ANALYSIS OF HOUSE SALE PRICES

for King County, Seattle

author: **Janina Smoła**

June 2018

Task description

This task is meant to verify the skills of applicants to Data Science team with regards to creating Machine Learning models. It is to perform data analysis and create a regression model that can predict the property price based on the other features.

Data set

Data set is given in the file „house.csv", that contains data describing real estate prices and certain features of houses located in area of Seattle, state of Washington (acc. variable 'long', 'lat' and 'zipcode').

Chosen variables

The variables chosen to the modelling phase according to the client instruction:

- '*id*' – database identifier of the current row,
- '*date'* – pricing date,
- '*price*' – the price
- '*bedrooms*' – number of bedrooms,
- '*bathrooms*' – number of bathrooms,
- '*sqft_living*' – living space area,
- '*sqft_lot*' – lot area,
- '*floors*' – number of floors,
- '*waterfront*' – indicator of whether the property is facing water {0,1},
- '*view*' – quality of view from the property (0:4),
- '*condition*' – property condition (1:5),
- '*grade*' – property grade (1:13),
- '*sqft_above*' – living area above ground level,
- '*sqft_basement'* – area of basement,
- '*yr_built*' – year the building was built

Source data has 21613 items with 21 variables (in task will be used only 15 mentioned above) : 'id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_living15' and 'sqft_lot15'. In the data set there are no missing values.

Used packages

The analysis and processing of data as well as the built models were carried out in the R version 3.4.4 "Someone to Lean On" program using system library and the following packages

caTools 1.17.1
dplyr 0.7.4
corrplot 0.84
ggplot2 2.2.1
DataExplorer 0.6.0
e1071 1.6-8
randomForest 4.6-14
Metrics 0.1.3
tseries 0.10-45

Plan of steps to taken

1. Data pre-processing – checking their format, cleaning, transformation and understanding the dependence to dependant variable
2. Choosing the variables to the final dataset that will be used to build the model (with help of Multiple Linear Regression)
3. Models construction: Multiple Linear Regression, Support Vector Regression and Random Forest
4. Models' comparison and final conclusions.

Data pre-processing – first remarks

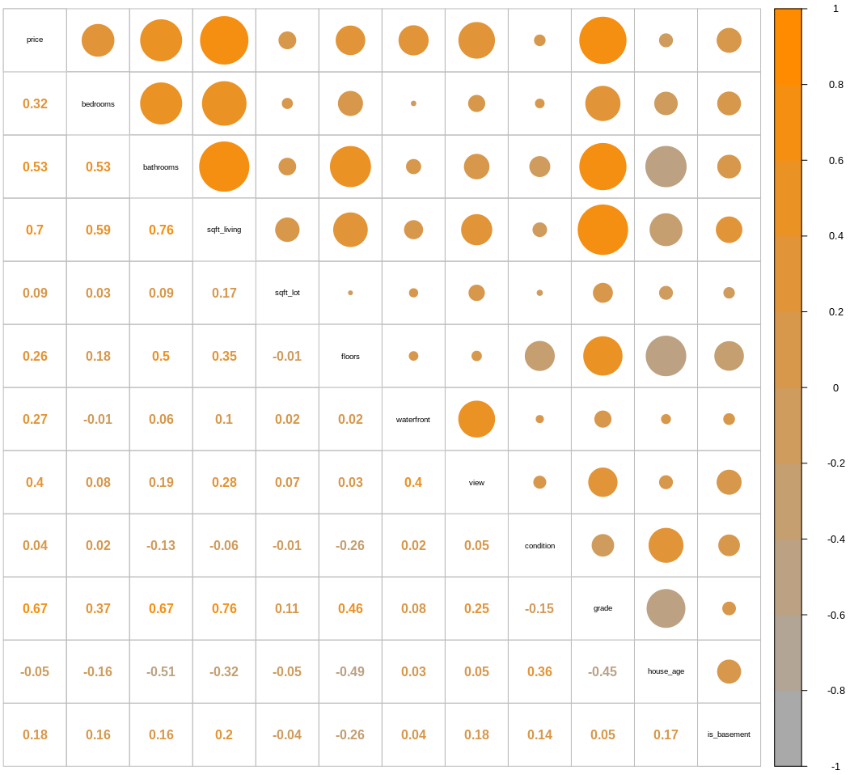
Based on the first look on the data set, the following observations, questions and conclusions arise:

