27th December, 2020

Analysis of GDP growth determinants with the regression tree - fast track project number 10

Mateusz Marzec, 316060

Mateusz Szysz, 298911

1 Introduction

The main goals of the report are as follows:

- providing information concerning method used in the analysis,
- describing R implementation details,
- presenting information on the experiment and results.

All remaining information is included in the R file which is attached.

2 Regression method description

The method which we decided to use is CART (Classification and Regression Tree) with minimizing residual sum of squares as a splitting criterion.

2.1 Overview of the algorithm

The goal is to divide the whole data set into disjunctive subsets by sequential binary splits by given independent attributes. As a result you get your original domain of acceptable attributes values divided into regions. For each of them you calculate mean of the dependent variable which will be the predicted value for independent observations.
Let's say we want to check splitting by attribute X. In each step of the algorithm the splitting procedure works as follows:

1. Firstly, the whole data set is ordered by the values of X and only the distinct of them are taken into account.

2. For each value x in the obtained vector, the dataset is divided in two and the cost function CF is computed:

$$CF(X, x) = \sum_{X:\ x_i <= x} (y_i - \hat{y_1})^2 + \sum_{X:\ x_i > x} (y_i - \hat{y_2})^2, \tag{1}$$

where y_1 and y_2 are means of the dependent variable in given subsets and y_i's are values of the predicted attribute for given observations.

3. When all possible thresholds are checked, the one with the smallest value of cost function is taken into account.

2.2 Overfitting

This procedure is repeated for all attributes. At the end the variable and the threshold which minimize the cost function are chosen and the split is made according to it. Next, for each subset the procedure is repeated. In theory this process may be continued until there is no possible split (all observations in the subset have the same value of dependent variable). However it could lead to enormous overfitting which means that the model works perfectly for the training set but is much worse on the independent one. To deal with this threat, the following improvements to the algorithm were applied:

- Minimum number of observations in given node - the algorithm stops not only when there is no possible split which may decrease residual sum of squares but even if the number of observations reaches minimal acceptable threshold.
- Max depth - the algorithm stops if the number of splits of given subset of observations reached maximal value (if the depth of the tree counted as maximal path from the root to one of the leaves reaches maximal value).
- Alpha (penalty function coefficient) - it modifies function (1) in the way that there is one more component $+\alpha * |L|$, where $|L|$ is number of leaves in the tree. It makes there is a penalty factor which prevents from splitting if the accuracy of prediction increases only a little.

As the first two of the above parameters are quite easy to chose, the last one should be probably chosen for each problem separately and experimentally.

2.3 Discussion

It's easy to spot some limitations of the proposed algorithm. For example, it might not give the best global solution but only local one. It's due to the fact that the algorithm is greedy - it makes the best split in given subtree without considering future splits. However it's questionable if in case of making the algorithm globally optimal by recursion, the performance of the model would increase largely but at the same time it's certain that the computational complexity would extremely increase. Another objection may refer to the way of finding splitting threshold for given attribute.

It may occur that the number of possible splits is enormously large so checking all of them would be inefficient. Apparently it's true but we were focused on conducting the experiment on the set of countries whose number is limited and relatively small. Secondly, choosing constant number of possible thresholds might decrease the accuracy of the predictions. To cope with these problems one may propose a compromise solution - if the number of possible values is smaller than some constant value (for example 1000) all splits are considered, otherwise the data is split by 1000 equally distributed points or 1000 distinct attribute values are chosen randomly. It could preserve trade-off between accuracy and computational efficiency.

3 Implementation

In general we tried to implement the tree on our own, no basing on external libraries. The only one that used we was *data.tree* which provides implementation of the tree structure.

3.1 Main functions

We implemented three functions for the use of users and three additional functions that are called by the former. The main functions are:

- *build_tree* - the most important function responsible for creating root of the tree and calling *add_nodes* function that recursively adds new nodes to the tree. This function also calls another function - *select_partition* that chooses the best split in given part of the tree calling *find_threshold* for each attribute. The function allows to set anti-overfitting parameters concerning minimal number of observations in each leaf, maximal depth of the tree and the cost function coefficient. What is more, it allows to provide vector of columns that should be used in tree construction instead of all possible attributes.

