Text mining project

Holler Zsuzsa

June 27, 2016

Problem statement

The question I aim to answer in this study is that what features of real estate determine their prices. I use a dataset of real estate advertisements scraped from a Hungarian real estate advertisement website. From the advertisements I extract itemized informations like the size, number of rooms and type of the property. I expect these information to explain a high proportion of the price variance. The online ads also contain a few sentence long description of the property. The question I pose is whether different features extracted from these descriptions contain extra information on the property, whether these features help to explain the price differences. The idea is that some additional details, like the characteristics of the location of the property or the details of the terms of payment are not described in the itemized part of the advertisements but they might be included in the description.

Modeling strategy

In order to see if information extracted from the description part of the ads helps to explain price differences I run a linear regression on the price using only features from the itemized part of the advertisement. Then, I extract features from the text using k-means clustering on the tf-idf matrix and LDA analysis. Finally, I include these features in the regression and check whether the fit of the regression model improved.

Data collection

The dataset used for the analysis is extracted from a popular online real estate advertisement website in Hungary. The scraper I created allows for obtaining all the information available on the website for any specific property or a set of properties specified by a given search condition (e.g.: all the results for a given city or house type). The scraper automatically searches for the given search term, then loops through all the search results and saves the webpage containing the information on the properties in html format.

For the analysis I chose to download the advertisements of all flats and houses located in a relatively big Hungarian city called Szeged. The scraping was done on the 4th of Jun 2016 and the total number of scarped advertisements is 3871; 1682 houses and 2189 flats.

Note that all the advertisements and analysis are in Hungarian. Regarding the results included here I tried to translate all the tokens to English in the most consistent way possible.

Data cleaning and formatting

There are two types of information on each property page. First, the most important features of the property like the price, location, number of rooms etc. are given in a tabular format. Second, the page contains a few sentence long description of the property created by the advertiser. This descriptions might describe the property in more detail, the neighborhood, the conditions of the sale or anything the advertiser wants to share.

As the first step of processing the raw htmls all the information from the itemized part of the advertisement is extracted and organized. The entries are cleaned from errors then arranged in a standard data table format. The columns of this data table form one set of variables included in the regression analysis. These variables are the following: number of rooms and half rooms, size in square meters, ceiling height, floor, balcony (yes/no), separate toilet (yes/no), type of heating, condition of the property, garden (yes/no), faces street/garden, comfort level, air-conditioning (yes/no), elevator (yes/no), lot size, parking options, cellar (yes/no), orientation, attic (yes/no), type (brick, panel etc.). As it can be seen, most of these variables are categorical and they provide a rich set of information on the property.

As the second step, the textual description of the property is obtained and organized in a format which is suitable for the later analysis.

Removing duplicates

Unfortunately, some real estates are uploaded to the website several times so additional cleaning is required to identify and remove these duplicates. Getting rid of these observations was done in two steps. First, advertisements with identical description were identified and duplicates were discarded. In this step I dropped 125 data points.

In the second step I tried to identify those advertisements which are not completely the same but which refer to the same property. To do this, I used the document term matrix (see the details of the construction of the document term matrix in the next chapter). After checking the distribution of cosine similarities of the rows of the document term matrix I decided to assume that advertisements with higher than 0.7 cosine similarity and identical location, size and type refer to the same property. Then based on this assumption I identified groups of advertisements and discarded all but one from each group.

After removing the duplicates I ended up with 3537 advertisements that I used in the final analysis.

Preparatory steps of text mining

The unit of the text mining analysis I performed is one word, so first the descriptions of real estates are split at space like characters into words and other units in the text e.g.: numbers and punctuation. Numbers and punctuations were removed using regular expressions as well as url's which were present in some of the descriptions.

In the second step, the remaining tokens were stemmed using a freely available Hungarian natural language processing toolkit[1]. Since this toolkit is available in Java only, the set of tokens were first written into simple text files, then the analysis was performed and the result was read into Python. The output of this natural language processing toolkit has the following form; each token is listed in a table in its original form along with the stemmed version of it and the corresponding part of speech.

