Prediction of housing rents in Copenhagen

Anne Sophie Schytt Lassen

Philip Lemaitre

Andreas Langholz

Dominik Lillemæhlum

15 December 2015

1 Introduction

This paper aims to use a variety of methods from the field of data science to gather, describe and predict rental prices of apartments in Copenhagen posted on boligportalen.dk. The paper is structured into three sections, data gathering, data description and modelling.[1] In the first section, we briefly describe how we scraped data from a Danish web page, what type of data emerged and what ethical challenges we should consider. Moreover we included a word matrix in order to conduct an analysis of the effect on price from the description and title of the post. In the second part the dataset is described using statistics and graphics in order to get an overview of the data and the variables which affect the rental price. In the third and final part we use statistical learning models to predict rental prices of apartments in Copenhagen. The emphasis will be on testing different models and comparing their predictive power on a test set. We find that when comparing a linear regression model, a lasso model, a support vector machine, a regression tree and finally a random forest. It is concluded that all models has roughly the same prediction power.

2 Data gathering

The source of the data used for this paper is the Danish online portal for apartment renting Boligportalen.dk[2]. This website only contains information on apartments that are rented out - not on people searching for housing. As we only look at the supply side of the market for rented apartment, the data on this webpage is very relevant. The data was extracted by a web scraper, which was build to the purpose. The scraper ran through all posts on the web page and saved the data as a text file. This was done every day for 21 days, creating 21 data sets. The data sets were then merged into one file, and striped off unnecessary variables used as predictors. This also implies, that our data can not be reproduced as posts might have been removed from the web page. The most important part of the cleaning process, was to make sure that all observations

[1] All calculations, graphics and writing was conducted using the programming language R and the IDE RStudio

[2] http://boligportalen.dk

only occurred in our data set once, as most observations was scrapped on multiple days. By construction, this means that our final data set is cross-sectional.

When scraping data from a website owned by a company there are certain ethical issues to consider. Since the data from boligportalen is used by the company to generate profits, one can make the argument that they have the right to privacy in order to preserve informational rents. This project and the used data from Boligportalen will not be published in any way that would conflict with this right. Moreover university assignments no need for an informed consent.[3]

However an ethical issue of a very different type arise from the fact that we use the function get_map from the package ggmap. When loading the package, one must accept Google Maps API Terms of Service,[4] including having a registered Google-account. This means that Google has access to our requests and can link them to our profile. In principle, this gives away parts of our collected data set to a third party agent, without knowing how they would utilize such information, and whether it is in accordance with the ethical and legal constraints of our project.

2.1 The final data set

The data scraped from boligportalen.dk contains the following variables after the cleaning: `address`, `longitude`, `latitude`, `zip code`, `name of neighborhood`, `rent` (measured in DKK), `size of apartment` (measured in square meters), `price pr. square meter`, number of `rooms`, the `description` and `title` that the owner has given and an `id` number. Number of rooms can take the values 1, 1.5, 2, 3, 4, or house. The value 1 indicates that it is a rented room in a bigger apartment, and 1.5 indicates that it is a 1-bedroom apartment.

Moreover, the variable meter has been added. The variable is the distance to City Square Hall. This is included as distance to the center of Copenhagen might affect the price and therefore the distance to City Square Hall is calculated for each observation. This is done by using the function get_map from the package ggmap. This function output distance and travel times from each address in our data set to the City Square Hall, as measured by Google Maps.

To make our predictions as precise as possible, we also seek to include a textual analysis as part of our overall pool of regressors in the modelling. In our data set, two variables have textual characteristics, these are the title of the post, and the description of the apartment. We have chosen to pool these together in a variable named `TotalDescription`, under the assumption that there is no direct difference in the chosen wording in the title and description. In order to make predictions based on variance in the wording of the descriptions, we then split the description into a bag-of-words as well as a set of principal components, which will later be applied in the general modelling.

[3] Danish Data Protection Agency (in Danish)
[4] https://developers.google.com/maps/terms

2.2 Bag-of-Words

To make predictions based on the wording in the description, a binary matrix was created using the package `quanteda`. The matrix contains information if specific words such as kichen, light, balcony (all in Danish) were present in the the description of an apartment. Next, the matrix is sorted by omitting general stop words of the Danish and English language (frequently used words with no contextual meaning, such as "the" and "is"), a vector of words chosen by us which is not seen as having any significant effect (this could e.g. be "lejlighed" or the month of the post) and words occurring in less than 5% of the texts. To further subset the text into valid predictors, we apply the lasso regression model (explained in detail in the modelling section), to return a vector of words which has been tested as valid predictors of rental prices on a test set in the data set. Sorting the matrix by the output of the non-zero lasso coefficients the final data-frequency-table is obtained, and transformed into a data-matrix to be further collapsed into principal components below.

2.3 Principal components

In order to construct more compressed predictors as well as making inference on the effect of different wordings on rental prices (as well as getting around a reccuring problem of differing factor levels when splitting the modelling set) principal component analysis is applied. Principal component decomposition is in its essence a way of collapsing the wording-matrix into components of less dimensions explaining the majority of variance in the wording. When mapping the wordings on the principal component axis's, it can be observed how certain words varies and co-varies in the different dimensions, as well as the tendency of grouping of certain words along the dimensions.

