Click Through Rate Prediction

Team Members:

Po-Han Yen, Shih-Siang Lin, Hsin-Yu Tsai, Yun-Jung Fan, Shu-Ping Chen

ORIGINAL WORK STATEMENT

We the undersigned certify that the actual composition of this proposal was done by us and is original work.

	Typed Name	Signature
Contact Author	Po-Han Yen	*Po-Han Yen*
	Shih-Siang Lin	*Shih-Siang Lin*
	Hsin-Yu Tsai	*Hsin-Yu Tsai*
	Yun-Jung Fan	*Yun-Jung Fan*
	Shu-Ping Chen	*Shu-Ping Chen*

I. Executive Summary

Online advertisements have a significant influence on the success of a business. Effective advertisements can help businesses establish long-term relationships with customers and target the right potential customers, resulting in repeat sales and a high conversion rate. Therefore, our project is aimed to forecast the click-through rate (CTR) for evaluating ad performance and identifying potential users. The process comprises four stages: gathering data, pre-processing, training and testing the model, and evaluation. We use the data present on Kaggle and then convert the raw data by partitioning, recategorizing, etc. After that, we use the cleaned data to train predictive models. Finally, experimental results reveal that our best performance model produces an accuracy of 65.2% and is good at identifying both clicked and non-clicked ads. Time is the most influential factor, followed by the types of device model, website and app in predicting click-through rate.

II. Data Description

1. Data source: Kaggle CTR prediction contest
   (https://www.kaggle.com/competitions/avazu-ctr-prediction)
2. Data information: 11 days worth of Avazu data
3. Sample size: 45006431 observations
4. Variables: (23 in total)
   (1) Dependent variable
      click (0/1 for non-click/click)
   (2) Independent variables
      Numerical: hour (format is YYMMDDHH, so 14091123 means 23:00 on Sept. 11, 2014 UTC)
      Categorical: id, banner_pos, site_id, site_domain, site_category, app_id, app_domain, app_category, device_id, device_ip, device_model, device_type, device_conn_type, C1, C14 – C21 (Anonymized categorical variable; due to the private issue for the company)

id	click	hour	C1	banner_pos	site_id	site_domain	site_category	app_id	app_domain	app_category	device_id	device_ip	device_mode	device_type	device_conn_	C14	C15	C16	C17	C18	C19	C20	C21
1E+18	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	ddd2926e	44956a24	1	2	15706	320	50	1722	0	35	-1	79
1E+19	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	96809ac8	711ee120	1	0	15704	320	50	1722	0	35	100084	79
1E+19	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	b3cf8def	8a4875bd	1	0	15704	320	50	1722	0	35	100084	79
1.0001E+19	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	e8275b8f	6332421a	1	0	15706	320	50	1722	0	35	100084	79
1.0001E+19	0	14102100	1005	1	fe8cc448	9166c161	0569f928	ecad2386	7801e8d9	07d7df22	a99f214a	9644d0bf	779d90c2	1	0	18993	320	50	2161	0	35	-1	157
1.0001E+19	0	14102100	1005	0	d6137915	bb1ef334	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	05241af0	8a4875bd	1	0	16920	320	50	1899	0	431	100077	117
1.0001E+19	0	14102100	1005	0	8fda644b	25d4cfcd	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	b264c159	be6db1d7	1	0	20362	320	50	2333	0	39	-1	157
1.0001E+19	0	14102100	1005	1	e151e245	7e091613	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	e6f67278	be74e6fe	1	0	20632	320	50	2374	3	39	-1	23
1.0001E+19	1	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	37e8da74	5db079b5	1	2	15707	320	50	1722	0	35	-1	79
1.0001E+19	0	14102100	1002	0	84c7ba46	c4e18dd6	50e219e0	ecad2386	7801e8d9	07d7df22	c357dbff	f1ac7184	373ecbe6	0	0	21689	320	50	2496	3	167	100191	23
1.0002E+19	0	14102100	1005	1	e151e245	7e091613	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	5d877109	8f5c9827	1	0	17747	320	50	1974	2	39	100019	33
1.0002E+19	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	6f407810	1f0bc64f	1	0	15701	320	50	1722	0	35	-1	79
1.0002E+19	0	14102100	1005	0	9e8cf15d	0d3cb7be	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	58811cdf	8326c04b	1	2	20596	320	50	2161	0	35	100148	157
1.0002E+19	0	14102100	1005	0	d6137915	bb1ef334	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	72aab6df	4258293	1	0	19771	320	50	2227	0	687	100077	48
1.0003E+19	0	14102100	1005	0	85f751fd	c4e18dd6	50e219e0	98fed791	d9b5648e	0f2161f8	a99f214a	6-Dec-96	aad45b01	1	0	20984	320	50	2371	0	551	-1	46
1.0004E+19	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	a4f47b2e	8a4875bd	1	0	15699	320	50	1722	0	35	100084	79
1.0004E+19	0	14102100	1005	0	d9750ee7	98572c79	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	9b1fe278	128f4ba1	1	0	17914	320	50	2043	2	39	-1	32
1.0004E+19	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	c26c53cf	be87996b	1	2	15708	320	50	1722	0	35	100084	79
1.0004E+19	0	14102100	1005	1	0c2fe9d6	27e3c518	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	b7a69808	158e4944	1	0	6558	320	50	571	2	39	-1	32
1.0004E+19	0	14102100	1005	0	85f751fd	c4e18dd6	50e219e0	66a5f0f3	d9b5648e	cef3e649	a99f214a	fa60af6b	b4b19c97	1	0	21234	320	50	2434	3	163	100088	61

