Online Ad Click Prediction for Outbrain on Kaggle

CSE 6242: Project Progress Report

Joseph Phillips, Shoili Pal, Kyle Zimmerman, Kaustubh Mohite, Arjun Mishra, and Gregory Parry

Georgia Institute of Technology

Outbrain is an internet advertising company which generates revenue when ads are clicked by a user. Through Kaggle, Outbrain has provided access to a large relational dataset (>150 GB) containing information about user ad-click behavior. This data set centers around the many ads that were presented to users from June 14th to June 28th of 2016. Ads were presented in groups called "displays". Displays contained from 2 to 12 ads each. For each display, one ad was clicked. In the training portion of the dataset, there is a variable indicating which ad was clicked. In the test portion, this variable is absent. The purpose of this project is to order ads in each display of the test set by decreasing predicted likelihood of being clicked. Many potentially predictive variables in the dataset provide information about the ads, the displays, or both. These variables include user id, user page views, user geographic location, time stamps, advertiser id, and more. Figures 1, 2, 3, and 4 in the appendix are included to provide further information about some of these predictive factors as well as the scale of the dataset.

Survey of the Literature

Many different approaches have been used to try to predict user clicking behaviors. These include field-aware factorization machines (Juan et al., 2016), Markov chains (Sarukkai, 2000), neural networks (Baqapuri & Trofimov, 2014; McMahan et al., 2013; Zhang et al., 2014), and user modeling (Attenberg, Suel, & Pandey, 2009; Beales, 2010; Wang & Chen, 2011; Wang et al., 2013). Markov chains are problematic because even as a whole, users do not tend to act completely at random (Liu et al., 2012). Research has demonstrated that ads and other links are chosen for their appeal rather than random choice (Liu et al., 2012, Wang et al., 2103) Some ads even appeal to psychological needs (Wang et al., 2013). Further supporting this point, efforts to optimize ad presentation have been successful (Beales, 2010; Besbes, Gur, & Zeevi, 2016).

Though user behavior is not random, modeling user behavior presents problems of its own. Targeting specific users also requires more computing power that may create a slower

experience (Nath et al. 2013). This type of targeting has also been shown to have the drawback of less ad views per advertising dollar (Beales, 2010).

Methods

For this project, emphasis was put on indirectly determining the quality of ads. The conditional probabilities of an ad being clicked were generated with Python script. This script implements an original ad scoring system that was inspired by Google's PageRank algorithm. The PageRank algorithm ranks pages based on their connection to other pages rather than content analysis (Ravi, Leng, & Singh, 2013). In the Outbrain Dataset, all information about ads and displays has been coded. For this reason, the relative quality of ads must be inferred from ad-click behavior in the training portion of the dataset. Ads were scored based on their performance as compared to the probability of being chosen at random. This method may be thought of using the classic marbles in a jar example from introduction to probability theory. If there are 10 blue marbles and 10 red marbles in a jar, then over the long run you would expect to pick blue marbles at a probability of 0.5. However, if blue marbles are picked at a greater rate than probability, this could indicate that something besides color was different about the blue marbles as compared to the red marbles. For instance, if the blue marbles are larger than the red marbles, it would make sense that they would be chosen more often. Our method applies the same concept to ad quality. If an ad is clicked at a greater rate than probability, it may be an indication that the ad's attractiveness to users is greater than other ads which are clicked at or below probability. This approach was based on the findings in research that ads are not generally chosen at random, but by user preference (Liu et al., 2012, Wang et al., 2103). This approach should be better than the state-of-the-art, because it can be used alone or as a feature in state-of-the-art modeling techniques. Furthermore, to our knowledge this method of indirectly determining the quality of ads has not been used before.

Because neural networks have been shown to be successful in the past at predicting ad-click behavior (Baqapuri & Trofimov, 2014; McMahan et al., 2013; Zhang et al., 2014), a neural network was implemented to compare with the ad score model. A random forest algorithm was also developed for comparison. The xgboost random forest model and the Neural Net (Multi-Layer Perceptron) were put together using Python with pandas and sklearn. The features used were campaign_id, advertiser_id, source_id, publisher_id, publish_year, publish_month, category_id, topic_id. Since we had over 80,000,000 rows, the data was split into two training sets and one testing set. Using this method, the data was small enough to fit in random access memory and be processed. We combined the results of the models trained of the halves of the data by averaging them.

