Data Analytics 2 - Product Type Sales Prediction

Tuomo Kareoja

Version Number	**Changes**	**Date**
0.9	Basic layout	02.09.2019
1.0	Finished Report	03.09.2019

Predicted Sales for New Laptops, Smartphones, Netbooks

Picture of

brand_preference_plain.png

These are results of the brand preference question after we added in our predictions. We can wee that Sony is clearly the more preferred brand.

brand_preference_marked.png

If we separate from the non-missing answers and the predicted answers, we see that there were quite a lot of predicted values, but that the brand preference distribution is the same for both. We can see this even more clearly in a percentage plot.

brand_preference_marked_percentages.png

If our model is correct (and it is very accurate) this suggests that the missing brand preference values do not bias our estimate of the brand preference distribution, so in this sense adding in the missing values does not give any value. On the other hand our model is very accurate and in the future the answers might not be missing at random. If this is the case, we now have proven that we can make an accurate model that can correct this bias.

Chosen Model and Its' Performance

The model used for the predictions is a C5.0 classification tree. From the the graphs below we can see that on all the important metrics this model performs really well with the train set accuracy being 93.5 % and with test set 92.3 %. This shows that the model is accurate and at the same time the risk of overfitting is low with the test and train metrics being so close to each other. This goes for all the other performance metrics as well.

cm_c5_train.png

cm_c5.png

As the C5.0 decision tree is too complex to understandably visualize. We can instead look at the relative model feature importances:

c5_full_feature_importance.png

We can see that by far the most import variable is the salary of the customer. If we plot the distribution of salary with the brand preference, we can see that there are clear differences in brand preference, but that these are quite complex and nonlinear and would have not been easily captured with a linear model. Tree type models catch these kind of complex relationships much more easily.

brand_salary_distribution.png

Model Comparison and Performance

In total three different model types were tried: C5.0, Random Forest (RF) and Extreme Gradient Boosting (EGB). All of these model are tree based models and all performed excellently. This made choosing the right model quite problematic.

From the cross validation boxplots we can see that all models perform very well in all metrics and that the results have a tight spread. This suggests that the models have good performance would probably generalize well to previously unknown data as well.

`dotplot_comparison.png`

The small differences between the models can best be seen in the ROC-curve calculated with test data:

AUC_comparison.png

The models are almost indistinguishable, with our chosen C5.0 model maybe somewhat getting a head around the optimal (Youden Index) point closest to the left upper corner of the graph.

If we look at the results of the other two models more closely, we see that the Random Forest model has the smallest difference between the metrics from train and test datasets. This would suggest that this model is the most generalizable of the models tested. It did not as well as the C5.0 with AUC-measure (previous boxplot) and sensitivity, but beat it on specificity. The confidence interval of the cross validated metrics were overlapping though so we can't say with high confidence that C5.0 is strictly better.

`cm_rf.png`

cm_rf_train.png

Extreme Gradient Boosting had a metric score difference between train and test datasets, which was comparable to C5.0 model. It also performed less well than C5.0 on AUC-measure and specificity, but had the best sensitivity of all the models. All in all also a very good model, but slightly worse than C5.0 and Random Forest.

cm_egb.png

cm_egb_train.png

Ensemble Models?

When looking at the cross validation model AUC-scores we see that the correlation between these scores is very low between the models (AUC scores don't line up neatly on the diagonal).

model_prediction_correlations.png

This means that the model seem to be making different kinds of mistakes. We could try to exploit this by building an ensemble model our individual model results. This was tried, with the best model being a combination of C5.0 and Random Forest. This model actually had slightly better scores on all metrics than individual models, but the improvement was very small and at the same time the model showed signs of overfitting with it having the biggest difference in performance between train and test set. Because the increase in performance so small it was deemed unwise to increase the complexity of the model by using an ensemble method. With the bigger difference between the train and test set this complex model was showing signs of overfit although it still performed better than other models with the test set.