5. Interesting points of the data

(1) Big data:

This dataset includes over 45 million observations, making the analysis process computationally expensive and much more complicated.

(2) Anonymized features:

Due to the personal information protection policy of the company, some features are anonymized. As a result, understanding and aggregating these features is a difficult and challenging task.

(3) Imbalanced classes of the dependent variable:

For the dependent variable, "click", most of the records are class 0; hence the classes of the dependent variable are unbalanced. As we do the modeling and evaluation, we have to take into account this special situation because it may greatly affect the predictive results.

III.   Research Questions

1. In Internet marketing, CTR stands for click-through rate: a metric that measures the number of clicks advertisers receive on their ads per number of impressions, which is tremendously important since it directly affects both your Quality Score and how much you pay every time someone clicks your search ad. (See the scatter plot between CTR and Quality Score in the appendix). High Quality Scores allow you to improve or maintain ad position for lower costs, and achieving a high click-through rate means that you are driving the highest possible number of people to your offering. Therefore, in order to help business increase the advertising revenue, our research aims to predict the

click through rate given the relative features including some anonymous predictors. Is it possible to predict whether an ad would be clicked by viewers or not based on historical data? Furthermore, is it possible to successfully identify both clicked ads and non-clicked ads?

2. Online advertisements are viewed by an enormous amount of people every day and thus the amount of data is accumulated at an extremely fast speed. However, the value of the data decreases as time proceeds. In order to derive the most value out of the data, companies must be able to handle big data using limited resources during limited time. Therefore, our research attempts to deal with issues with big data. Is it possible to drive values from the data in an efficient way under constraints?

IV. Methodology

Stage I

Tackle with three major issues:

1. Anonymous features contain a lot of categories and thus fitting them into the model would suffer from the curse of dimensionality. However, trying to reduce dimensionality using regular feature engineering methods based on domain knowledge is unsuitable for our case since there is no way to figure out the meaning of anonymous features. Therefore, we map categories into five smaller groups based on their mean CTR, compared to the overall mean CTR. This as expected requires expensive computational efforts and thus later on we have to sample many subsets of data and process them one by one at a time.

2. Unbalanced data would hinder the model's ability to successfully classify the minority class and hence we consider three major practices to handle this issue: Assign weights to each class, synthetic minority class and downsample majority class. Assign weights is an easier approach but at the same time is model dependent. Synthetic minority class ensures full data but increases the computational time and effort. Downsampling makes the data easier to handle but comes with a tradeoff of losing part of the data. Considering the data

size and the diversity of our models, we decide to downsample the data to get relatively balanced classes.

3. 45 millions of data is beyond the handling capability of our local devices. R studio can not load this amount of data and Python can not process the whole dataset at once. Not to mention fitting the model with full data. As a result, we decide to sample the data (125 samples) and build numerous weak predictors based on each subset of data (100 thousand observations). After that, we ensemble all the models to get a strong classifier.

Stage II

Make possible of our ensemble approach:

1. Test different algorithms on a small subset of data to decide which ones to use for our ensemble model. Tested algorithms include logistic regression, naive bayes, KNN, classification tree, bagging, random forest, gradient boosting, XGB. Decide to use logistic regression, naive bayes and classification tree as our base models and build both homogeneous & heterogeneous ensemble algorithms as our final models for later testing. KNN is not chosen because the computational requirement at scale is too large. Random forest is preferred for our developed ensemble model because of the relatively good performance on unbalanced test data.

2. Develop our own programs to automate the preprocessing, sampling and modeling procedures, which require extremely expensive human power and are not realistic to finish without automated machines.

3. Develop our own ensemble algorithm that randomly chooses 2 to 6 features as predictors on base models to de-correlate the models, detect the types of input models to apply appropriate prediction methods and average the predicted results of individual models to generate final predictions.