Clustering was also used to minimize comparison groups. The Outbrain dataset contains many discrete categories for several anonymized variables. These were minimized using clustering which has been demonstrated to be effective preprocessor of variables with unknown values (Goswami & Shishodia, 2013; Unver & Gundem, 2016). We used the Python implementation of the k-modes algorithm (package kmodes). K-modes clustering is an extension of k-means for categorical data (Huang, 1998). We clustered Documents based on the following categorical attributes (all anonymized by Outbrain in the training data set): source_id, publisher_id, category_id, and topic_id. These were selected based on a trade-off between computation cost and improvement in the performance of the model. These clusters were then matched to ad displays based on the documents on which the displays were located. The clusters were then used as predictive groups in our final ad score model.

While exploring the data, we considered several different ways to visualize the data. One was a spatial-temporal cube that represents both geographic location and time as suggested by Kisilevich, Mansmann, Nanni, and Rinzivillo (2009). This graphic would have included text-

based search capabilities which has been demonstrated to make user interaction faster and easier (Cartwright et al., 2001). Though this would have been interesting, it did not seem vital to our analyses. Our second consideration for the interactive visualization was a platform for analyzing the results of different models for this type of data. Because Kaggle only allows two submissions per team per day, we realized that we would need to develop a reliable validation system to access the quality of the models and model variations we intended to test. We also considered what would be most useful to Outbrain moving forward. The intent of the Kaggle competition was to crowd-source a better algorithm for ad prediction. For these reasons, we decided that the best visualization would integrate with the modeling algorithm and produce visual results whenever a new version of the model was tested. Our visualization, therefore, includes both the current model's performance as well as the ability to compare the results with previous models. Figure 5 in the appendix shows the Ad Score Verification System being used to compare the results of grouped data.

Experiments/Evaluation

The experiments for this project were designed to determine which model would produce the most predictive results. Further experiments were conducted to determine what methods were most likely to produce the most predictive results. For instance, the ad score technique faced a problem when it encountered ads on which it had not trained. This could occur in one of two ways. The first was that there were some ads in the test set that never appeared in the training set. The second was when data was sliced to improve the predictive ability of the ad score system based on different factors. In these cases, there were occurrences of ads which appeared in the test set for that slice and in the full training set, but not in the slice's training set. The question that needed to be answered was which method of imputation was best for scoring these "unknown" ads.

The metric used to measure the predictive ability of the models was mean average precision. The models were required to list ads in decreasing order of their likelihood of being clicked. The mean average precision calculation results in a value between 0 and 1 that reflects higher values as clicked ads are ordered closer to the first position. Each ad display received an average precision score. Because there was only one clicked ad in each display, the numerator in this average was always one. The denominator was the number of attempts before correctly listing the clicked ad. For example, a clicked ad ranked first would receive an average precision of 1, whereas a clicked ad ranked 12th would receive an average precision of 1/12. These average precision scores are then summed and divided by the total number of displays ranked. This final score is the mean average precision. Using this metric, we also developed a yardstick for measuring whether two precision scores are significantly different. The greatest variation possible for the mean of average precision would occur if half of the ads were ranked 1st and the other half were ranked 12th. Using this arrangement, we determined that the standard error of the mean of average precision for our validation sets (n = 1,687,459) was 0.0004 and the standard error for the Kaggle test set (n = 6,245,533) was 0.0002. Though these values are arguably small due to the sample sizes, a change of these values in mean average precision could reflect the movement of 1299 ads from 12th place to 1st in the validation sets and the same movement of 2499 ads in the Kaggle test set. Because the maximum value of variance was used in determining the standard error, these were the maximum values of standard error for each. Therefore, we could easily determine if two models were significantly different by comparing the two mean average precision scores and determining whether they were within two maximum standard errors of each other.

Among the models, the ad score system (MAP@12: 0.64846) outperformed both the xgboost model (MAP@12: 0.57942) and the neural net (MAP@12: 0.48666) algorithms in its

ability to predict ads. However, all models performed better than a model with randomized order submissions (MAP@12: 0.47266). Figure 6 in the appendix shows the results of each as compared to the ad score model. Furthermore, improvement was made on the random forest model over the last 24 hours. We may continue training this model with more features for the Kaggle contest to see what results we can get.