Initial inference might suggest that especially along the y-axis, words describing locational characteristics ("taet","beliggende","ligger","frederiksberg","koebenhavn") has a positive effect on the rental price, while internal descriptions of the apartment ("koekken","stort","moebleret" etc.) has a negative effect. Along the x-axis, most words are grouped slightly below zero, except a cluster formed around 0.4. The words here are easier read in the 3-dimensional plot in the appendix. The cluster with positive effect on rental price along the x-axis mainly consists of words describing overall features of the apartments such as "indflytningsklar", "ny" and "etageejendom".

The twelve largest principal components is then returned and merged together with the dataset giving us the text based predictors we want to apply in the model section. The final data set thus contains the variables of interest from boliportalen.dk, distance to the City Square Hall in Copenhagen measured in meter and the principal components from the textual analysis.

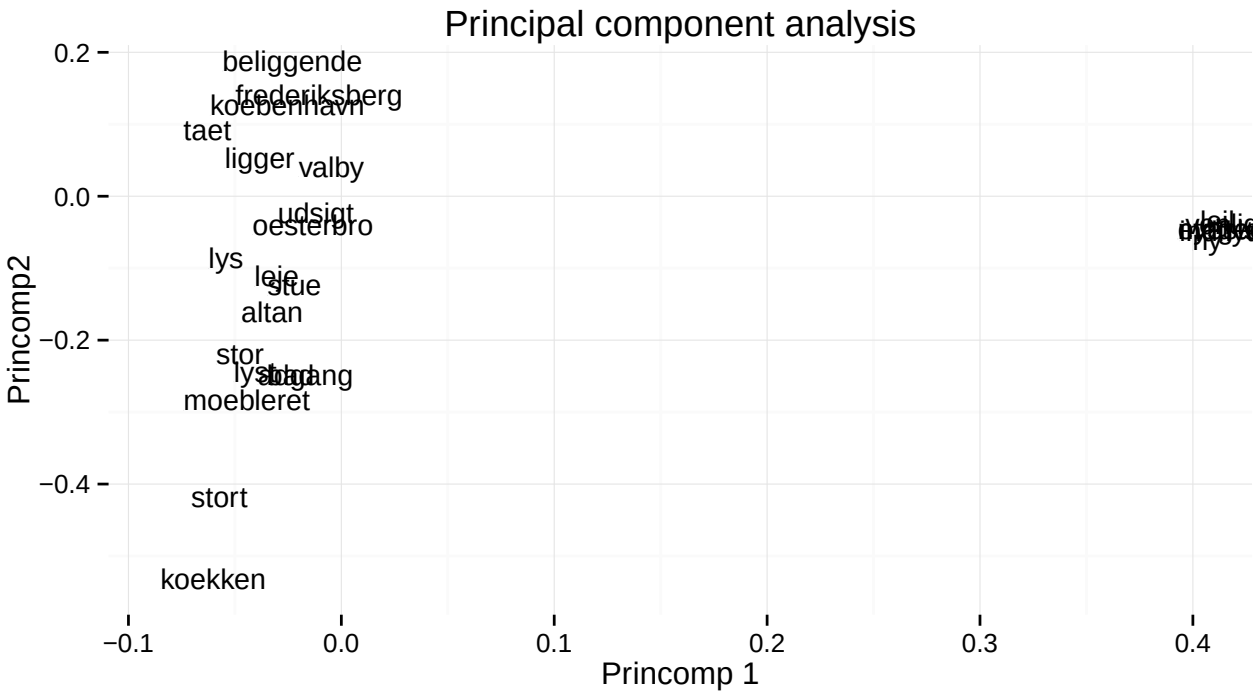

Figure 1: Principal component 1 and 2

3 Describing the data

Before continuing with the prediction models, we should take a closer look at the data. Our data set includes 1031 apartment distributed in the 17 postal areas that are defined as located within Copenhagen. In some areas we have quite many observations (202 in Copenhagen SV and 177 in Copenhagen S), but in some areas we have very few (8 observations on Frederiksberg and 13 in Copenhagen K around Østergade). Plotting the rental price against the size of the apartment, it is clear that there is a positive relationship, see figure 7 in the appendix. This is not surprising, but confirms that our dependent variable in all models should be price pr. square meter rather than absolute rental price.

When examining the distribution of the rental price pr. m^2, see figure 3, it is clear that distribution is right-skewed, meaning that the data set includes some observations with a very high price. Including all observations the average price pr. m^2 is 195.9 DKK. If the observations with a m^2 price above 500 DKK is excluded (19 observations), the average price falls to 187,1 DKK.

In order to investigate why some observations have a very high price pr. square meter, a simple plot of the size of the apartments is plotted against the price pr. square meter. This plot is shown below, figure 2. This plot shows, that the smaller apartments (approximately below 20 m^2) has a very different distribution that the rest of the data. This result holds even if we remove apartments with a rental price pr. m^2 above 500 DKK, see figure 8 in appendix. For apartments smaller than 20 m^2 the mean price is 343,5 DDK. However, for the apartments bigger than 20 m^2 the average price falls to 154,3 DKK. Different number of rooms is

marked by different sizes of data-points, and the data seems to suggests, that the price setting for small apartments or rented rooms is different that the price setting for bigger apartments and houses.