The final list of stemmed tokens for each advertisement were created using this table. In order to get rid of words which are not interesting from the perspective of the analysis, only adjectives, adverbs, nouns and verbs were kept. From the remaining list of tokens stopwords were removed using the Hungarian stopwords of the stop-words python library.

Data description

From the final set of advertisements and token list a 3537 by 11002 document term matrix were created containing the frequency counts of each token in each advertisement.

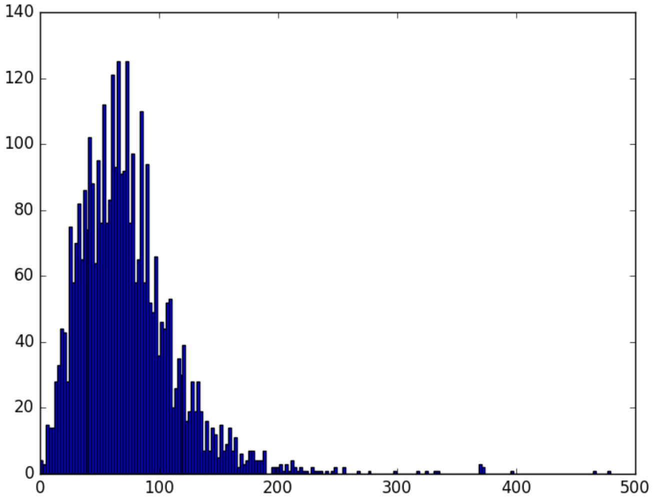
The distribution of the number of occurrences of different tokens is extremely skewed with a minimum 1, maximum 6417 and average 24. The ten most frequent words - translated from Hungarian - are shown in Table1.

The distribution of the number of tokens by advertisement is show in Figure1. As it can be seen the number of tokens by description is relatively balanced having a maximum around 500.

Table 1: Most frequent tokens

Token	**Frequency**
room	3537
real estate	2808
house	4258
separate	1970
bathroom	2161
kitchen	2281
living room	2287
flat	6417
sale	2764
located	2888

Figure 1: Cleaned token length



Clustering

The goal of the clustering analysis is to identify relevant groups - clusters - of advertisements based on the content of the descriptions. The cluster assignments then can be included in the regression analysis to see whether the different content groups are associated with different price levels.

In this part of the analysis I experimented with different representations of the advertisement descriptions. First, I computed the tf-idf representation of the descriptions and applied the clustering algorithm on the tf-idf representation of the text. Then, to reduce dimensionality, I computed a lower dimensional representation of the tf-idf matrix using Singular Value Decomposition.

For both representations, I apply k-means clustering with k going from 2 to 8. Then, I compute the sum of intra-cluster variation and based on the relative value of these I decide on the final number of clusters used.

Figure2 shows the intra-cluster sum of squares for the different k values using the tf-idf matrix. It can be seen that the optimal number of clusters is not evident in this case, but there is a relatively big drop at 7, so I chose the final number of clusters to be 7.

Table2 shows the topics corresponding to the different clusters. To determine the topics I took the cluster centroids and for each cluster centroid I identified the tokens with the highest tf-idf number. In the table the words listed are examples from the top 10 tokens according to the tf-idf ranking in the cluster-centroid, while

