Data Analytics 2 - Lessons Learned

Tuomo Kareoja

September 19, 2019

Brand Preference Prediction

Lesson 1: Modelling can be used to fix data problems

Instead of keeping dealing with missing data as just a preprocessing step, we can also deal with it as a separate problem. Although there are multiple packages and ways to input missing values in a preprocessing way where we can assess the reliability and effect of different input methods to our final outcome, if we do not care so much about reliability we can fix the problems in the data with separate models before giving it to other analyst or before it gets processed further in the business logic.

It seems that this might actually be quite a common way to deal with this problem judging from a data science game I played recently called "while True: learn()". This game abstracts the nitty gritty away from data science and turns it into simple puzzle game. The most interesting part in the game were the task descriptions that come from real life data science problems. Almost all the task in the beginning of the game were about fixing problems of missing data in databases.

Lesson 2: You have to rethink your metrics if multiple models are performing extremely well

If model accuracy is approaching 100 %, just comparing the accuracy numbers can be difficult. If a metric is over 99 % in multiple models, you have to compare the decimals and this is intuitively much harder. Also plots don't help much: for example AUR curves will be almost totally overlapping in this case.

Even though the metrics are close to perfect in all models, the differences can still be big comparatively. Even though the overall performance can be excellent in to different models, if we compare for example the number of false predictions that these models make against one another, one of the models could still make only half as many mistakes as the other one.

Good solutions to compare models in this case seems to be to take one of the models as a baseline and then compare our chosen metrics by reporting the metrics as percentage of the chosen baseline model. For example if we have a SVM model as a baseline, its AUR would be 1 while a random forest model that is performing better would have a AUR of 1.02, so it would perform 2 % better.

Another solution is to keep the metrics as they are but use zoom in plots to better see the differences. For example we could zoom the AUR plot to only show a small are around one of the models optimal point (Youden index). In this way small differences are easier to notice.

Predicting future sales

Lesson 1: Data that seems only somewhat unrelated to the task at hand can still help

In the task we were interested in only predicting the future sales volumes of a couple of classes of products, but we still created a much more accurate model we trained the model with the full dataset. This was bit surprising because some of the product categories were far removed from the categories that we trying to predict, e.g. how does information about sales in accessories relate to the sales of laptops?

Looked from a another angle this is not surprising. Of course the data that we use in model training is always somewhat unrelated to the thing that we are trying to predict, if not in any other way, then at least temporally. As the data that we use to train the model is always in someway or another removed from our goal, so we need to always evaluate how well it suit our needs. Irrelevance should not be assumed without good reason, and should always be tested. We might surprised by which phenomenon actually can help us make good predictions.

Lesson 2: Sometimes hyperparameter tuning can make the model perform radically better

The usual wisdom is that hyperparameter always makes the model perform better, but not by much. If two untuned models differ greatly in their performance, it is unlikely that tuning will make the worse model actually perform better than the one which performs well out of the box. It seems though that this is not always true and in some cases hyperparameter tuning can have radical effects. This happened to me in this task with a SVM model that performed absolutely terribly before tuning. This might have been bit of a special case because it seemed that without tuning the model did not really work at all to the point that I wondered if it was somehow broken. In the future I will keep in mind that models that seems broken might possible be to fix with good tuning.

Market Basket Analysis

Lesson 1: Always keep the business question in mind

It is easy to get distracted by the technical difficulties in the analysis and completely forget the business goal of the analysis. This is especially true if some of the packages or models used are foreign to you. Without having the business goals constantly in your

mind most of your time might be spent in learning the packages and models and trying them out instead of creating something that has actual business value.

It is also possible that the advised way of answering the business questions is actually not the best one and in this case you should in real life be in contact with managers and bring the problem up. If the management is anticipating some certain approach, but during the analysis it becomes clear that the advised approach does not really answer the business questions, you should change your approach, but also remember to notify the management before hand so that they will not expect something very different than what you will be delivering.

Lesson 2: Ignore the instructions if they seem confusing

Sometimes the advise for completing the task can be overly specific and make it seem that there is one clearly best way of performing the analysis. This can lead you astray if the advise is built more around helping you learn new concepts than actually delivering a good report. Taking the "Plan of Attack" as more of a documentation than guides for delivering the report seems to be a better approach.

Lesson 3: Be prepared to change your approach completely

It might be that you end spending a long time with some kind of analysis path, but then it becomes clear that this path is somehow problematic. Even though you have spent a lot of time working on something does not mean that it has any value. You must be ready to abandon the previous effort and try something new if it is clear that current method is not working.