Business Master of Science Capstone Project
Name: Peter Tanai
Degree Program: MSc in Business Analytics
Project title: What features drive site visitors' interest?
Capstone Project Summary

1 Context and Objectives

Online classifieds website with very successful track record and development. Several data projects to further deepen business understanding to drive future strategy.

The company is keen to understand **what ad features drive site visitors' interest in the particular "for sale" item**. Is there any relation between item features and visitors' interest? Can we predict visitors' interest based on ad features?

2 The Data

I have received a data table including the **population of ads for a period of 12 months**, altogether cca. 100k records. For each advertisement the ad features were listed (characteristics that the advertiser can specify when posting the ad) along with two variables measuring interest of site visitors.

3 The Solution

My solution followed a **classic machine learning approach:** splitting data into train and test. Trian data applied for calibrating the models with cross-validation. Afterwards the best model was selected and tested on the test set. Four different models were developed (Logit Regression, Elastic Net, Random Forest, Gradient Boosting Machine) in line with supervised learning approaches that could be applied to the problem (I had binary targets meaning I went for probabilty prediction and then classification).

4 Approach

Analysis followed a five step process:

- Data cleansing – data required extensive recoding regarding extreme values, different coding of Nas. Dropped around 1% of records.
- Data exploration – my target variables had a long-right tail (lots of zeros, almost power-law distribution). This is why I have decided to build binary targets from my quantitative targets, and

not focusing on predicting number of interests but rather predicting if there was any interest at all (0/1). Interestingly qunatitative features of ads did not seem to correlate with targets, which was a surprising finding, as later on these variables seemed to be decisive / important variables for most models. There was some interaction between features (independent variables), however, I did not consider those when building my logit model

- Label and feature egineering – as mentioned above my labels (targets) had to be transformed into binary variables. Built some derived variables that could be used for modeling and did not exist in the raw table. I also decrease the number of category options of factor variables (categorical variables), this is important especially for formula based regression models.
- Modelling – I applied limited parameter tuning while building the four models. Reason: the data table was big, and even with a small % of train set it was not very fast to calibrate models. I started out from best-practice parameter sets, and then adjustment them somewhat. The key parameters for tuning were: alpha and lambda for Elastic Net, mtry and min. node size for random forest, Nrounds, col sample, sub-sample, eta, gamma, min.child weight for GBM.
- Evaluation and results – see next section

5 Evaluation and results

Overall **GBM outperformed other models** even with limited calibration. My baseline logit regression model could have been further improved, still it was far behind random forest and GBM.

In each 4 models **broadly similar variables were considered important**. Results were different from what I would have expected based on data exploration. It was a key learning to me that data exploration can only provide with limited insight in how the model should look like. This is why in addition to formula-based models we should also build black-box type of models so that we can check what is the maximum potential RMSE or AUC that we can make out of the data.

After I selected the best model, I wanted to also **predict the classifications** for the different records. The company gave me the insight that we should minimize false positive (since for the company it might be a bigger mistake and would cost more if we falsely predict an interest vs predicting a false non-interest). I tried 0.5, 0.6 and 0.8 thresholds (as I did not have explicit loss function available). While higher the threshold the better we eliminate false positives, at the same time we almost fail to identify any true positives. This is an important problem when target variable is very skewed with lots of zeros, that is event probability is low.

6 Overall conclusions

Can we predict visitors' interest based on ad features?

Yes we can, with GBM and limited parameter tuning arriving at AUC of over 0.8.
Classification accuracy of over 0.75 at 0.5 threshold. High thresholds make it very hard to identify positives given low even probability

What ad features drive site visitors' interest in the particular "for sale" item?

Add features depend on the chosen model, but broadly key features are similar.
It makes a big difference from the prediction perspective if an advertisement is posted by an individual, and also some key quantitative features make a difference.

7 What could be further improved?

Better modelling

- Doing some more parameter tuning, fine tuning around optimum
- Trying out deeper trees for GBM (e.g. Depth of 10)
- Modeling level of interest (once there was any interest) and separate e.g. less than 5 and more than 5 interests
- Improve logistic regression via inclusion of quant variables and interactions

Making my code production ready

- Consistent deployment of packages
- Developing more functions and making code simpler, shorter
- Keeping company-used labels intact as much as possible, using numeric versions of factor variables in modeling
- Developing grouped variable importance plots and generating more visual graphs

Better cleansing

- Deploying cleaner code (e.g. Regex over replace etc.)
- Minimizing recodings and follow company approach in cleansing

8 Key learnings

Key learnings for myself and also for students working on capstone projects in the future:

- In spite completing lots of assignments during the program, doing a project from scratch takes time, and has to be **planned carefully**
- Pacing the work and **distributing over a period of 1.5 months** makes it mission possible!
- **Data cleansing** takes most of time and one should always **align best approach with client** to make sure reusability (limited recoding, using numeric version of factors, handling of NAs, etc.). Each company might have a different practice for this.
- **Training models take also lots of time** given size of data and provided limited computer resources. Need to do **parameter tuning in a smart way and using small train ratio** when calibrating models.
- There is a solution for everything, just need to look extensively. Usually **stackoverflow will give an answe**r.
- Need to spend much more time to **explore packages**, in order to be in better control of what's feasible