Cornell Tech CS 5304 Assignments #2 and #3
April 12, 2017

Assignments #2 and #3 are linked. The same data set will be used as the base for each assignment, but different processing will be completed by different due dates. These assignments are not graded on the tier approach.

The data and the underlying problem is from the KDD Cup 2012, Track 2. The original problem asked for the data scientist to **Predict the click-through rate of ads given the query and user information.**

The description of the problem can be found at:
https://www.kaggle.com/c/kddcup2012-track2#description

The data for the problem can be found at:
https://www.kaggle.com/c/kddcup2012-track2/data

This assignment contains an optional Big Data component. Students opting to work on the Big Data version must note that clearly within their assignment submissions to receive credit for the work.

Like assignment #1, the solution should be built using Spark and the Spark libraries. Data set submissions should be compressed using zip. See the TAs comments in Slack for exact naming conventions.

ASSIGNMENT #2, due by April 23rd, 2017

We consider data used to predict the click-through-rate (pCTR) of online ads. An accurate model is necessary in the search advertising market in order to appropriately rank ads and price clicks. The data contains 11 variables and 1 output, corresponding to the number of times a given ad was clicked by the user among the number of times it was displayed.

For each instance (training example), the input variables serve to classify various properties of the ad displayed, in addition to the specific search query entered. The identifiers include unique identifiers for each query, ad, keyword, advertiser, title, description, display url, user, ad position, and ad depth (further details available in the KDD documentation).

PART 1: In both the training and test set provided …

To reduce the data size, combine instances with the same user id, ad id, query, and setting, so that the output may take on any positive integer value.

PART 2 (non-Big-Data): identify the top 25,000 instances with the same ad id and query by frequency from the training set. Select all instances using these ad id/query id combos for the training set. Only operate on this subset of the training for the remainder of the assignment. Operate on the full test set for the remainder of the assignment.

PART 2 (Big-Data): operate on all of the instances in the training set for the remainder of the assignment.

PART 3: Compute a position and depth normalized click-through-rate for each identifier, as well as combinations (conjunctions) of these identifiers.

PART 4: Annotate each instance in the training and testing set with the normalized click through rates. Submit these 2 data sets with the code for parts 1 – 4.

PART 5: Shape the data into feature vectors suitable for input into machine learning. Describe a selected learning model and the shaping selected using empirical analysis of the data. Note that you may select any model type from the Spark library and your submission will not be judged by whether your selected model turns out to be the best model for the problem. For assignment #2, you DO NOT need to implement the machine learning model. Submit the code and the training/testing data sets generated for Part 5.

ASSIGNMENT #3 due by April 30th, 2017

Using the training set and test generated using the instructions for assignment #2 …

PART 1: Use machine learning to predict the click-through-rate. Performance for this task should be measured using the Area Under Curve (AUC) metric. In short, the AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. The AUC results should be computed for the training set and for test set that has about 2 million examples. Describe the experimental setup and discuss bias-variance trade-offs in your experimental design. Submit the code and the predictions for the training set and the test set.

PART 2: Conduct an error analysis of the model's predictions on the training and testing set. Describe limitations of the model exposed by the analysis.

PART 3: Suggest a modification to the training set to improve model performance. Justify using the error analysis.

PART 4: Implement the suggested feature change to the original model, report results, and compare the performance to the original model. Submit the code and the data sets with the new predictions for the training and test set.