Figure 2: Tf-Idf Matrix, Intra-Cluster Variation

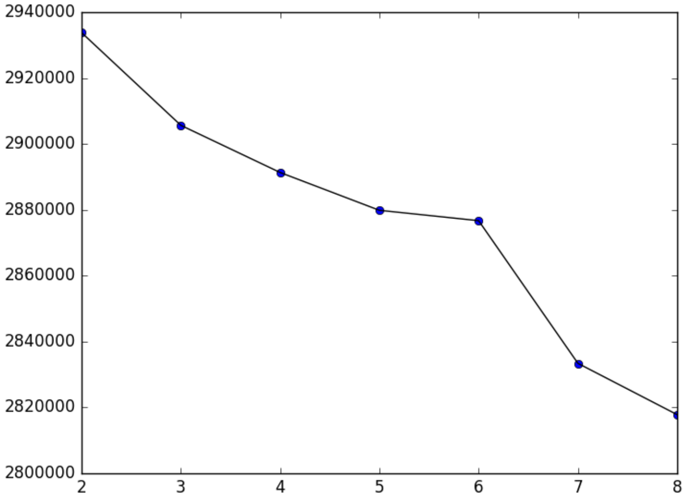


Table 2: Tf-Idf Matrix, Clusters

Cluster	Content
1	Description of property. E.g.: wc, room, terrace, living room, located
2	Terms related to mortgage loans. E.g.: proportional, discount, monthly, year
3	Technical terms. E.g.: concrete, plaster, bridging, mortar
4	Technical terms related to utilities. E.g.: wire, water meter, conduit
5	Technical terms. E.g.: concrete, plaster, bridging, mortar
6	Description of property. E.g.: bedroom, door, hall
7	General description terms of the location and the property. E.g.: renovated, apartment house, building

the topic descriptions are created based on intuition.

In order to get rid of the irrelevant dimensions in the tf-idf matrix I applied Singular Value Decomposition. Unfortunately, the singular values of the matrix did not suggest any clear cut-off value so I decided to choose the dimensionality of the reduced matrix to be 100 rather arbitrarily.

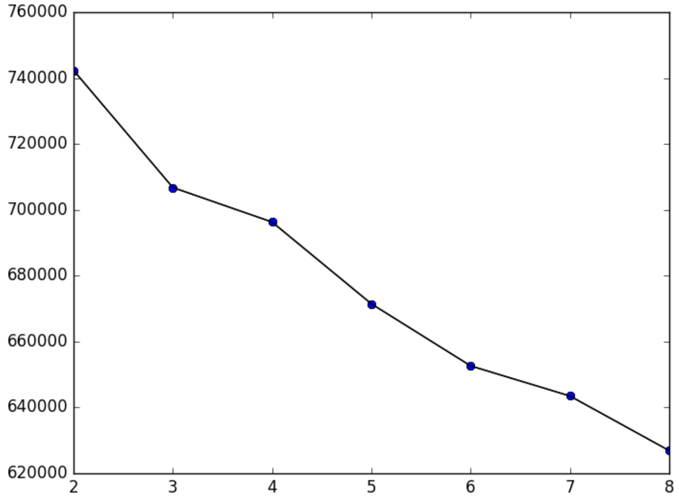
Figure3 shows the intra-cluster sum of squares for the different k values using the dimension-reduced tf-idf matrix. It can be seen that the optimal number of clusters is not evident in this case either, but it seems that there is a smaller elbow at 3 and 6. Since the number of observations is high I decided to keep the number of clusters higher as well so I run the final clustering with k = 6.

Table 3: Reduced Tf-Idf Matrix, Clusters

Cluster	Content
1	General description terms of the location and the property. E.g.: renovated, apartment house, center
2	Terms related to mortgage loans. E.g.: proportional, discount, monthly, year
3	Description of property. E.g.: wc, room, terrace, living room, located
4	? E.g.: loggia, sold, appearance
5	Technical terms related to renovation. E.g.: work, ventilation, installation
6	Technical terms. E.g.: concrete, plaster, bridging, mortar

Table3 shows that the topics emerged are relatively similar to the previous case. It seems that there are three clearly different topics which appear with different weight in the different advertisements and create this groupings. First, there are the general description terms of location and property features. Then terms

Figure 3: Tf-Idf Matrix, Intra-Cluster Variation



related to renovation form a different topic. Finally, there are the clusters corresponding to mortgage loans and real estate agency activity.

LDA analysis

LDA, similarly to the clustering analysis, aims to identify different topics appearing in the text without explicitly specifying these topics and the corresponding words. On the other hand, while the clustering method links one description to one topic/cluster, the LDA might be capable of identifying more complex structures where advertisements are linked to possibly more than one topic. To perform the LDA analysis I used the gensim library of python.

After experimenting with the regression using different number of topics I chose to set the number of topics to 10 in the final specification. The reason for this is that while the regression results seemed to improve with the number of topics, I wanted to keep the number relatively low so I can interpret the outcome more easily.

