University of Zurich

Big Data Methods for Economists

Final Project

Predicting Housing Prices in Switzerland

Jingyan Yang, Marc Richter

May 5, 2020

1 Data Pre-Processing

Before we get down to build models, we want to see what the information this data set contains and we perform a summary analysis for our data set.

1.1 Overview our Data

In the training data set we have 44 variables including our outcome variable, price, but 22 variables have a over 70% missing value percentage. This means we cannot just leave out all rows with missing data, otherwise we will lose most of our information. We therefore have to treat our data set with care and find out how we can reshape our data to preserve information but not introduce bias by altering the distribution.

1.2 Manual Data Cleaning

1.2.1 Dummy Variables

We find that there are many dummy variables, like balcony, basement, furnished and so on, indicating if the apartment has this characteristic or not. However, most of the dummy variables only take the value 1 or contain missing information. This way, we can not know which of the missing values are actually 0's and which are truly missing. Our approach: We replace missing values with 0. This does create a new meaning to the dummy variables: A 1 indicates that this characteristic is mentioned for the observation, a 0 the opposite. For other numeric predictors, we decide to use mice package to impute the data set. Before doing so, we want to clean our data set furthermore and get a more precise imputation result in consequence.

1.2.2 Deletion of non useful predictors.

(1) id: just like row names.
(2) address, street, municipality: we have the variable zipcode in the dataset, which we use as a location indicator
(3) area_useable: too many missing values, and part of this is already implied by the variable area.
(4) date: date of publishing of the advertisement should not affect the housing price.
(5) date_available: too sparse information.
(6) sale: only contains 1's.

1.2.3 Location Indicator

We think using the original zipcode may lead to over-fitting our model with too many clusters having too few observations - besides having a high computational

cost. We decide that the first two digits of zip code should be a good indicator of location and create dummy variables for each "double digit zip code".

1.2.4 Outlier detection

We detect if there are some significant outliers for specific numeric predictors and delete 4 observations of year_built, which have values of year_built before 1300. We also are suspicious about observations taking the value 1 for area, however we decide to keep these as we have observations taking this value in the test data as well.

1.3 Imputation

We use two imputation methods from the mice R package.

For a first imputation. we use the cart method from mice, which uses classification and regression trees to impute the missing variables. As we only have missing values that are continuous, regression trees are used to compute the imputed variables. We use this as our baseline data set.

As a second reference model we use mean imputation, which is a very simple and straight forward approach to imputing. The upside is the clear way of imputation - we are not going to create outliers and the mean of the variables does not get changed. On the other hand, the imputed values might be very different to the values that are the actual (unobserved) values. Also, even though the mean stays the same, the variation is going to get lower, which is not what we might want to get a robust model.

We create one imputed data set each (m = 1), as multiple imputed data sets can only be used as a joint for model selection/prediction when we can assume a linear relationship. Furthermore we use maxit = 20 to make sure the predictors for the regression trees converge.

After finishing the imputation, we compare our baseline and our reference data with the original train data (with the missing values) using density plots. The density functions show that the data is differently distributed between the data sets. The mean imputation leads to spikes at the mean value, and the cart imputation leads to more evenly distributed data. Moreover, we find that the distribution of the cart imputation is much more similar to that of the original data set, so we assume the cart imputation should be more suited for our prediction. We also check the correlation matrices of these three data sets, which return us very similar results. This gives us confidence that it is safe to use the imputed data sets to make predictions.

2 Modeling

Building a model that fits the training data well but also can be used to predict the labels of the test set is our goal. We use the root mean squared error (RMSE) as our model accuracy indicator.

2.1 Preparation

We now have a first look at which variables might influence our outcome variable and in which way. We take a look at graphs showing the bivariate relationship between different independent variables and our outcome variable, price. From the scatter plots, the relationship between area and price seems to be linear, but it is very hard to tell the relationship between other predictors and price.

To estimate our models, we split our training data into a training and a validation partition, in a 80/20 relationship, to not considerably overestimate the test error.

2.2 Linear Regression

We build a first basic linear model, regressing the response on all other predictors, which gives us an RMSE of 413295.7 (for the cart imputed data set). We test for influential observations via cooks distance to see if these influential observations have any impact on our model, but it seems outliers do affect our model only arbitrarily as removing them does not considerably alter our RMSE. Plotting the fitted values vs. the residuals shows that there is heteroskedasticity present and our linear model assumptions are very likely to be hurt.

In a second model we include interaction terms of area and rooms, floors and elevator, area and floors, area and zipcode dummies, and this lowers the RMSE to 393761.4. In a third model continue to try on polynomial regressions to allow our model become more flexible. We determine the optimal degree of polynomial by performing a 5-fold cross-validation. And we also set the polynomial to have a maximum degree of 5 to avoid over-fitting.The RMSE lowers again, to now 379504.8.