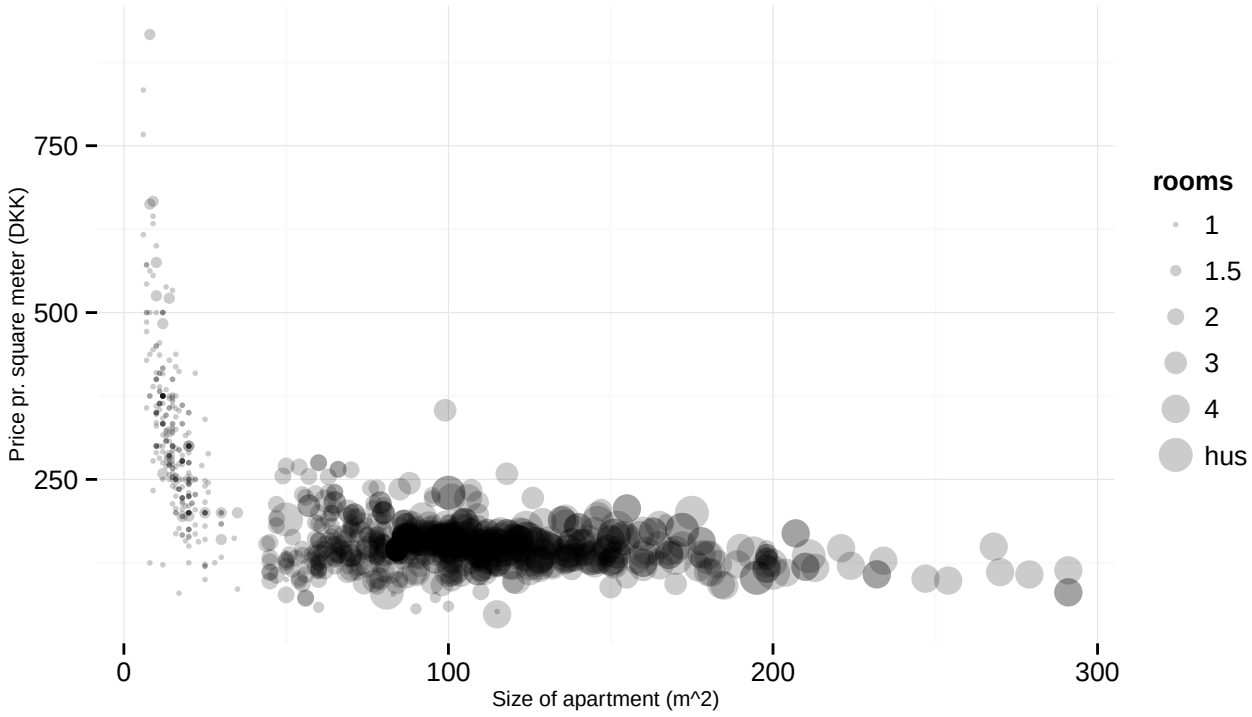

Figure 2: Apartments against the price pr. square meter

Earlier it was argued that distance to City Square Hall might affect the price. Plotting distance in meters against the price pr. m^2, see appendix figure 9, it seems not to be the case. However, this plot also shows that houses and big apartments are priced at a relative constant price pr. square meter, whereas the pricing for smaller apartments and rooms seems to be more volatile.

Another variable this might affect the price pr. square meter is which neighborhood the apartment is located in. A bar chart of average price in different zip codes is illustrated in figure 6, and we observe great difference. The cheapest neighborhood is København K around Østergade (1100) where the average price is 146.2 DKK, while the most expensive neighborhood is København K around Borgergade (1300) where the average price is 280,9.

However, both of these groups has below 20 observations, so these results might be due to the small sample size. The average price pr. square meter is also illustrated by using a map of Copenhagen see appendix, figure 10. There seems to be no strong relationship between average price in adjoining neighborhoods, which also might indicate that our sample size is too small in some areas.

Even though the sample size is small, a map of Copenhagen with all observations plotted, showing the respective price pr. square meter, gives a clear picture of the variance of the data. This is shown below and it is clear that the variance within neighborhoods is quite big.