Three factors were determined to increase the predictive ability of the ad score system. Whether the user was in the United States or not located in the United States was the first. The second was whether the user was accessing the internet with a mobile phone or with a tablet or computer. Finally, k-modes clustering, mentioned above resulted in five clusters of displays. Two of the clusters (MAP@12: 0.66826 & 0.64963) were found to be relatively predictive. The other clusters were too small in sample size and were combined. Using these three factors, 12 training and test sets were subdivided from the whole dataset, processed by the ad-score system, and reconsolidated into a complete predictive submission. The result was our best standard validation result, but fell 0.00027 mean average precision points below our best Kaggle score of 0.63867. However, the public leaderboard on Kaggle is only based on 30% of the test set, so we have hopes that this model will do better in the final competition.

Three methods of imputation were tested for unknown ads. One was the ad's probability of being picked in that display. Another was the mean of ad scores in the ad's training set. Finally, z-scores were generated from the ad scores of the full training set. The z-scores were then used to impute the score of "unknown" ads when the data was sliced. Of the methods of imputation tested, z-scores performed best followed by probability imputation which was followed by mean imputation. However, z-scores could not be used on the types of "unknown" ads which never appeared in the training set. Therefore, a 'zp' parameter was added to the ad

score algorithm which allows it to impute ad score values for ads not trained in the slice with z-values and ads which were not in the full training set at their probability for that display.

Conclusions and Discussion

The results of this research may be important to Outbrain and other advertising concerns. The success of this project could help to improve the ability of advertisers to reach their audiences. One of the biggest successes of our project was the visualization of verification results. This interactive display was extremely helpful when testing different models and model attributes.

Though the ad score system did well by itself up to a point, the margin of return when using the system on sliced datasets was not very high. Moving forward, a couple of approaches may be tried with this system to improve predictive performance. As mentioned before, these ad scores could be used as input for other models. Ad score could be used as feature in a Poisson regression model. During our exploration of the data, we found that the order the ads appeared in played a role in their performance. Chi-square tests for independence provided evidence that the distribution of ad-clicks by ad position are significantly different from random chance for all display sizes ($p < .0001$) except for displays with 11 ads ($p = .1024$). Generally, the ads performed better the further left they were in the display. Though this phenomenon was verifiable, it did not a large enough effect to be used with the ad score system alone. However, question order may also be useful as a feature in the Poisson regression model.

Though our time to report improvement has ended, the Kaggle competition continues through January 18, 2017. We are currently ranked 107 out of almost 600 teams on Kaggle and have been ranked as high as 77th. We plan to continue working on the predictive ability of our models and achieve a higher rank.

All members contributed equally to the project. In fact, due to illness early in the semester, Gregory Parry joined the project just before the project update was due. However, his contributions from that point forward were above and beyond expectations. He developed the

random selection algorithm, took the lead on our poster, and contributed to the visualization. Joseph Phillips took the lead on the ad score algorithm and validating the models. Kyle Zimmerman took the lead creating the interactive visualization using D3. Kaustubh Mohite was in charge of the K-modes clustering. Arjun Mishra performed exploratory data analyses and visualizations using R, and played a large role in the feature generation for model building. Shoili Pal built the xgboost and neural net models.

