DMC 2013 Report

Team: Uni_Aristotle_Thessaloniki_1
Members: Eleftherios Spyromitros-Xioufis & Emmanouela Stachtiari

How To Create The Best Model With The Highest Error

- Keep only the last line from each session to create a single row instance.
- Evaluate various algorithms from the Weka machine learning toolbox to find one that works well on this problem. Model selection was based on lowest Mean Absolute Error (MAE) calculated using 3-fold cross-validation on the training set. From this step we selected MultiBoostAB (Weka's implementation of MultiBoosting[1]) with J48 (Weka's implementation of C4.5) as the base classifier and 100 boosting iterations. This model gave a MAE of approximately 0.032.[2]
- The last step involved some feature engineering. We computed several additional attributes and evaluated the contribution of each attribute separately to our model. We selected those attributes that individually gave us the largest error reduction and at the same time did not seem redundant with the initial attributes or with each other. Our final model contains the 65 attributes listed in Final Dataset
- Table 1. Pay attention to the 36 attributes with "a{xx}" names. These attributes represent the graph (adjacency matrix) of a user's transitions between different purchase processing steps and try to model the dynamics of a session. There are 6 nodes in this graph (5 bsteps + 1 for missing bstep) and one edge (with a weight of 1) for each transition from one bstep to another across a session. Finally edge weights are divided by the total duration of the session.[3] This feature engineering procedure improved our MAE from 0.032 to ~0.029, a 10% error reduction!
- According to our MAE, we expected a final score of 5111*~0.029 = ~147 (or slightly better because we used 3-fold cross-validation instead of 10-fold) that was really close to our final score of 140.78 (5111-4970.22). However, as it turned out we returned the probability of not making an order and thus got almost the highest error possible![4]

---

[1] Webb, Geoffrey I. "Multiboosting: A technique for combining boosting and wagging." *Machine learning* 40.2 (2000): 159-196.
[2] We replaced missing values of all nominal attributes with an extreme value (-100). This replacement was important for J48 to work well on this data.
[3] Without this normalization the inclusion of the graph variables did not improve our model!
[4] It is interesting that several of the top teams predicted 1/0 decisions instead of probabilities. It can be proven that this is not optimal for all classes of prediction models and it was NOT optimal for our model.

Final Dataset

Table 1: Attributes included in the final model

Name	Type	Description
startWeekday	{1,5,6,7}	provided
duration	numeric	provided
cCount	numeric	provided
cMinPrice	numeric	provided
cMaxPrice	numeric	provided
cSumPrice	numeric	provided
bCount	numeric	provided
bMinPrice	numeric	provided
bMaxPrice	numeric	provided
bSumPrice	numeric	provided
bStep	{1,2,3,4,5}	provided
onlineStatus	{y,n}	provided
availability	{co, cno, mo, m, mno, cnd, mnd}	provided
maxVal	numeric	provided
customerScore	numeric	provided
accountLifetime	numeric	provided
payments	numeric	provided
age	numeric	provided
address	{1,2,3}	provided
lastOrder	numeric	provided
bSPcSPRatio	numeric	bSumPrice/cSumPrice
bCountNorm	numeric	bCount/duration
overpriced	{y,n}	(bSumPrice>maxVal) ? y : n
bAvgPrice	numeric	bSumPrice/bCount
thisHourCirc1	numeric	startHour+duration circular cos
thisHourCirc2	numeric	startHour+duration circular sin
startHourCirc1	numeric	startHour circular cos
startHourCirc2	numeric	startHour circular sin
a00	numeric	missing -> missing
a01	numeric	missing -> bstep1
a02	numeric	missing -> bstep2
a03	numeric	missing -> bstep3
a04	numeric	missing -> bstep4
a05	numeric	missing -> bstep5
a10	numeric	bstep1 -> missing
a11	numeric	bstep1 -> bstep1
a12	numeric	bstep1 -> bstep2
a13	numeric	bstep1 -> bstep3
a14	numeric	bstep1 -> bstep4
a15	numeric	bstep1 -> bstep5
a20	numeric	bstep2 -> missing
a21	numeric	bstep2 -> bstep1
a22	numeric	bstep2 -> bstep2
a23	numeric	bstep2 -> bstep3
a24	numeric	bstep2 -> bstep4

a25	numeric	bstep2 -> bstep5
a30	numeric	bstep3 -> missing
a31	numeric	bstep3 -> bstep1
a32	numeric	bstep3 -> bstep2
a33	numeric	bstep3 -> bstep3
a34	numeric	bstep3 -> bstep4
a35	numeric	bstep3 -> bstep5
a40	numeric	bstep4 -> missing
a41	numeric	bstep4 -> bstep1
a42	numeric	bstep4 -> bstep2
a43	numeric	bstep4 -> bstep3
a44	numeric	bstep4 -> bstep4
a45	numeric	bstep4 -> bstep5
a50	numeric	bstep5 -> missing
a51	numeric	bstep5 -> bstep1
a52	numeric	bstep5 -> bstep2
a53	numeric	bstep5 -> bstep3
a54	numeric	bstep5 -> bstep4
a55	numeric	bstep5 -> bstep5
order	{y,n}	class