Case Study Questions

Luis Valino Borau

30.06.21

1. To make a meaningful classification model we need to make sure of a few things. First, we need enough examples from each target class, so that the model can learn a generalizable boundary between the different classes. In this case, we see we have less than 30% instances of female users. This shouldn’t be a big issue, but a split closer to 50% would be better.

Another important factor is whether we have enough data. We are given 3700 rows for only a handful of features, so it seems reasonable; however, we might encounter a problem, since there are many missing values (60%) in the variable device\_name, which is one of the most telling ones, given it may contain the name of the user. This does not mean we should discard the variable since we can get useful information from the remaining 40% and group the rest in one big category. One idea to extract useful information from the variable would be to create dictionaries for male and female names based on the countries we operate. Then we could split the name into words and search for names included in the dictionaries (considering of course the occasional possessive prefix/suffix) and thus create one category for male names, one for female ones and one 'unknown', where we would also include the 60% missing values.

Given the fact that we have several events from the same users, we should make sure we have a large enough pool of individual users to make generalizable predictions, so it would be wise to see the variability in user\_id/device\_name. If we have 3700 events corresponding to 20 users we are in severe danger of overfitting.

Regarding NaN values in App and Ad category, we may just substitute for the most frequent category (or distribute them randomly according to category frequency) or assign them their own category. Given there are so few such instances, they shouldn’t be an issue. Another issue is the number of categories we have and the frequency of each one. If we deem that a problem we could group them into broader categories. In any case, I would first try with the categories we have, since it would make sense that they have predicting power. Perhaps given the overwhelming amount of ‘News’ events, we could reduce the number of categories there to ‘News’, ‘Weather’, Health’, ‘Dating’ and ‘Others’. I would in any case first try with the data as it is.

There is one important problem with the Click variable. With only 18 ‘Yes’ events in 3700 rows, this variable does not provide a large enough sample to infer anything and should be dropped.

The variable interaction\_with\_app on its own probably does not yield much information regarding gender, but it might do so in conjunction with others.

2. To find the most meaningful variables I would perform several steps. It is very fast to perform correlation analysis to see which variables are linearly correlated to the target, so I would do that first. I would also plot the data from each feature colored according to their target value, that way it would be straightforward to see whether there are categories that separate the target value clearly. Even three-dimensional plots or heatmaps could add valuable information as they allow us to see the influence of several features at once and given the low dimensionality of our data they seem to be especially valuable.

In any case, I would fit a decision tree or a random forest classifier to the data and after selecting the optimal model I would take a look at the feature importance. In this case, it should be fast and it would give us the desired information.

3. I would choose a random forest classifier. That way we would have a clear idea of the rules behind the predictions the model makes and we could compute feature importance directly. Random forests also tend to overfit much less than decision trees and give us the chance to perform out-of-bag error calculations to have an estimate of the accuracy of our model. They are also simple to build; however, if the desired accuracy is not reached, boosting methods could be used such as AdaBoost or GradientBoosting. If we had several orders of magnitude more data and features, a decision tree-based approach would make less sense and a neural network might be more appropriate.