Appendix

Figure 1. Ad Frequency Distribution

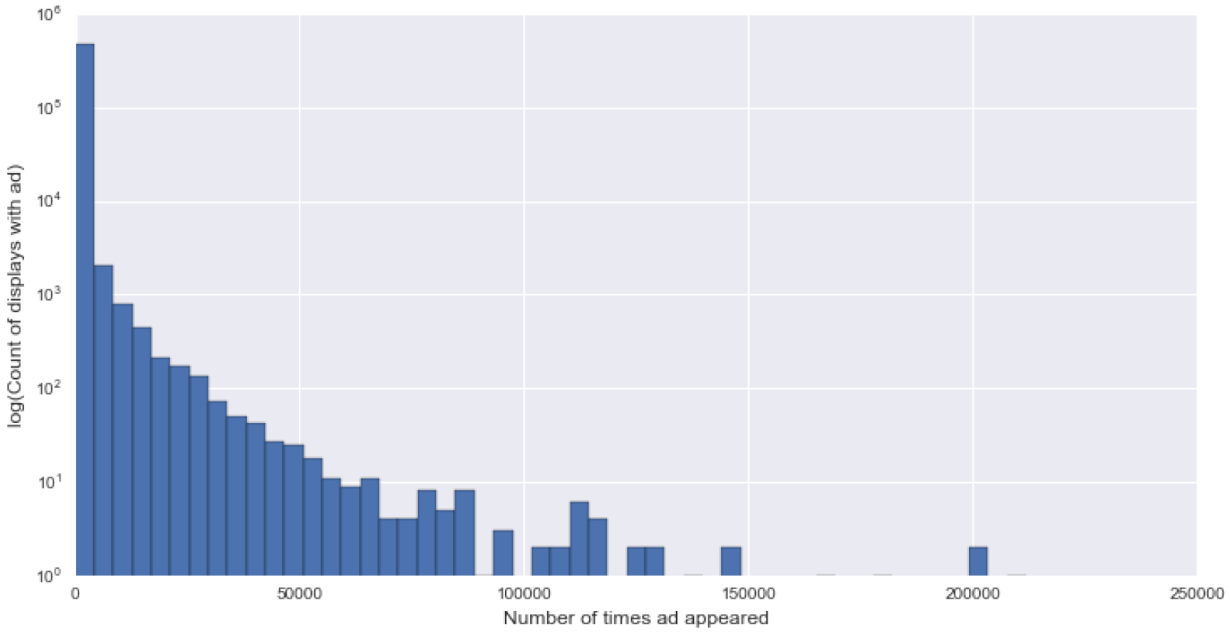


Figure 2. Platform Frequency Distribution

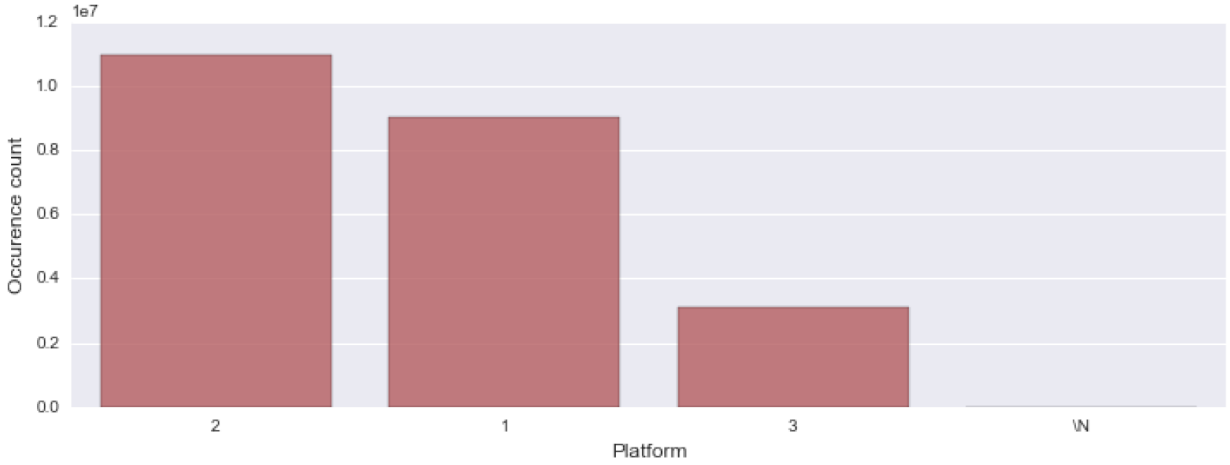