Table4 shows the tokens and probabilities associated with the 10 most likely tokens for each topic. It can be seen that while there are similar topics as we have seen in the cluster analysis, a few new, interesting ones also emerged. Topic 1 and 5 for example seem to be associated with high quality, luxury things. Also, properties relevant for flats and family houses are more or less separated into different topics.

The features extracted from the LDA analysis to include in the regression are the probabilities that a word of a given topic appears in the description of a given advertisment.

Regression

In this part of the analysis, I tried to explore whether the features extracted from the descriptions help to improve the fit of a regression model estimating the price of the properties. I fit simple linear regressions on the price variable using different set of variables. In each case, I train the regression model using approximately 85% of the observations and I use the remaining 15% for testing.

Table5 shows the R-squared values computed on the test set for the different specifications. The first row contains the fit of the regression model using only variables extracted from the itemized part of the webpage. As it can be seen, the test R-squared is very high. The second row shows the result after including the cluster

Table 4: LDA topics

Topic	Token probabilities
1	0.005*built-in + 0.005*modern + 0.005*electric + 0.005*water + 0.004*x + 0.004*full + 0.004*area + 0.004*pool + 0.003*wellness + 0.003*equipment
2	0.031*year + 0.020*free + 0.019*call + 0.018*be + 0.016*even + 0.016*described + 0.016*number + 0.016*contact + 0.016*no + 0.015*only
3	0.045*flat + 0.017*room + 0.016*sale + 0.015*new + 0.014*renovated + 0.013*room + 0.010*full + 0.010*upstairs + 0.009*located + 0.009*doors and windows
4	0.037*flat + 0.019*room + 0.016*living room + 0.014*separate + 0.013*located + 0.012*kitchen + 0.012*real estate + 0.011*bathroom + 0.011*premise + 0.010*be
5	0.012*elegance + 0.010*high + 0.006*premium + 0.006*ticket + 0.005*well + 0.005*satisfy + 0.005*category + 0.005*none + 0.005*category + 0.005*lighting
6	0.021*house + 0.014*real estate + 0.012*located + 0.011*room + 0.010*family + 0.010*family + 0.009*lot + 0.009*know + 0.008*kitchen + 0.007*building + 0.007*outbuilding
7	0.045*flat + 0.026*apartment house + 0.018*sale + 0.017*offer 0.016*call + 0.016*szeged + 0.014*number + 0.014*interest + 0.014*get + 0.011*excellent
8	0.054*flat + 0.025*construction-ready + 0.014*price + 0.014*garage + 0.012*terrace + 0.012*condition + 0.009*sale + 0.009*db + 0.009*deliverance + 0.009*apartment house
9	0.037*house + 0.021*living room + 0.014*family + 0.013*bathroom + 0.013*located + 0.013*terrace + 0.013*garage + 0.012*kitchen + 0.011*be + 0.010*floor
10	0.035*house + 0.017*room + 0.015*be + 0.015*lot + 0.014*located + 0.013*sale + 0.012*real estate + 0.011*family + 0.009*room + 0.009*heating

labels created based on the tf-idf matrix. The test R-squared is slightly higher but the difference seems negligible. I do not include here the results with the variables based on the reduced tf-idf matrix clustering because those are very similar to the previous one. The third row contains the result of the model after adding the probabilities coming from the LDA model. As it can be seen, the R-squared is again a bit higher compared to both of the other two models.

Table 5: Regression results

Variables	R-squared
Base	0.68
Cluster	0.69
LDA	0.71

Conclusion

In general, it seems that the text mining techniques used in the analysis were able to identify relevant topics in the descriptions. Even with the relatively simple clustering analysis intuitively interpretable topics were identified. Unfortunately, the features extracted from the clustering analysis did not improve the regression model fit. The LDA analysis performed somewhat better in both aspects. Some interesting new topics arose compared to the clustering and the regression fit somewhat improved compared to the model using only the itemized properties extracted from the ads. The fact that the test R-squared could not be improved significantly might be explained by the rich set of information available in the itemized part of the advertisements. Also, some of the topics extracted from the text clearly overlap with the itemized information.

References

[1] Zsibrita, János; Vincze, Veronika; Farkas, Richárd 2013: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: Proceedings of RANLP 2013, pp. 763-771.