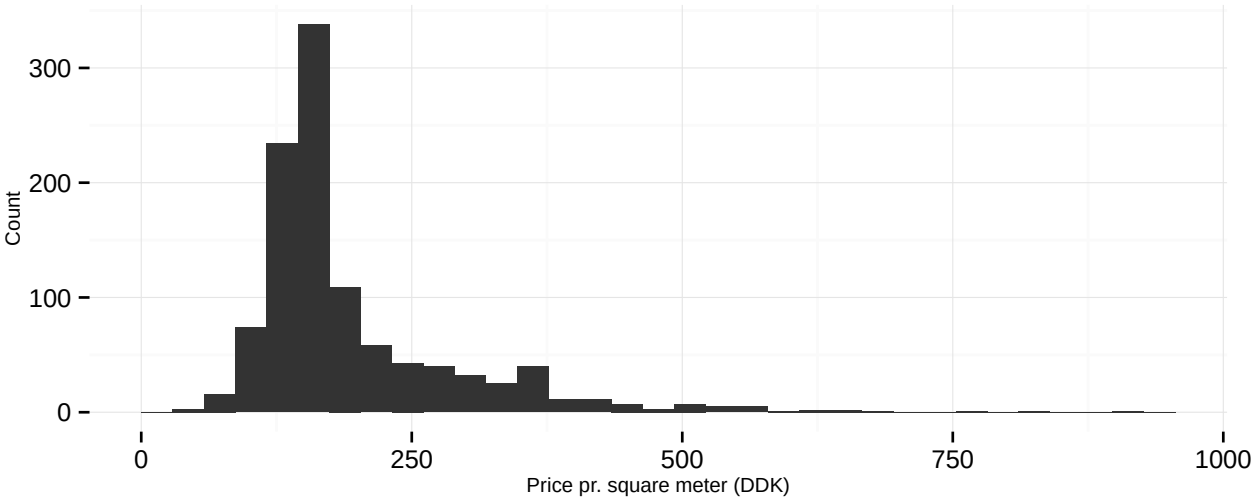

Figure 3: Distribution of price pr. sq. meter

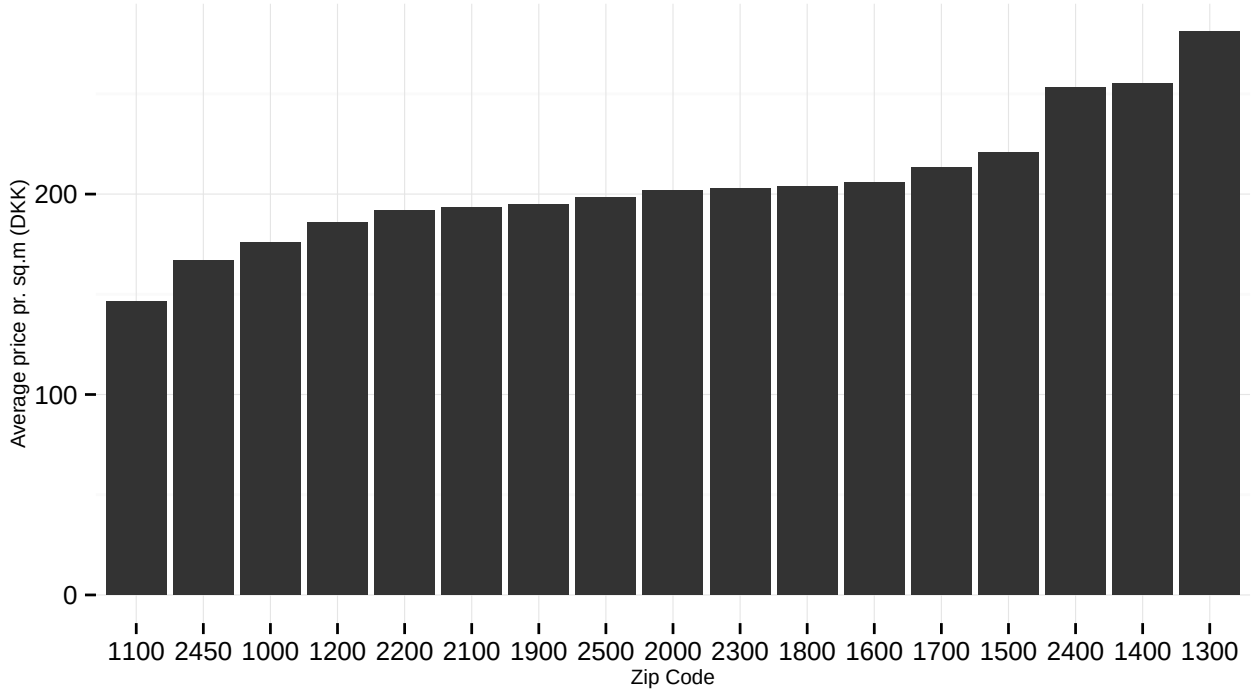

Figure 4: Apartments against the price pr. square meter

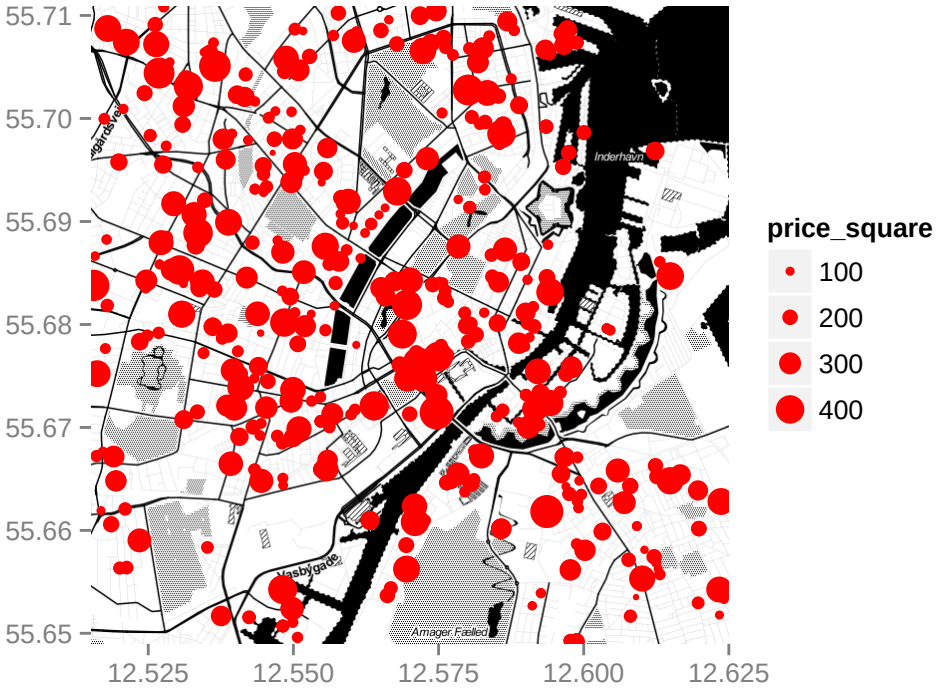

Figure 5: Price pr. square meter in different neighborhoods

From here, it is clear that we must conduct a thorough analysis of what determines rental price pr. square meter. Besides location in terms of `zip code`, `type of housing` seems to have an impact as well as absolute size of the apartment.