- variables 'waterfront', 'view', 'condition' and 'grade' are the categorical type variable (Factor), now coded as integer (it can stay like that),
- variable 'date' has to be change to date type (now is coded as Factor),
- variable 'id' has to be in normal not scientific format,
- to consideration: variable 'yr_built' has to be change to date type,
- question: why variable 'bathrooms' - number of bathrooms - has not integer value?
- question: why variable 'floors' - number of floor - has not integer value?
- question: what describes variable 'condition'? is that the expert opinion (subjective) or same evaluation made on an objective scale? (this need to be clarify with the data provider)
- question: what has coded variable 'grade'? (this need to be clarify with the data provider)
- quite good correlation – dependant and independent variable: ‘price’ ~ ‘sqft_living’, ‘price’ ~ ‘grade’, ‘price’ ~ ‘sqft_above’,
- very strong correlation – independent variables: ‘sqft_above’ ~ ‘sqft_living’,
- quite good correlation – independent variables: ‘sqft_above’ ~ ‘bathrooms’, ‘sqft_living’ ~ ‘grade’, ‘sqft_above’ ~ ‘grade’, ‘bathrooms’ ~ ‘grade’, ‘bathrooms’ ~ ‘sqft_above’.

Variable types

The data set consists of:

- dependent variable - numerical, continuous: *'price'*
- independent variable - object identifier: *'id'*
- independent variable – numerical, discrete: *'date', 'bedrooms', 'bathrooms', 'floors', 'yr_built'*
- independent variable – numerical, continuous: *'sqft_living', sqft_lot', 'sqft_above', 'sqft_basement'*
- independent variable – categorical, nominal: *'waterfront', 'view', 'grade'*
- *independent variable – categorical, probably ordinal: 'condition'*



price											
0.32	bedrooms										
0.53	0.53	bathrooms									
0.7	0.59	0.76	sqft_living								
0.09	0.03	0.09	0.17	sqft_lot							
0.26	0.18	0.5	0.35	-0.01	floors						
0.27	-0.01	0.06	0.1	0.02	0.02	waterfront					
0.4	0.08	0.19	0.28	0.07	0.03	0.4	view				
0.04	0.02	-0.13	-0.06	-0.01	-0.26	0.02	0.05	condition			
0.67	0.37	0.67	0.76	0.11	0.46	0.08	0.25	-0.15	grade		
-0.05	-0.16	-0.51	-0.32	-0.05	-0.49	0.03	0.05	0.36	-0.45	house_age	
0.18	0.16	0.16	0.2	-0.04	-0.26	0.04	0.18	0.14	0.05	0.17	is_basement

Variable type was determined based on the given variable description and the information provided by function str() and correlogram. That’s target state of type.

New variables

New variables were built during the data pre-processing as a proper format of given data or as a replacement or as minor (helping) variable.

- ‘rm’ - indicators of whether to delete the given item (0 – to stay in final dataset, 1 – to be deleted)
- ‘house_age’ - difference between ‘date_pro’ and ‘yr_built’ ', in numeric format,
- ‘is_basement ‘ - indicator of the house have basement - to consideration to use as a pair for ‘sqft_living’ (instead of pair ‘sqft_above’ and ‘sqtf_basement’),
- ‘date_pro’ - the proper format of variable 'date', in format Date,
- 'date_Ym' built only on information about year and month of pricing date, in character format,
- 'date_Y' built only on information about year of pricing date, in numeric format.

Data pre-processing - issues to be solved

During the data pre-processing shown up a lot of inaccuracies and problems with given data. Some of them resulted from lack of information (the specialist branch knowledge), some – from bad format of data itself and some was determined by relation inside dataset.