- *plot_tree* - function for plotting the tree generated by *build_tree*. It allows to set parameters concerning rounding of numbers on the plot, displaying numbers of leaves on each of them and displaying number of observations in given node.

- *predict* - function that predicts value of dependent variable for object containing all attributes used by generated tree.

All implementation details and functions parameters are described in the R file attached to the report.

4 Results

In this section we will describe obtained results. The source of our data is World Pen Table. We have created two distinct models and we will present results obtained for both of them.

4.1 Model 1

In this model the determinant variable is GDP per capita in 2017 in different countries. Due to some missing data the final number of observations is equal to 144. Our independent variables are:

- *pop* which is simply the population of given country,
- *xr* which is exchange rate (national currency/USD),
- *delta* which is an average depreciation rate of the capital stock
- *hc* which is Human Capital Index (based on years of schooling and returns to education).

Below we present the splits obtained with the algorithm described in section 2.

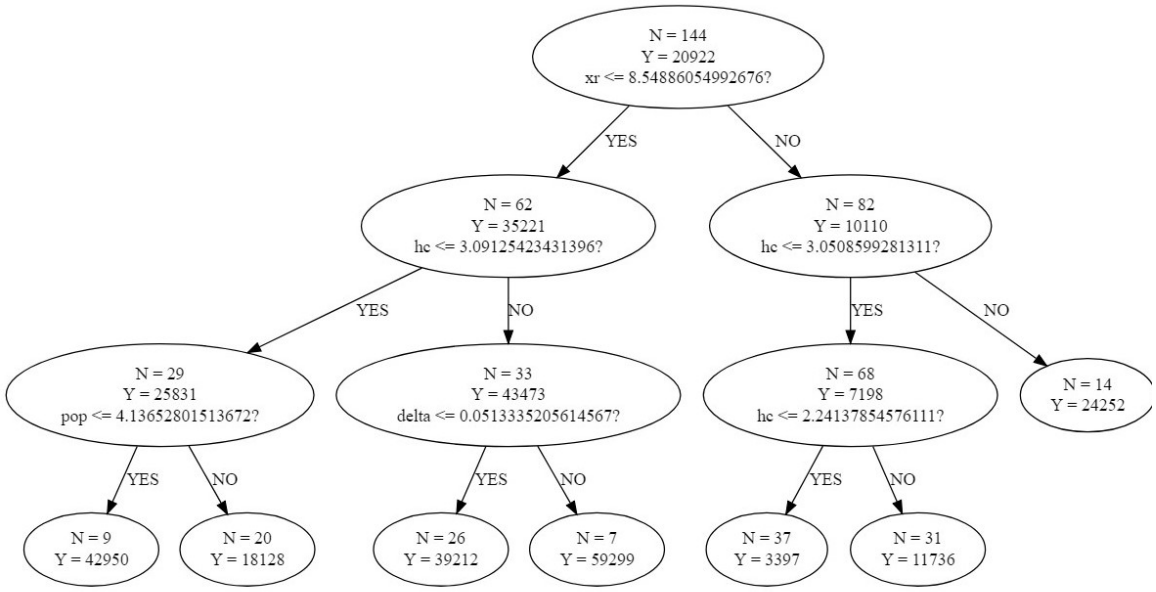

Figure 1: Tree splits for model 1

The N is the size of given subgroup, the Y is predicted GDP per capita of given subgroup. At each root we can see how many observations are there, what is the average Y of given sample, the variable chosen for split and splitting value. For more clear explanation lets look closer to one of the nodes.

