Assignment 1 for the course

Artem Duplinskiy
Higher School of Economics
St. Petersburg

November 16, 2020

Instructions: This file contains descriptions for the two assignments for the course *Machine Learning and Econometrics in Economics*.

Quick notes:

1. These assignments are mandatory.
2. The deadline to submit reports: Monday, December 14, at 18:30.
3. The assignments are to be made in groups of three students. No more, no less.
4. The report must be sent to `a.duplinskiy@vu.nl`, in pdf format before the deadline mentioned above.
5. The first page of the report must state the names and student numbers of all the group members.
6. In the email subject mention Assignment 1 Machine Learning and Econometrics so I can easily find your assignment.

General Information

The assignment is composed of two parts. The data for both parts can be found on Canvas.

For both assignments, you will spend some time preparing the data and constructing features – this is part of a daily task of a data scientist. The other part is to do the analysis, but often it is beneficial to spend a little bit of time thinking what are you doing and why. In companies there are stakeholders who make sure that data scientists work on stuff that is relevant for business. For these assignments, you take this role yourself.

Not all questions can be solved using software packages. Some questions require you to take pen and paper and think for a bit. When you find a question unclear or impossible to answer, then explain your thoughts. Why do you find the question unclear and how it should have been phrased?

Finally, please note that in life you may have to make some decisions. How do you select tuning parameters? Which statistical test or criteria should I take into consideration to give my final answer? If two tests disagree, then what should I conclude? These questions are part of the daily life of data scientists. Sometimes in data science there is no right or wrong answer. The only thing that really matters, is that you explain carefully and clearly the problems you face, and you justify your decisions convincingly.

Autosuggestions

This assignment is based on data about id's. The goal is to develop an autosuggestion of the place where the id was issued. When a new client applies for a service, for example, a new sim card or a new bank account, information about his or her id must be inserted in the system of a company that provides the service. Each id has a unique number, the issue date, a code that represents where the id was issued (*dept_code* in the data) and the description of the place where the id was issued (*issue_eng*).

Suppose there is an operator that puts the following information into the system: *dept_code*, *issue_date*, *issue_eng*. What we want to do is to develop a system that autosuggests the *issue_eng* once the other two are filled in.

1. Download the data. And let's talk money right away! Is this autosuggestion service valuable for the potential client? How valuable is it? To answer that I suggest you think about the following questions. What is the average length of the issuer field? Suppose the service (bank or a telecom operator) has 100000 new customers a year. Suppose, a minute of work of the operator costs them 0.05 euro. Suppose to look up the information about the passport manually takes 30 seconds. What is the average typing pace characters/minute? How much money could we potentially save with an autosuggestion system that works 100% correctly?

2. To build the system let's split the data into test and train sets. We will use the train set to train the system and the test set to test the performance. To have a performance baseline let's start with something simple. Pick a *dept_code* and for each entry with this *dept_code* in test set autosuggest *issue_eng* with the closest date from the train set. How well does it perform? Use levenshtein distance to evaluate the performance. Make a plot on the x axis - levenshtein distance on the y-axis proportion of the test data that has the prediction error smaller than the levenshtein distance. (see the lecture slides for an example!)

This data could be cleaned and that will improve the performance. There are many entries in the *issue_eng* that essentially mean the same thing, but are spelled slightly differently. Sometimes it is a typo or an extra space. Other times it is an abbreviation instead of the full word. For example, region or province in Russian is "Oblast" and could be shortened to "Obl". Instead of us manually correcting all the options. Let us use a clustering algorithm to find similar descriptions and make an auto correction.

3. Pick a *dept_code* and get the list of all unique descriptions *issue_eng*. Calculate the levenshtein distance between them and put it in an array. Use a clustering algorithm to cluster the descriptions (for example, use the affnity propagation clustering method). Create a new column in your data frame that links each unique description to the "center" of the cluster where each of the descriptions ended up (called "examplar").

4. Now instead of using the actual *issue_eng* we can use the "cleaned version"Repeat 2 but using the new column for prediction.