1. Variable **‘date’**, that describe pricing date, has wrong format – it is coded as Factor (categorical) not as Date; to solve this problem new variables were built: ‘date_pro’ (the full date information) and 'date_Ym' (date as year and month). This second one was meant as this more useful, because of format of variable ‘yr_built’ and its own generalized character.
2. Variable **‘id’** - database identifier of the current item – has scientific format that is hard to work with and which was changed into user-friendly character (string) format. It doesn't change the potential of dataset, because this variable will be not used in the model and it will not be used in any calculations. Also was found that in base are 177 duplicated items, that means that in dataset 87 items show up twice and one – triple. To decide how to deal with duplicates the analysis dependence of ‘id’, ‘data_pro’ and ‘price’ was provided. The conclusion is that all these duplicated items has different value of ‘data_pro’ means different pricing time and – with exception of items with three ‘id’ - different value of ‘price’. It was decided to remove from dataset only second appearance of the same id of items with the same 'price' and different 'data_pro'. For that purpose was built new variable ‘rm’ - indicator whether to remove item.
3. Variable **‘bedrooms’** gives information about number of bedrooms in house. According to the American standards for houses appraisal bedroom must be of adequate size (100 square feet or more), have a closet, a window and a door. It must be heated/cooled and finished in the same quality as the rest of the house. It must also be above grade and have reasonable access to a full bathroom. Basement bedrooms (50% below adjacent grade level) may not be counted in the total bedrooms count. The contingency table shows that there is one outlier with value equal 33. After analysis of similar items it was considered a mistake and changed into value 3 (the most frequent for similar items). Another remarks regarding number of bedrooms - some items has 0 as a value of bedrooms. It is possible because basement bedrooms may not be counted in the total bedrooms count (however none of these items have basement). It's also possible that some of them do not meet with other requirements (area, heating/cooling, etc.). This remark needs to be remembered.
4. Variable **‘bathrooms’** giving the information about number of bathrooms – is not integer. According to the American standards for houses appraisal a full bathroom is made up of four parts: a sink, a shower, a bathtub, and a toilet and counts as 1 bathroom. Each utility is counted as one-quarter, so for example a sink and a toilet are 0,5 bathroom. So the data - even if the number of bathroom isn't integer - should be considered as correct. It’ also

important that Basement bathroom may not be counted in the total bathroom count. The contingency table of bathrooms and bedrooms shows that some items has 0 bedrooms and 0 bathrooms. It is possible to imagine, that exist houses which bedrooms doesn't meet all bedroom requirements (no cooling) so the number of bedroom will be equal 0, but number of bathroom should not be equal 0 (it could be equal 0 only when the house has basement but none of these items have basement) so the item with no bathroom will be removed. The case when number of bedrooms and number of bathrooms, both are equal 0 and there is no basement is strange. It is considered as a mistake and these items will be removed.

5. Variable **'floors'** giving the information about number of bathrooms – is not integer. It should be consider as a proper value because according to the American standards for houses appraisal attics and mezzanine should be count as half floor. It's important to remember that basement should not be counted in the total floor count.
6. Variable **'waterfront'** indicate the property is facing water. Interesting remarks that almost no house (only 0,7% of all) has a view of the water, what is quite interesting in the Seattle (port city).
7. Variable **'yr_built'** gives information about year the building was built. Because the pricing date and the building date are given with different accuracy (day-month-year versus year) and because the pricing date covers one year but at the turn of the years 2014 and 2015 new variable **'house_age'** was built. It solves a problem with difference of house age that is bind with different pricing date (house built in 1950 priced in 2014 is 64 years old and in 2015 - 65). Now the price will be a function of one variable 'house_age' and not of two variables 'date' (pricing) and 'yr_built'. That has also its consequences: after deleting 'date' (pricing) and 'yr_built' there will be no longer possible to analysis the price change according the month of pricing or difference during the year.
8. Variable **'house_age'** was built as a difference between 'date_pro' and 'yr_built' and is given in years. The contingency table of 'house_age' shows that some items has value equal 0 (430 items) and less than 0 (12 items). The value 0 can be considered as a proper because the pricing can be done in the same year as house was built. However there are 12 items with the house age with value equal -1. It's possible if investor asks about pricing based on design - not based on really existing building. Not to put 'strange' data into model and because it's only 12 (0.06% of all) such items, the value -1 will be changed into 0 (means the houses that were priced in the same year that were also built). To check of quality of new variable the correlation between 'price', 'yr_built', 'house_age', 'date_Y' (new variable 'date_Y' giving only information about year of pricing date, in numeric format) was checked. The price is not correlated with the pricing date so it's possible to delete 'yr_built' and 'date' (with all group of variable based on it) and leave only 'house_age'.
9. Variable **'is_basement'** is an indicator of the house have basement. Because 'sqft_basement' + 'sqft_above' = 'sqft_living' and both value 'sqft_above' and 'sqft_living' are strongly correlated (0,88) it seems a better solution to delete one of these value ('sqft_basement', 'sqft_above', 'sqft_living') and not to lose any information put in the model one of pairs: 'sqft_basement' + 'sqft_above' or 'is_basement' + 'sqft_living'.