Figure 3. Top Advertisers by Country

Figure 4. Platform by Hour for July 15, 2016

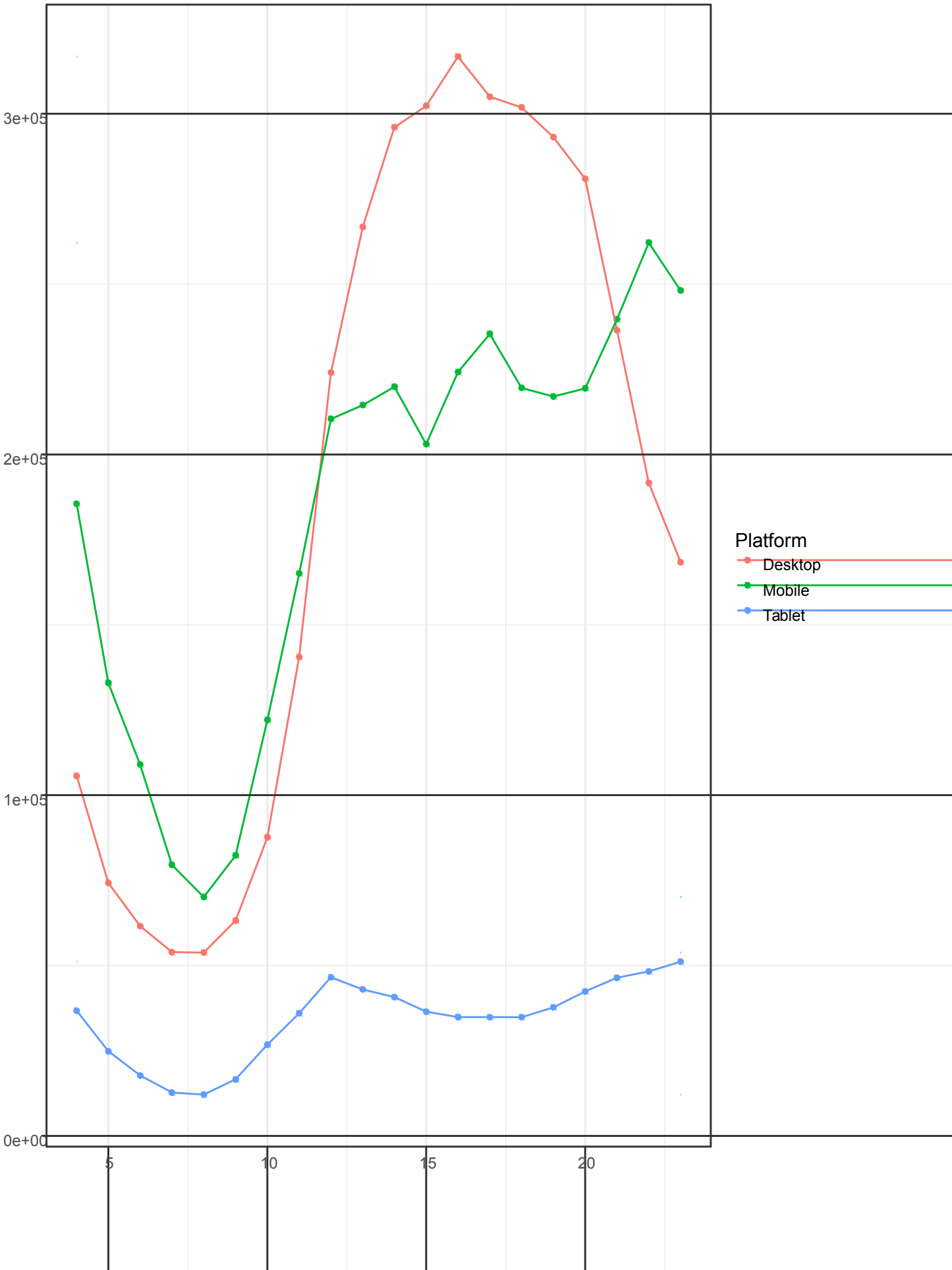


Figure 5. Ad Score Validation System as Used to Compare Grouped Data

