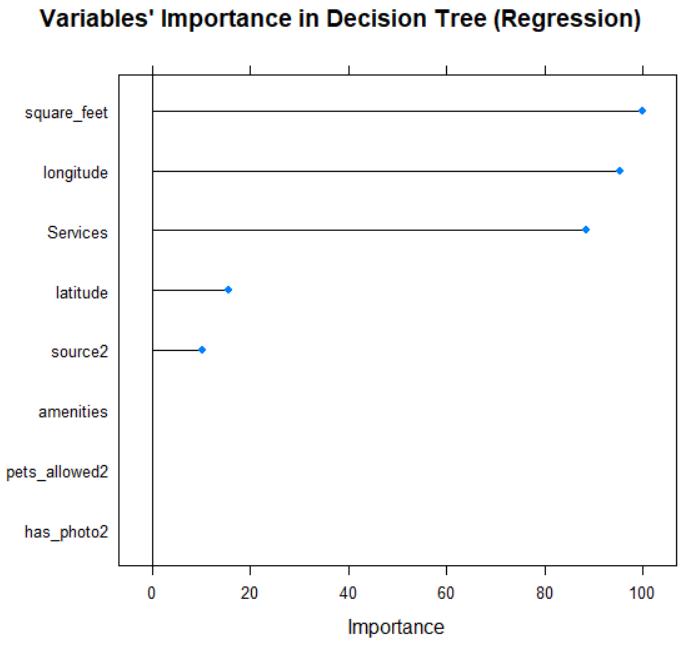
4.2.5 Support Vector Machines with Radial Kernel



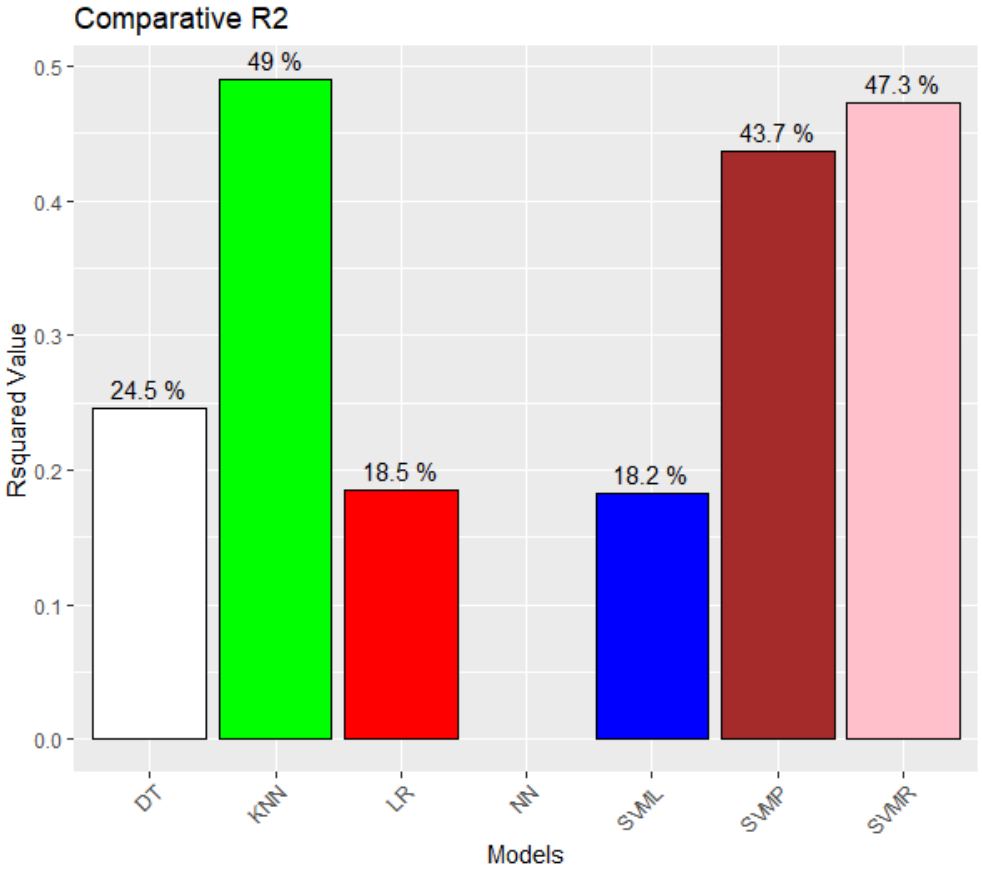
4.2.6 K-Nearest Neighbors



4.2.7 Decision Tree



4.2.8 Regression models R^2 comparison



5 Conclusion

As depicted by plots above, regression models do not perform well in determining the rental price (R^2<50%). However, different results are performed by classification models which are used to predict the price range. In fact, with the exception of the Naive Bayes algorithm, accurancy range is 55%-66%, with a good predictive performance. Among all models, the best performance values occur with the K-Nearest Neighbors.

Better results may be obtained as follows:

1. replace the method from Cross-Validation to Repeated Cross-Validation;
2. consider more tuning parameters for each model;
3. increase the number of records (*use the 100K dataset*);

Unfortunately, due to the limitations imposed by the device performances and the associated computational complexity, it was not possible to proceed with the implementation of such improving actions.