4 Data modeling

The purpose of the section is to illustrate different ways to make predictions of the rental price pr. square meter. This is done by applying the approach of positive economics formulated by Milton Friedman, where models are "to be judged by the precision, scope, and conformity with experience of the predictions it. In short, positive economics is, or can be, an"objective" science, in precisely the same sense as any of the physical sciences."(Friedman 1953)

Our base model will be a simple average, which will be compared to OLS, lasso, support vector machine, Regression Tree and Random Forest with respect to ability to predict. In order to do so, we divide our data set into a training and a test data set. First we train the models on the training data set, and afterward the model is tested on the test data set.

When we want to predict, we want to minimize the out of sample error. In order to evaluate and compare the models, we compare the ability to predict measured by root mean square error. The model will be evaluated by the below quation, which return RMSE as a measure of how good the models are at predicting the values

from our test data set:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$$

Prediction error can be decomposed into two categories: errors due to bias and errors due to variance. Dealing with bias and variance is really about dealing with over- and under-fitting.(Scott, n.d.) When bias is reduced, the variance is increased in relation to model complexity. As more and more variables are added to the model, the complexity increases and variance becomes our primary concern, while bias steadily falls. When trying to understand the behavior of prediction models, it is critical to understand the trade-off between bias and variance. Even though, we care about the overall error and not the specific composition the following expression is helpful:

$$\frac{dBias}{dComplexity} = -\frac{dVariance}{dComplexity}$$

This tells us, that the sweet spot for any model is the level of complexity at which the decrease in variance is equivalent to the increase in bias.

4.1 Simple average

Our base model is a simple average. The average price in our training data set is 198,71. This gives us a RMSE of 99,88, when we compare to our test data set.

4.2 OLS

When applying OLS, we minimize the in sample error, which also minimizes the bias. This is not an optimal solution when prediction is the goal, as the risk of overfitting increases. The main strength by using OLS is inference. Each parameter in our model can be interpreted and tested for significance. Using OLS gives us a RMSE of 78,67.

4.3 Lasso

The Lasso regression model is based on the traditional OLS regression but extended by adding a Loss function to make sure that the model is punished for overfitting the variables. In the Lasso, the loss function is stated as the sum of absolute beta values, which returns a corner solution, of only the most significant predictors in the model. The optimization problem hence takes the following form:

$$\text{Minimize: } \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Finding the optimal weighting of the loss function λ, we loop the predictions of the Lasso model on our test set with different values for λ, returning the λ value with the lowest root mean square error (See appendix for RMSE returns for different λ's). The final model with the lowest λ returns a root mean square error of 82,2

4.4 Support Vector machine

Support Vector Machine Moving away from linear based modelling, we now turn to the Support Vector Machine (SVM) model. The SVM is usually applied in a classification problem but can also be applied in a regression process predicting continuous outcomes, known as support vector regressions. The basic intuition behind SVM models is to create a classifying vector that splits the data by maximizing the Euclidean distance between the points and the vector, known as the margin. When expanded beyond the two dimensional case, the vector becomes a K-1 dimensional hyperplane for K dimensional classification problem. This can be done with both a linear and a non-linear vector, as well as expanded to continuous outputs as is the case of this paper. Using the e1071 library, the SVM used here applies a "one-versus-one" approach to subdivide the continuous spectrum using several support vectors until further classification is pointless, and using the subdivision to assign prediction values. As always, we then test the model performance on a test set, and return the RMSE value 78,81.

4.5 Regression Tree

When trying to find patterns in data and using this to predict rental prices, machine learning is a helpful tool. An example of a machine learning method is a decision tree. A subset of decision trees are regressions trees which allows the target variable to take continuous values. A regression tree is a way of constructing conditional probabilities. The way the regression tree constructs the conditional probabilities is by dividing the data into different regions. The regions are created so the RSS given by the equation is minimized:

$$\sum_{m=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_J})^2$$

This is will lead to overfitting, therefore the following equation is used:

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T|$$

Where $|T|$ is the number of nodes in tree T, R_m is the subspace of the region M, and $\hat{y}_{R_m}$ is the predicted response associated with R_m. Adding additional nodes will increase the tree's prediction accuracy, but also the complexity and thus the variance. As mentioned before there is always a trade-off between bias and variance, and when using decision trees, the approach is called pruning. The benefit of pruning a tree is a more interpretive model, but at the expense of a higher bias. Thus we handle the problem with overfitting the parameter in this case, by adding a punishment for more complex trees and thus obtain a sequence of best subtrees as a function of α. The parameter α is choose by using K-fold cross-validation on the training data set. More precisely the training data set is divided into K-folds where k=1,2,3...K. Then for each value of α a new tree is grown on all but the the kth fold of the training data. Then the MSE is calculated on each tree as a function of α. Afterwards the average of MSE for each α is calculated and then the α with lowest average MSE is pick to make the final tree.