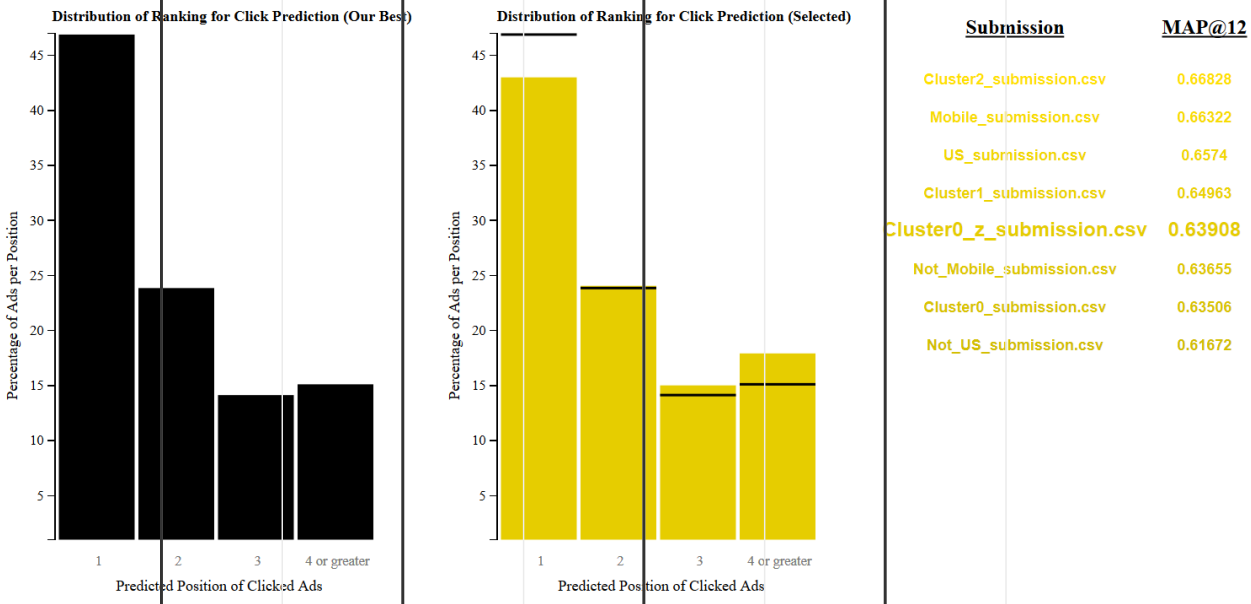
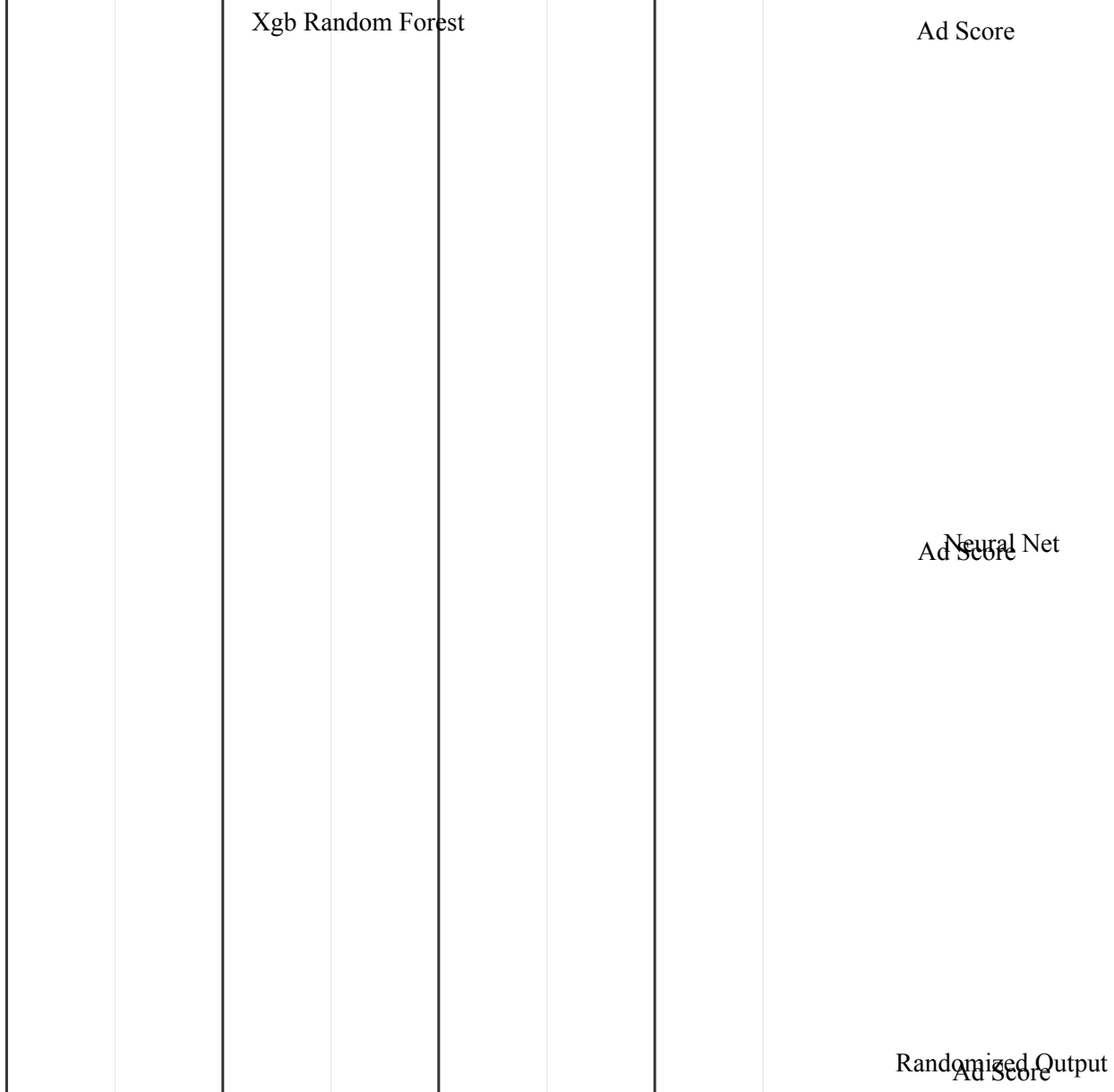
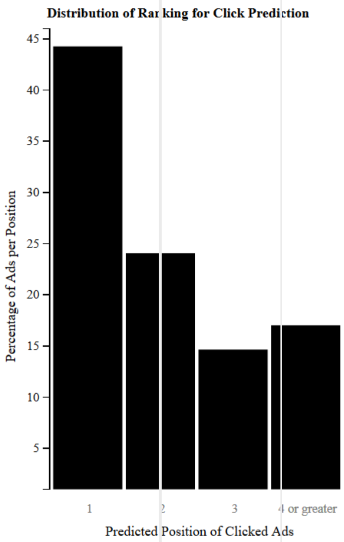
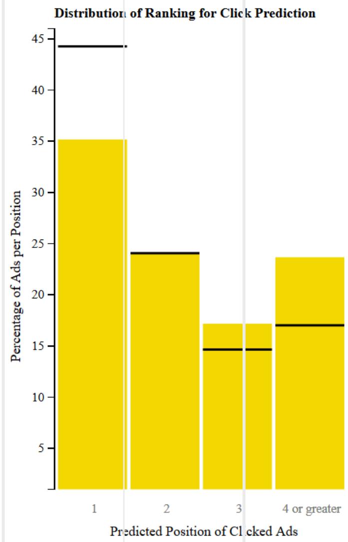
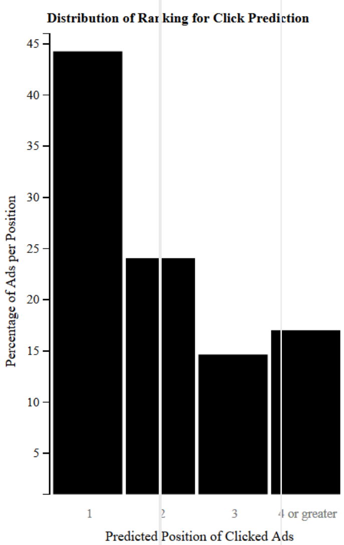