4. Test our ensemble algorithms and conclude that the performance of the balanced random forest which ensembles 375 classification trees is the best and should be used for the CTR prediction.

V.   Results and Findings

The goal of our research is not only to increase the prediction accuracy of click-through rate but also the sensitivity and specificity. The reason is that clicked ads only consist of 17% total ads and if we were to simply predict all ads to be non-clicked, we would have an accuracy of 83% classifier but perform poorly when identifying clicked ads. As a result, we want a relatively good model in successfully classifying both classes. So that to help companies better allocate their advertising budgets to accurately deliver different categories of ads to different customers with different preferences.

In the first stage, we compare the performance of traditional classification models and ensemble methods. As shown in the table below, ensemble methods like XGB, Bagging & Random Forest generally have a more balanced performance (similarity among three matrices) than traditional models like NaiveBayes & Classification Tree. As a result, we choose ensemble methods to proceed to the next stage. KNN has a balanced performance but considering the computational power it needs and the amount of data we are dealing with, we do not include KNN to our ensemble model.

Stage I	Accuracy	Sensitivity	Specificity
NaiveBayes	0.637	0.271	0.894
KNN	0.636	0.609	0.641
Prune Tree	0.608	0.267	0.907
Bagging	0.591	0.666	0.575
Gradient Boosting	0.562	0.216	0.863
XGB	0.635	0.402	0.684
Random Forest	0.627	0.775	0.595

In the second stage, we compare the performance of our self-build homogeneous and heterogeneous ensemble algorithms. As shown in the table below, both homogeneous ensembles and heterogeneous ensembles have quite balanced performance but tree based ensemble models (higher accuracy) perform better than NaiveBayes or Logistic Regression ensemble models (lower accuracy). Interestingly, the results correspond to our first stage test where random forest has one of the best performance.

Stage II	Accuracy	Sensitivity	Specificity
125 Tree	0.651	0.638	0.653
125 NB	0.598	0.706	0.575
125 LR	0.609	0.695	0.59
125 Tree + 125 NB	0.623	0.666	0.613
125 Tree + 125 LR	0.64	0.666	0.634
125 NB + 125 LR	0.604	0.683	0.587
125 Tree + 125 NB + 125 LR	0.628	0.666	0.619
375 Tree	0.652	0.638	0.655

After using our self-build ensemble algorithm, we are able to not only increase the prediction accuracy but also balance the performance of our model. We have reached an accuracy of 65.2%, sensitivity of 63.8% and specificity of 65.5% from ensembling 375 classification tree models. Our model can successfully identify 63.8% clicked ads and 65.5% non-clicked ads.

VI.   Conclusion

CTR plays a crucial role in measuring the impact of companies' advertisements and has become an effective method of precisely targeting customers. Therefore, CTR prediction will be a hot topic in the coming decades. Our project implements CTR prediction by developing and evaluating different models, including Bagging, Random Forest, Gradient Boosting, XGB, Naive Bayes, Classification Tree, KNN and self-build ensemble models. Results show that our self-build ensemble tree based model outperformed all the other models with an accuracy of 65.2% and less than 2% variability among three matrices.

These results are quite significant considering the computational and time limitations of our research, we have gradually increased the prediction accuracy and dramatically decreased the variability among different performance metrics. Moreover, since we have already built the entire automated modeling system, we can simply sample more training data and include more models in our ensemble algorithm to further boost the performance

in the future. Other methods to boost the performance include implementing deep learning algorithms or processing the data on big data platforms like Hadoop and Spark.

VII.   Appendix
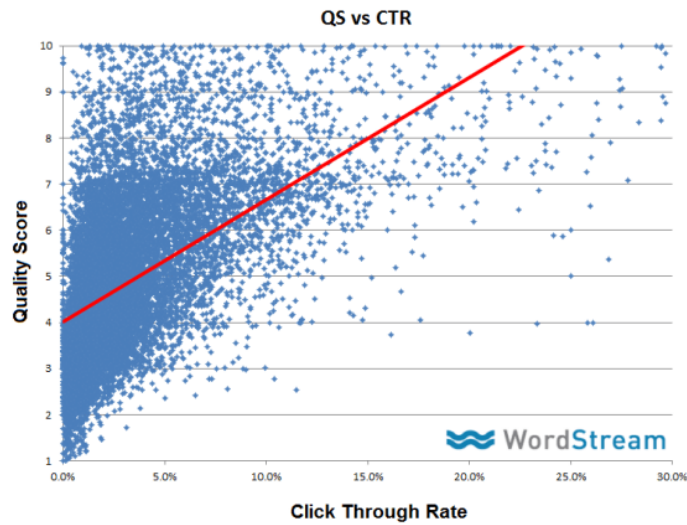
Scatterplot showing the relationship between Quality Score and Click Through Rate