Choosing the variables to the final dataset

The dataset with candidate variable to the final data set has 21600 items and 14 variables: 'price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'house_age' and 'is_basement'. The further analysis with the use of Multiple Linear Regression (backward/forward/both stepwise model selection by AIC) shows that the final model will not have variable 'sqft_above' and 'sqft_basement' (so it is 21600 items and 12 variables). The final data was divided into train and test set in proportion 80:20.

Models construction

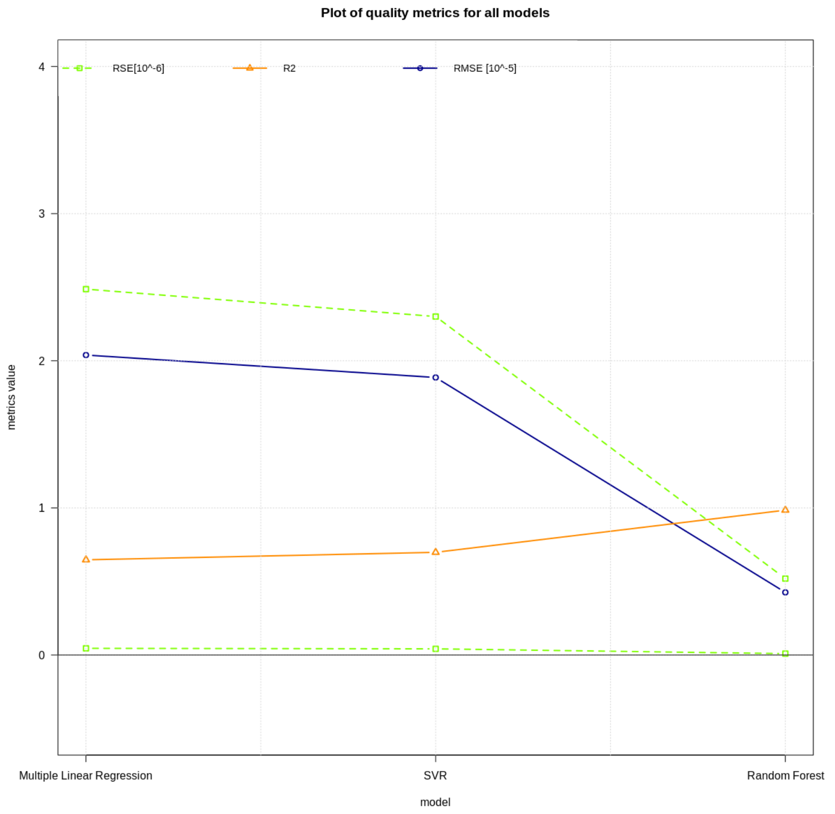
Three models was built: Multiple Linear Regression, Support Vector Regression and Random Forest with default parameters. Models were trained on train set and after that were checked with test set. The MLR model predict for the variable ‘price’ value less than 0 (it’s only 12/5400 but still - ‘price’ should be greater than 0); also normality of error does not look good. The SVR model predict proper value – greater than 0, but normality of errors still are far from perfect. The Random Forest model (with 100 trees) provide better outcome; predicted values are greater than 0 and normality of errors looks good.

Models comparison

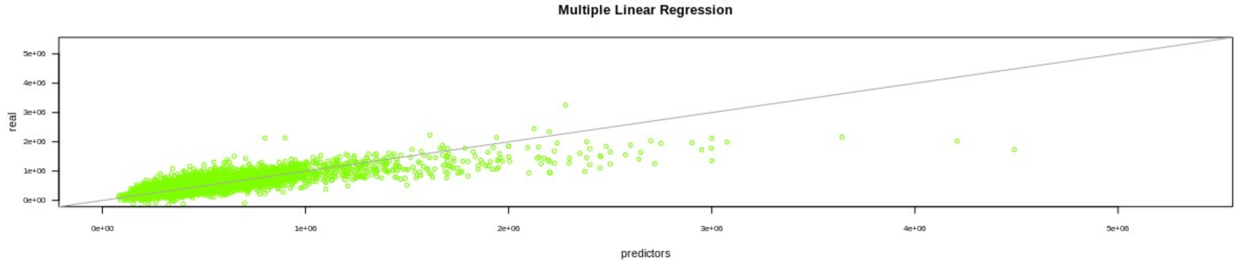
Models were evaluated on base of metric RSE (Relative Squared Error), RMSE (Root Mean Squared Error) and R2 (R-squared - coefficient of determination).