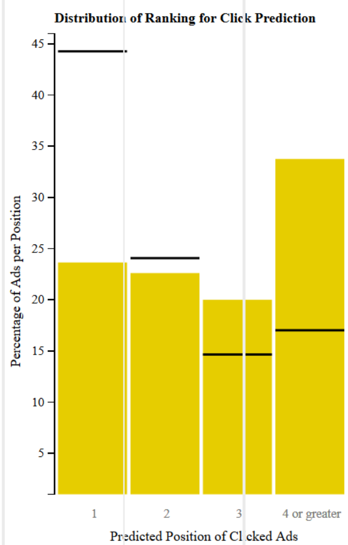
Figure 6. Performance of the Different Models

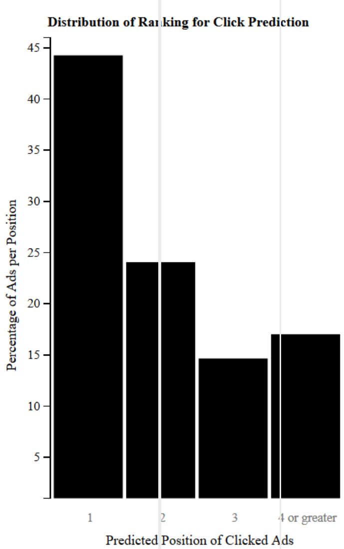


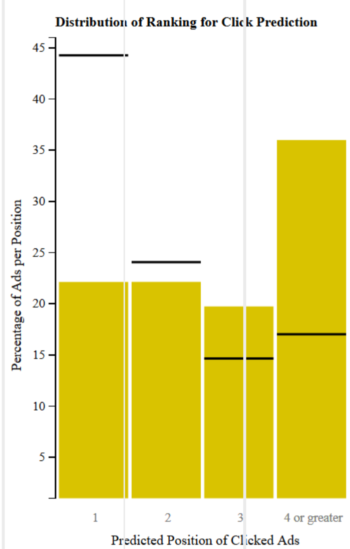

Distribution of Ranking for Click Prediction
Percentage of Ads per Position
45
40
35
30
25
20
15
10
5
1
2
3
4 or greater
Predicted Position of Clicked Ads