This new fit including interaction terms and polynomials contains a lot of predictors, which might introduce some serious over-fitting problem in the model. It likely contains many predictors that do not explain the data well. Because of this, we perform automated step-wise model selection. The forward selection, backward selection and hybrid-way selection give us different results. Therefore, we choose the backward selection with lowest AIC and lowest RMSE of 379297.9 as our best step-wise selection method.

2.3 Ridge, LASSO and Elastic-Net Regression

For now our data has quite some variables. Especially as we redefined NA's to 0 in the case of dummy variables and imputed the other ones, this can introduce bias. Also, we do not want our model to get into the trap of what is known as "the curse of dimensionality". Therefore we choose to use Ridge and Lasso regressions by introducing the tuning parameter λ to regularize our models.

We use a loop to find the optimal α (When $\alpha = 0$, it is ridge regression, and when $\alpha = 1$, it is LASSO regression) and use cross validation to find an optimal λ. We can see that LASSO performs best among Ridge, LASSO and Elastic-Net regressions since it is designed to force coefficients to get exactly zero. This way, we are able to drop some variables which do not have good explanation power.

We also implement LASSO with interaction terms as these have worked really good in the basic linear model. The RMSE of 394108.4, however, is still higher than the linear model described above.

2.4 Moving to Non-Linearity: Smoothing Spline and GAM

As mentioned before, likely our linear model assumptions do not hold for the true relationship. Despite already having tried out including polynomials in our regression, we want to use more sophisticated approaches, such as splines and generalized additive models. The reason is that smoothing splines are more flexible than polynomials and generalized additive models allow us to apply the non-linearity methods to multiple variables.

We choose λ by using LOOCV(Leave-one-out-cross-validation), which results in 3.39 degrees of freedom for room, 9.37 degrees of freedom for floors, and 92.78 degrees of freedom for area. We include all predictors in the GAM at first, which results in a RMSE of 402343.4, then we perform a GAM with smoothing splines and the predictors from backward selection (including interaction terms) which gives a lower RMSE of 381470.8.

2.5 Walking into the Forest for some Qualitree Times

We can see that the regression tree is not competitive with other supervised learning approaches shown before. The plot tree with node labels suggests that the regression tree only considers two variables, area and zipcode, and has only 7 terminal nodes. Consequently, its prediction accuracy is pretty low with an RMSE of 478696.5.

	RMSE values	
Model	Cart Imputed Data	Mean Imputed Data
Linear Model	413295.7	432917.4
Linear Model without Outliers	413327.6	432974
Linear Model with Interaction Terms	393761.4	411990.7
Linear Model with Interaction and Polynomial Terms	379504.8	402627.6
Backward selection of Linear Model with Interaction and Polynomial Terms	379297.9	402487.2
Lasso	413341.8	433000.2
Lasso with Interaction Terms	394108.4	412291
GAM	402343.4	420829.8
GAM with Smoothing Splines	381470.8	402666.9
Lasso with Smoothing Splines	394148.2	412374.9
Regression Tree	478696.5	495772.5
Random Forest	395692.7	413603.9

Table 1: RMSE Values of different Models and Data Sets

In order to produce a more powerful prediction model, we also use a random forest approach. This lowers the RMSE considerably, but with 395692.7 it is still higher than other models.

3 Comparison with Reference Data

So far all our models have been trained and validated by the data set imputed via the cart method. We used this as our go to data as we expected it to outperform the data imputed by the simple mean method.

We trained all our data on this reference data set with mean imputation as well, and found our expectation fulfilled. All models did have a higher RMSE compared to the ones trained by the cart data.

3.1 Model Selection and Prediction

Table 1 gives a full overview of our models and their accompanying RMSE values. At the end we select the model with the lowest RMSE as our model which we want to use to make our predictions. This is the linear model containing interaction terms and polynomials trimmed by the step-wise backward selection method.

We do not use this very model though - because we trained it on only 80% of our data (to validate it with the remaining 20%. Now having chosen it, we can train the model again, but this time on our entire training data. This gives

us different parameters for our prediction.

After training our final model, we import the test data and do the very same data cleaning procedures we did on the train data - redefining the dummy variables, deleting obsolete variables and imputing the rest of the data via the cart method. We can then make our final predictions for the test data on swiss housing prices.

4 Conclusion

Doing this project we encountered numerous difficulties. Being given such a sparse data set we had to deal with a - very often encountered situation in business as well as research - situation of focusing very much on the data cleaning process. While first having thought of extracting data from data containing so much missing information as a nearly impossible task, we found out that different coding and imputation leads to smaller differences than we expected.

Our model selection was surprising as well - in the end our winning model is a linear model chosen by step-wise selection. It turns out, that - in our case at least - building a linear model including polynomials and interaction terms and further trimmed by backward selection can outperform sophisticated models like GAM or tree methods.