Final Project STATS 495

December 23, 2017

By Meron Gedrago and Wayne Maumbe

kNN versus CART for the prediction of categorical variables

Our aim for this project was to compare machine-learning methods used for the prediction of categorical data. We focused on comparing the k-Nearest Neighbors (kNN) method to Classification and Regression Trees (CART) methods. We employed the Expedia Hotel Recommendations Kaggle Competition for our analysis. The aim of this competition is to predict what type of hotel an Expedia customer would book. This competition's aim served our aim well because it's outcome variable, hotel cluster, is a categorical variable with 100 levels. We found that these algorithms which predict categorical data are optimized for usage with numerical data and hence are not easy to use specially if handling big data.

The data provided by Expedia for both the training and test data was very big. The data was provided compressed in a .csv.gz file and hence required extraction. We successfully extracted it using 7 zip on a Windows laptop and using WinZip on a Mac laptop. The training data was 3.8 GB, which translated to about 30 million rows of 24 variables, and the test data was 263.7 MB that is 2, 528, 243 rows and 22 columns. The variables detailed information pertaining an Expedia user's search information. This include predictor variables that entail the user's location, desired check in day, desired number of rooms and hotel location. Our approach in the exploratory data analysis was to consider variables as divided into the following categories:

- User's information for example user location
- Hotel's information for example hotel location
- Search's information for example the check in date

Our aim was to predict the outcome predictors from each of these categories. We used correlation to find which of these variables would be mostly correlated to the outcome variable hotel cluster. We found that the highly correlated variables were:

- hotel continent
- package deal
- Length of stay*
- hotel market

We used ANOVA to test if there was any significant effect on the distribution of the hotel clusters given our choice of predictors. The ANOVA test confirmed there was an interaction effect amongst these levels of these predictors and the outcome variable hence suggesting that using these predictors is justified because of the implied significant effect on the outcome variable.

Using these predictors, we used the following approaches to get to our final models.

CART Model

Our first approach was to run the cross-validation and train the model a random sample of the training dataset. This did not work. The reason was that this method couldn't handle too many levels in the outcome variable. We mitigated this by condensing the 100 levels of hotel cluster into 9 levels we duped hotel popularity. In this condensation we created an ordinal categorical variable that showed increasing popularity by number of searches from level 1 to 9.

Using this new outcome variable, we trained and then predicted hotel popularity using the test set. However, for submission to Kaggle we had to expand the outcome back to the original 100 levels. For this we used weighted sampling of the hotel clusters. This approach gave us a score of 0.009.

We also performed the same analysis on searches that resulted in a booking. We did this because the test data set is comprised of only searches that resulted in a booking. However, we were not able to obtain a score for this because of time constraints. We believe this might have boosted our score since it would have been a trained on a bigger and a more representative training dataset.

kNN Model

For this method, we used the training dataset with only the searches that amounted to a booking. We first used this filtered dataset for this model, but it had too many points hence we got errors suggesting we cut the size of our dataset. This error was because a single point in the test set was equidistant from too many points, hence giving a 'too many ties' error. We finally used a dataset with 500 observations to resolve this error.

The training dataset with 500 observations was able to give us a score of 0.031 on Kaggle. This was an improvement from our CART model and our best score for this competition.

Challenges

The most notorious problem we encountered was with runtime. We tried to mitigate this by using different computers and the Amherst RStudio server to vary computing power and using smaller training sets, but we always had to wait for long when opening the project, running code chunks and knitting the Rmarkdown. We often had to terminate/restart RStudio and as a result this hindered our progress. To put this problem in perspective, for kNN, the average runtime on the least sized training dataset (n=500) we had was 8 mins.