Distribution of Ranking for Click Prediction
Percentage of Ads per Position
45
40
35
30
25
20
15
10
5
1
2
3
4 or greater
Predicted Position of Clicked Ads


Distribution of Ranking for Click Prediction
Percentage of Ads per Position
45
40
35
30
25
20
15
10
5
1
2
3
4 or greater
Predicted Position of Clicked Ads


Distribution of Ranking for Click Prediction
Percentage of Ads per Position
45
40
35
30
25
20
15
10
5
1
2
3
4 or greater
Predicted Position of Clicked Ads


Distribution of Ranking for Click Prediction
Percentage of Ads per Position
45
40
35
30
25
20
15
10
5
1
2
3
4 or greater
Predicted Position of Clicked Ads


Distribution of Ranking for Click Prediction
Percentage of Ads per Position
45
40
35
30
25
20
15
10
5
1
2
3
4 or greater
Predicted Position of Clicked Ads

References

Attenberg, J , Suel, T., & Pandey, S. (2009). Modeling and predicting user behavior in sponsored search. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1067–1075.

Beales, H. (2010). The value of behavioral targeting. Retrieved from: http://networkadvertising.org/pdfs/Beales NAI Study.pdf

Besbes, O., Gur, Y., & Zeevi, A. (2016). Optimization in online content recommendation services: Beyond click-through rates. *Manufacturing & Service Operations Management, 18*, 15–33. doi:10.1287/msom.2015.0548

Baqapuri, A., & Trofimov, I. (2014). Using Neural Networks for Click Prediction of Sponsored Search.

Cartwright, W., Crampton, J., Gartner, G., Miller, S., Mitchell, K., Siekierska, E., & Wood, J. (2001). Geospatial information visualization user interface issues. *Cartography and Geographic Information Science, 28(1)*, 45–60.

Goswami, S., & Shishodia, M. S. (2013). A fuzzy based approach to text mining and document clustering. *International Journal of Data Mining & Knowledge Management Process, 3*, 43–52.

Huang, Z. (1998). Extensions to the k-means for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery, 2*, 283– 304.

Juan, Y., Zhuang, Y., Chin, W., & Lin, C. (2016). Field-aware factorization machines for CTR prediction. *RecSys '16–Proceedings of the 10th ACM Conference on Recommender Systems*, 43–50.

Kisilevich, S., Mansmann, F., Nanni, M., & Rinzivillo, S. (2009). Spatio-temporal clustering (pp. 855–874). Springer US.

Liu, Y., Xue, Y., Xu, D., Cen, R., Zhang, M., Ma, S., & Ru, L. (2012). Constructing a reliable Web graph with information on browsing behavior. *Decision Support Systems, 54*, 390–401. doi:10.1016/j.dss.2012.06.001

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., . . . Kubica, J. (2013). Ad click prediction: a view from the trenches. *KDD'13– Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining,* 1222–1230.

Nath, S. X., Ravindranath, L., Padhye, J., & Lin, F. (2013). SmartAds: Bringing contextual ads to mobile apps. *MobiSys 2013 - Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 111–123.

Ravi, K. P., Leng, A. G. K., & Singh, A. K. (2013). Application of Markov chain in the PageRank algorithm. *Pertanika Journal of Science and Technology, 21*, 541–554.

Sarukkai, R. (2000). Link prediction and path analysis using Markov chains. *Computer Networks, 33*, 377–386.

Wang, C. J., & Chen, H. H. (2011). Learning user behaviors for advertisements click prediction. *34rd international ACM SIGIR conference on Research and development in information retrieval Workshop on Internet Advertising*.

Wang, T., Bian, J., Liu, S., Zhang, Y., & Liu, T. (2013). Psychological advertising: Exploring user psychology for click prediction in sponsored search. *KDD'13– Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 563–571.

Unver, L., & Gundem, T. I. (2016). Authentication of uncertain data based on k-means clustering. *Turkish Journal of Electrical Engineering & Computer Sciences, 24*, 2910–2928.

Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., . . . Liu, T. (2014). Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks.