e-conomics case

# Part 1:

I would like to mention the following important points regarding the code for part 1 of the case:

I decided to try out a predictive model based on a One Vs All analysis (which is a multiclass classification algorithm). The thought behind this idea was that there seemed to be a set pattern behind the bank entry text. A pattern that looked similar among the entries to the same account when manually inspecting a couple of the companies. Using a One Vs All analysis would be able to recognize such patterns and predict the accounts to be used based on the best fit version of such patterns.

There is quite a big variation in the accuracy among the models based on the different companies (spanning from 100% to 22,8%). Generally the models based on the biggest datasets are more accurate. Especially the model that is based on ~90% of the entire data sheet scores a test result of 99,96%.

An interesting step would be to compare the models and find out what exactly makes the models with a high accuracy work so well. This would tell something about the potential problems in predicting the correct account. From theorizing I would expect two major potential problems in the data that could cause the One Vs All model to fail.

One could imagine that if a company uses different accounts to the same type of transactions, then the One-VS-All model will not be able to predict correctly which of the account to use for the given transaction. A potential work around on this problem could be to e.g. get the top 3 answers in such a case and see if the correct answer is among those.

The other problem occurs when there is a large amount of different types of transaction texts going to the same account. Giving more top answers will not necessarily help in this case, as the One-VS-All model will still not be able to acquire a good model. The problem is that the high variance in the data causes the model to be unable to depict any of the different transactions correctly, but instead has a blurry “in-between” picture. A way to potentially work around this could be to use a neural network. Where the One Vs All method requires the bank entry text to show the same or similar pattern, a neural network can form more complex models that form non-linear hypotheses.

My code was only tested on the ability to recreate the result on the same dataset that the model was based on. Splitting the data into a training and a test set (roughly 2/3 and 1/3 respectively) would give a better idea of the models ability to correctly guess new unseen data. It should be noted though, that some of the models are based on so few data-points that it would make no sense to split them up.

# Part 2:

In order to make a model that can create a general prediction based on the data from all companies, the logical step would be to modify and extend the code used in part 1. This modification should be able to use the entire dataset in order to create a more generally covering version of the most likely bank entries to a given account. The major problem in order to do so is the different account numbers and names used by different companies. A good way to work around this could be as follows:

Pool together all models into one dataset and use a cluster analysis to locate accounts that shows a similar model pattern. The bank entry text data used to create these similar account models individually could then be pooled together and treated as a large dataset for one account “type”. The extra added benefit of this solution is that we will get a list of bank account names that are used by different companies to cover the same category of accounts. Given a training set that is large enough and covers a sufficient amount of different account names we could potentially be able to locate not only which of our existing models that would best fit the accounts of a new costumer, but we could also create suggestions on how to optimally create accounts based on what type of bank entries they have created so far.