Winning Model Documentation

Name: Owen Zhang
Location: Edgewater, NJ
Email: zhonghua.zhang2006@gmail.com
Competition: Avito Context Ad Clicks

1. Summary
    a. This comp has fairly large data and non-trivial structure. I spent about ⅔ of time doing feature engineering and ⅓ training models.
    b. It is interesting that pure xgboost models were performing so well that I never had the need to build other models to add to the ensemble.
    c. The structure of the data provide interesting feature engineering opportunities.
    d. The data are large enough that there is virtually no risk of overfitting public LB.
2. Feature Selection / Extraction
    a. I went through each data table to extract features, mostly one table at a time.
    b. There are several kinds of features:
        i. raw features, such as Position, HistCTR, etc
            1. position is a quite useful feature, as expected.
        ii. simple time based features, such as time of day and day of the week
        iii. sequence based features, such as # of ads seen up to a given impression
        iv. average prior response, such as # of ads clicked up to a given impression
        v. text based features, using ngram/tfidf/svd, such as SearchQuery
        vi. text similarity, for example, similarity between SearchQuery and Title
        vii. simple text stats, such as # of characters in Title
        viii. price based features, such as average price for given category views of a given user
        ix. entropy based features -- how diverse/concentrated a user's history is
        x. categorical features that are encoded with click rate, adjusted for credibility
    c. SearchInfo, UserInfo, AdsInfo, SearchStream, VisitStream all provide many useful features
    d. I only created one feature using PhoneRequestStream, which is marginally useful.
3. Modeling Techniques and Training
    a. xgboost was the only type of model used
    b. to speed up training time, I down sample non-events at 50:1 ratio.
    c. xgboost was run over different down sampled data, with different random seeds.
        i. resulting predictions are averaged to produce final prediction.
    d. two different xgboost models (beyond simple averaging the same model with different seed and sample):
        i. 1 with 65 features on last 200 impressions

        ii. 1 with 61 features on all impressions
4. Code Description
    a. Please see README.md
5. Dependencies
    a. Please see README.md
6. How to generate the solution
    a. Please see README.md
7. Additional comments and observations
    a. It is interesting to see that while the previous two CTR competitions (criteo and avazu) were both won by FFM models, this time xgboost clearly out performed FFM, or VW models. I tried both FFM and VW, but found their performance way worse than xgboost, and won't add much to ensemble either.
    b. It is worth noting that some of the stronger features, such as number of search results for each query, are not really meaningful predictors. Such features are very powerful because the evaluation metric is overall (across all queries) logistic loss.
        i. It might be worth considering to use intra-query rank metrics, such as NDCG, instead of logistic loss, as model evaluation metric.
8. Simple features and methods
    a. The modeling approach is quite straight forward, just xgboost. So I don't see any way to make it much simpler
    b. It is probably possible to reduce the number of features but don't lose too much prediction accuracy. However the model has only 65 features, which isn't a big number to begin with.
    c. The training process can be significantly sped up by choosing a smaller data sample
        i. Smart sample can be applied even before feature extraction -- keeping all searches with clicks, and only a small portion of searches without.
            1. This will make the model performance drop a little.
            2. Also in order to make predictions on all test data points we still need to process quite a bit of data.
        ii. Keeping only the last N impressions.
            1. Keeping last 200 doesn't impact model result much, and cut the data size to about half.
            2. Keeping even smaller number probably will still produce reasonably good models.
9. Figures
    a. None.
10. References
    a. https://github.com/dmlc/xgboost