The RMSE from this model is 84,24

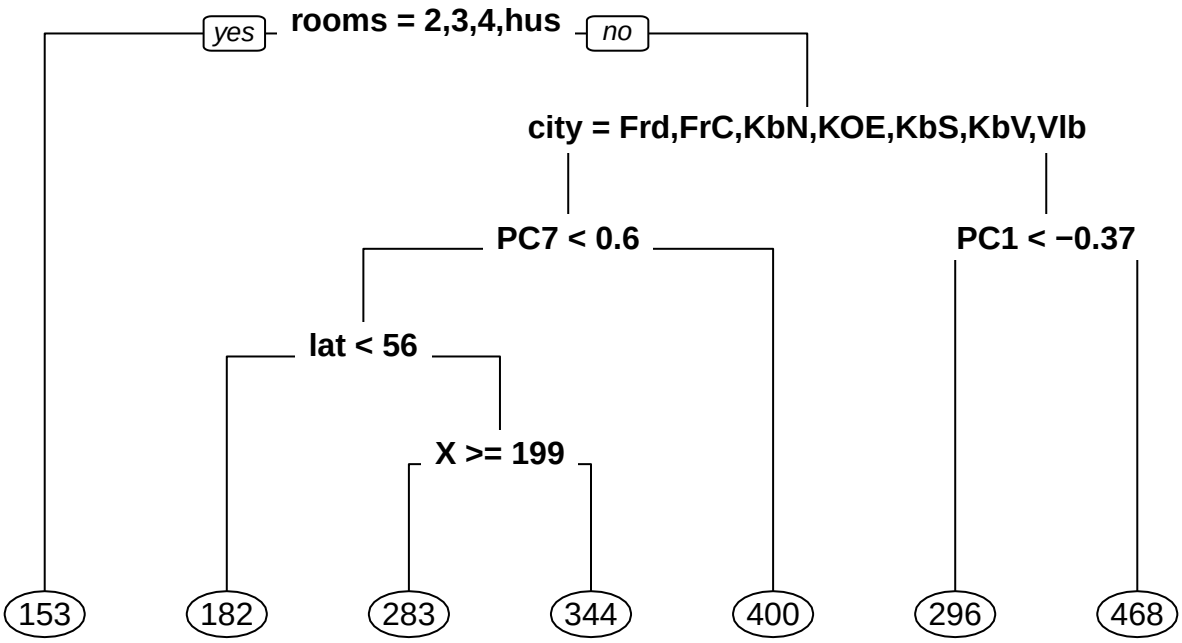

Figure 6: Tree

4.6 Random Forest

As mentioned, one of the disadvantage of the decision tree model is often a high variance and thereby bad prediction. This can be solved by applying Random Forest algorithm, which is an ensemble model based on submodels which are either decision or regression trees. The perfect way of minimizing the variance, would to have large number of train-set and construct a model on each set and averaging their predictions. This is not possible and therefore bootstrapping is applied. The principal of bootstrapping is to increase the information about a distribution of variable in population by resampling a sampling of the population. The resampling

is done by, taking N observation of the original sample with N observation, but with replacement. The replacement makes it possible to draw the same observations several times, making it possible for different bootstrap samplings. Then for each bootstrap sample the mean is calculated and when this is done 10.000 times, there is clear distribution of the mean. By applying this method to the regression tree, it is possible to construct B trees on B bootstrap samples of the training data set. These trees are not pruned and their predictors will therefore have a low bias and high variance. This procedure is call bagging and is basically bootstrap aggregating. The core idea is to decrease the variance of the predictions of one model by fitting several model and averaging over their prediction.

Using the bagging procedure, there is a risk of having high correlation between bagged trees. This occurs when there are one or more strong predictors in the data set, which will lead to all the trees having the same top node split. There is a risk of highly correlated predictors, and the variance in our model is not reduced as much, as if we has low correlation between the predictors. When applying RandomForest only a random subset of the explanatory variables are used in each node tree. This is done because strong predictors tend to overshadow the effect of weaker predictors, because the algorithm searches for the split that results in the largest reduction in the loss function. When only a subset of predictors are available to be chosen, weaker predictors get a chance to be selected more often, and thus reduce the risk of overlooking these variables. This gives the possibility of different trees constructed on different predictors, thereby lowering the variance of average predictor of the trees. The RMSE from this model is

RMSE: 77,55

4.7 Summary test scores

Model	Root mean squared error
Base model, average	99,88
OLS	78,67
Lasso	82,2
Support Vector Machine	78,81
Regression Tree	84,24
Random Forest	77,55

5 Conclusion

In this paper we have presented data scraped from the website Boligportalen.dk. First we described how the data was collected, then we described some dependecies between the variables mainly through graphics.

Lastly we made an atampt at predicting one variable: apartment rent through 5 statistical learning models. Our measure of comparison was the root mean square error: A measure of the differences between values in the training set and the test set. As we run the models we observed that all the models have an RMSE of around 80. The unit of the RMSE is the same as apartment rent, which is monthly rent per square meter. Taken into account that the average monthly square meter rent is 197, our models are far from well specified when they on average miss the test set values by 80 DKK, which about 40 pct. 197 DKK.

Moreover we observed that the mechanism that splits observations into training and test sets: the `set seed` parameter had an impact on the RMSE scores. That could indicate a low sample size compared to the variance in our data. Once again we were reminded that more data usually beats better algorithms, especially algorithms that we do not fully understand.