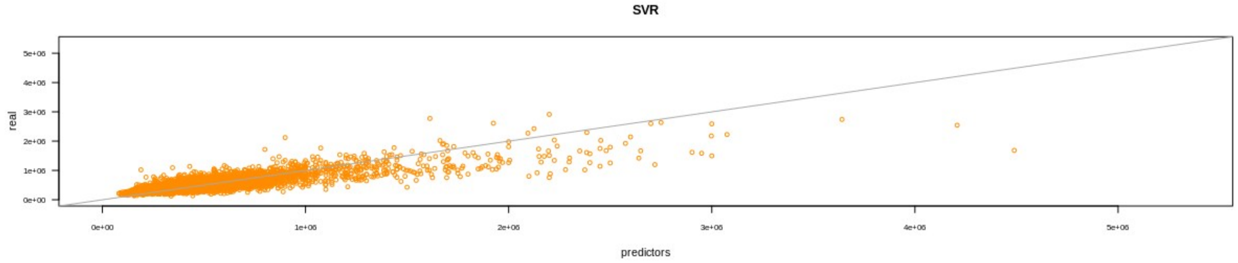
	RMSE	RSE	R2
MLR	203878,61	203916,37	0,6471001
SVR	188619,79	188654,73	0,6979473
RF	42572,17	42580,06	0,9846128

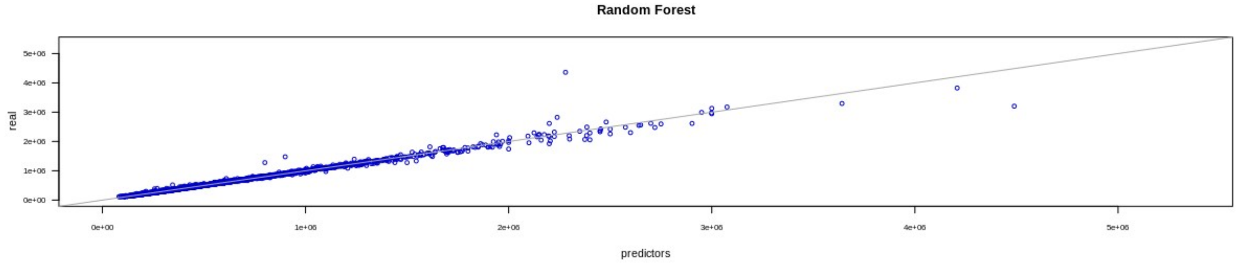
R-squared is a statistical measure of how close the data are to the fitted regression line, in another word it is the proportion of the variance in the dependent variable that is predictable from the independent variable. The greater R2 is the better the model fits given data and better explains the variation. The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data – lower values of RMSE indicate better fit. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit. The RSE is a measure of lack of fit of the model to the given data and it should be referred to value of dependent variable. The smaller difference between RSE/min(‘price’) and RSE/max(‘price’) the better is the model.



Analysing the values of quality metrics – table and plot - it can be noticed that the best statistics have the Random Forest model and the worst - Multiple Linear Regression. R2 equal 0,985 is very good result. Knowing the limitation of each of them – we receive what we should expect.
The following plot shows the match between the predicted values and the real ones - the relation predicted value to the real are concentrated on diagonal that means that the error of prediction is small.







Potential next steps

The results of the above analysis show only an approximate potential of the model, but the one that should be developed is Random Forest.
The next step should be checking if deleting variables ‘floors’ and/or ‘is_basement’ could improve the model. It would be also a good idea do dig a little if any other variable could not be changed in some new one more explain the ‘price’ value. The other interesting possibility is to divide the dataset in group according to the features (for example acc. to the ‘grade’ which probably is hidden information about house) and try to build such model – with some predefined group. Also principal components analysis could be used to find out new variable that possibly provide better prediction; the regression could be proceed on PCA variable. Also driven the analysis based on time data could give interesting view – so checking if the month of pricing influence the price (but for that the dataset could be to small – mean number of items for each month is 1662).

Attachments

(1) file *code_Price_of_houses_in_Seattle_JS.R*
contains the complete code in R, in which in part *01_DATA* the given data are loaded, in part *02_EDA* the analysis and data pre-processing is driven, in part *03_FINAL DATASET* final set of variable is chosen and divided in train and test set, in part *04_MODEL: REGRESSION, 05_MODEL: SUPPORT VECTOR REGGRESION and 06_MODEL: RANDOM FOREST* the described model are built and in part *07_COMPARISION OF MODELS* the best model is chosen