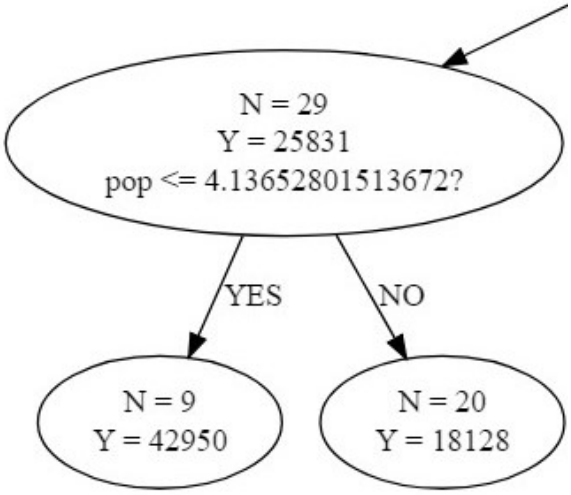

Figure 2: Node explanation

In this node we have 29 observations (countires) with predicted GDP per capita equal to 25831. In the next step we are splitting this sample based on the *pop* variable. The first group contains the countries in which $pop \leq 4.1365$, and the second countries in which $pop > 4.1365$. Because the stop criterion has been achieved the is no more splitting and finally we have obtained two leafs with 9 and 20 countries respectively. For the first group the predicted GDP per capita is equal to 42950 and for the second 18128. Below are model predictions for chosen countries.

Country	Predicted_GDP_per_capita	True_GDP_per_capita
Poland	39211.982	26911.4678
United Kingdom	39211.982	38152.6208
Germany	39211.982	46348.6551
China	18128.245	13464.5385
Ethiopia	3396.684	1793.1672

Due to the nature of the model it's predictions are not flexible, so if the country's dependent variable is an outlier in it's final group the predicted value may differ from the true observed value. It happens mostly with poorest countries for which the value of GDP per capita is extremely low (Burundi - 756, Niger - 916, Zimababwe - 1913).

4.2 Model 2

For model 2 we have decided to take the differences between the log GDPs between 2017 and 2000 as dependent variable. The independent variables are: *pop*, *xr*, and *delta* transformed in the same way as independent variable, that is, respective logarithms are calculated and then subtracted. This time we are not considering *hc* variable and for that reason we may use more observations (the *hc* variable has around 40 missing values) which leaves us with 180 countries.

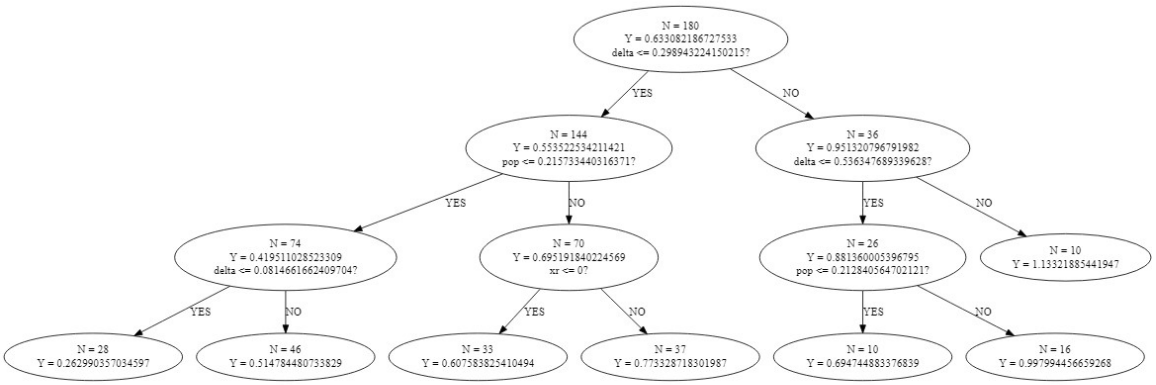

Figure 3: Tree splits for model 2

The interpretation of obtained results is similar for the one in Model 1 subsection.

5 Summary

The whole project allowed us to expand our knowledge about regression trees, by firstly getting theoretical fundamentals and later on programming this abstract knowledge into R language. The final part let us work with real world data, in this case we have explored World Pen Table, which provided us information of relative levels of income, output, input and productivity covering 182 countries. The results of such work are not always perfect, but they are uncovering some information about the world around us.