6 Appendix

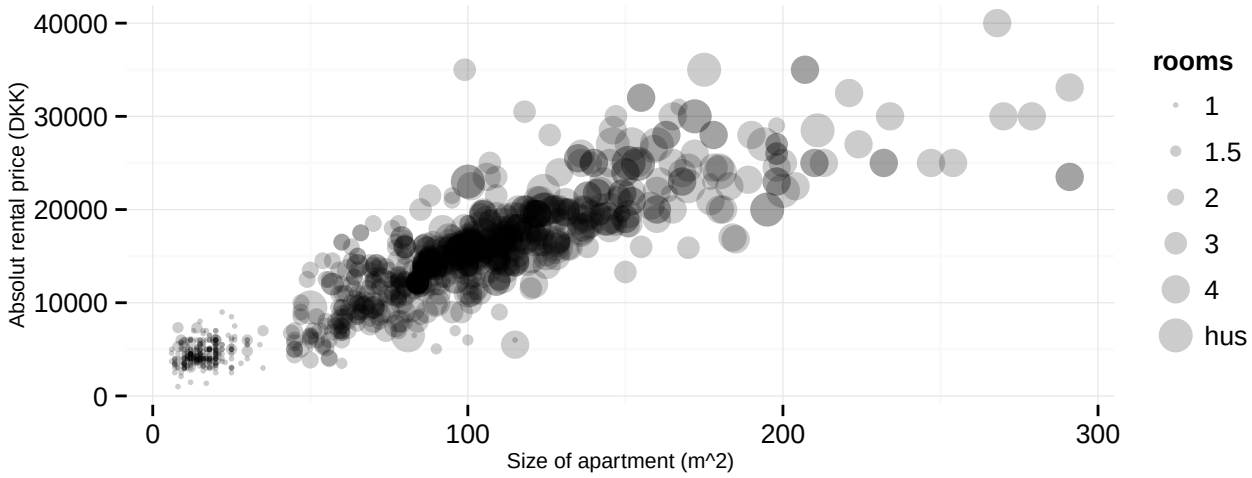

Figure 7: Rental price against the size of the apartment

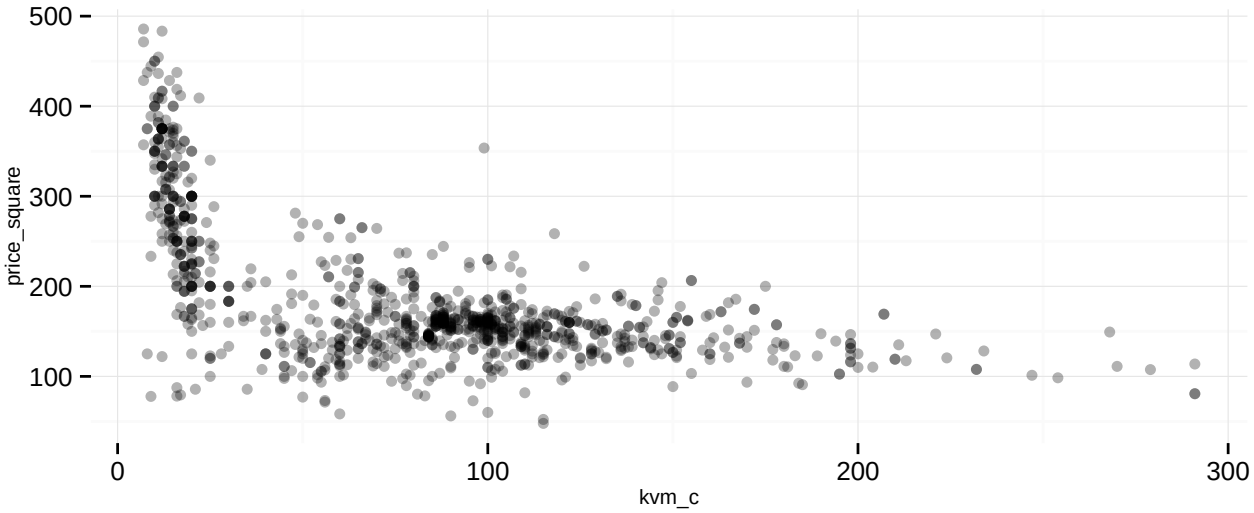

Figure 8: Price and size, without expensive apartments

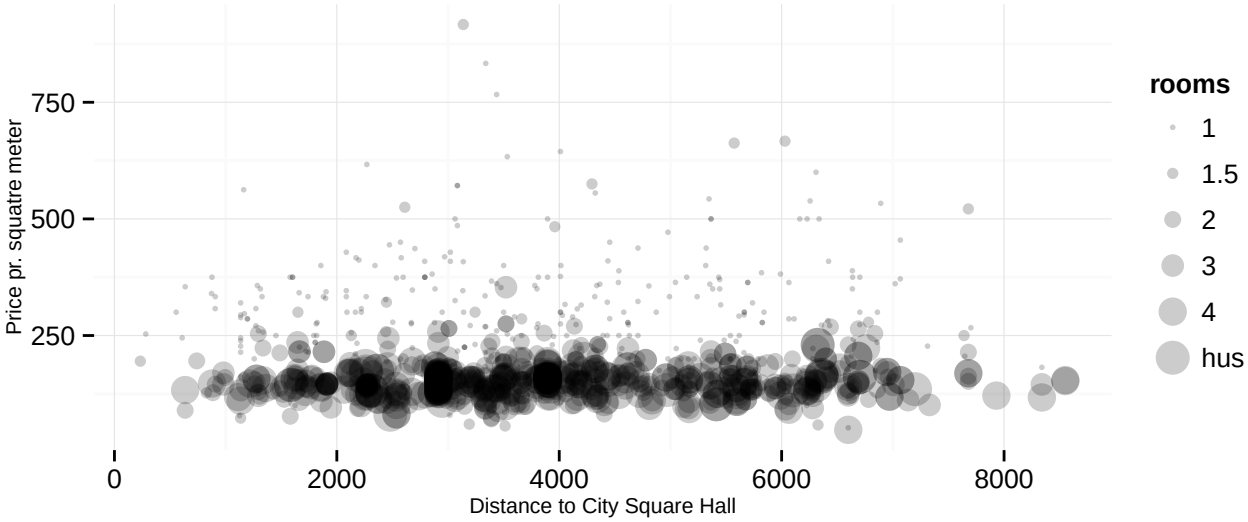

Figure 9: Distance to city square hall

Figure 10: Price differences distributed geographicaly

