Nikita Traynin, nxt170002, 2021413175

9/19/2021

Linear Regression on the Russian Housing Market

**Introduction**

Our goal is to predict a property's sale price based on a collection of other known variables related to that property. We take a dataset from Kaggle and perform exploratory data analysis, followed by model application (linear regression in this case), and finally evaluating the model and re-tuning it.

We are using the Sberbank Russian Housing Market dataset. It is pre-split into two sets for us - train and test. These use an approximately 80/20 train/test split pattern. In total, there are 38,133 observations. The train observations are before 7/1/2015, and the test observations are after. The train/test split is not random, since it is based on timestamp, so I will avoid re-mixing the data and performing cross-validation.

There are 292 columns in this data. One is a superfluous id variable, and another is our target variable "price\_doc". So, we have a large 290 predictors. For this reason, we will implement a gradient descent, iterative algorithm for our linear regression instead of the closed-form solution.

**EDA**

There are 157 integer, 119 float, and 16 categorical variables. There are too many to fully describe here, though some examples are: "full\_sq" (integer – the total square footage),"green\_zone\_part" (float – the proportion of greenery in the nearby area), and "hospital\_beds\_raion" (float – the number of hospital beds in the district). We now analyze the normality of the numeric variables. Since there are so many, I will only show the first 10.

Chart, bar chart

Description automatically generated

Graphical user interface, chart

Description automatically generated with medium confidence

As we can see, the histograms are very skewed. With these variables, they are all skewed right. Furthermore, extreme outliers are very common, as most of the bins are not big enough to be visible in the plots.

This is largely because many quantities do not have a set upper limit, while they are non-negative. For example, the number of rooms in a property. Large mansions are rare, while an overwhelming majority of buildings have a small quantity of rooms. As such, we must standardize and normalize all the variables. Although this won't remove the asymmetry, this is as far as we should go as empirically it is good enough.

We must deal with the categorical and Boolean variables before applying linear regression. In this dataset, we will convert booleans into 0s and 1s and leave them untouched. The categoricals will be one-hot-encoded, with one column dropped afterwards to prevent multicollinearity. However, there is one categorical that has 146 categories. This is such a huge amount that we will drop this variable (the sub\_area) for performance reasons, but try to put it back later. The only other categorical is the ecology rating, and we will one-hot-encode it. Enumerating it is another viable option but we will stick to OHE.

A note must be made that some of the categorical variables may be pre-enumerated (I found such a variable "material"), even if there is no logical ordering. However, since they are also numerics, finding them would likely involve a complex program or else combing through all 290+ variables by hand to find categoricals. Thus, we will leave them be as their numeric type will not break the linear regression.

Another issue is missing data. This set has a large proportion of NaNs which will not allow linear regression to work. Therefore, we must handle them first. We will impute all NaNs with the median value of that column. Mean is usually chosen but there are some mistaken data that will skew the mean,

Note, for example, that one of the properties has its build year as "20052009". Another has "4965". Obviously, these are mistaken inputs. But these two observations alone are enough to skew the mean build year to 3068. We will fix these inputs in particular but there are likely many more mistakes in this set. Furthermore, some mistakes may not be corrected without outside research. For example, some of the properties could have their wall material inputted as wood, when in reality it was stone. We cannot fix all such mistakes, so our model further distances itself from practical useage, in addition to the linearity constraint.

**Correlation**

We will set a threshold of .1 correlation to allow a variable to be included in the model, as an initial setup. We end up with 44 remaining variables when we do this, out of almost 300 original. That is still a lot to display, so we will increase the threshold to .15. Using the R2 metric, the best threshold (the ideal quantity of variables) was about 33 variables.

We also need to investigate multicollinearity. Linear regression is known to perform poorly with multicollinear data. This is a correlation matrix between the important predictors:

Graphical user interface, application, table, Excel

Description automatically generated

There is a logical explanation for many of the extremely high correlations. For instance, it's obvious that the population of adolescents will be hugely proportional to the total population. Of course, younger and older areas will deviate, but the value is likely to be close to 1. We can see that indeed a majority of the "important" variables are age-related, so it is no surprise that they are all highly correlated. This will throw off the model, so we have to remove those variables.

To make sure we still have some variables, we will start with a modest correlation cut-off of 0.4. That is, for every possible pair of variables, if any have a correlation larger than 0.4, one will be removed. After that we shoul be left

**Initial Model**

The initial model performed very poorly. This could be due to many underlying reasons. It's very possible that the linearity assumption is incorrect. The relationship between the price of a house and the variables in this particular dataset could be highly non-linear, and we did not investigate this possibility for the sake of this report. Also, we did not tune the hyperparameters of this model yet. However, multicollinearity is almost definitely largely responsible.

Our baseline model started with a constant learning rate for simplicity. We also turned on verbose output to analyze the loss over time. Here is the loss graph, which should help us understand if our learning process is ideal or not. Our initial R^2 was highly negative. It was on the order of magnitued of -10^20. But this is before removing multicollinearity.

After removing multicollinearity,

**Tuning**