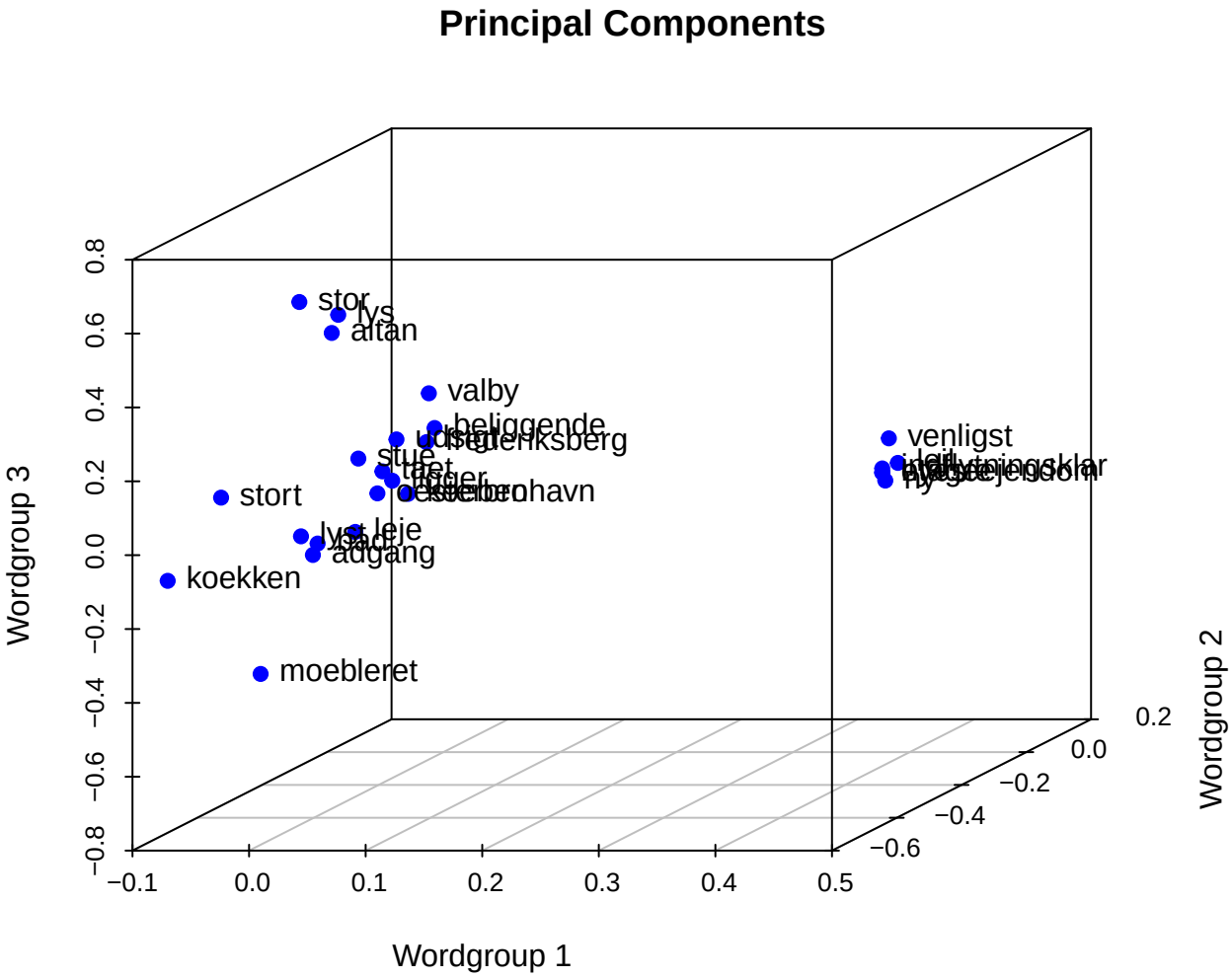

Figure 11: Price differences distributed geographicaly

Bibliography

Friedman, Milton. 1953. *Essays in Positive Economics*. Repr. Chicago: Univ. of Chicago Pr.

Scott, Fortmann-Roe. n.d. "Understanding the Bias-Variance Tradeoff." http://scott.fortmann-roe.com/docs/BiasVariance.html.