Facebook Ad Performance Prediction System (December 2019)

TAM. WC Edwin (e0267707@u.nus.edu), *MTech ISS, National University of Singapore*,
LIN. JL (e0267606@u.nus.edu), *MTech ISS, National University of Singapore*

Abstract—This paper describes how an ads performance prediction solution developed for digital marketers can help them make informed decision before launching an image ad for any advertising campaigns running on Facebook Ads Platform. To answer this question, we developed a system that predict cost-per-click (CPC) of ad (image + text) and compare the performance of 10 actual campaigns with the prediction from the system using a mix of feature extraction and engineering techniques together with text processing and modelling techniques that utilizes python coding and KNIME software. Our results are twofold:

1. R^2 of .493 (which is1.8X better than our previous iteration)
2. The model is suitable for use by Marketing teams 65% of the time (within CPC error of +/- $0.60)

These results are meaningful for ads with CPCs lower than $12. Predicted higher CPC results tend to be under predicted as such ads are less frequent in our dataset. However, the performance may be further improved when there are more high CPC post-campaign data to continuously refine the model. From a proof of concept perspective, this paper shed some light on how marketers could "game" the Facebook Ads Platform to get the most out of their marketing spends.

Index Terms— Advertising, Data Mining, Evaluation and Performance, Marketing, Social Media.

I. INTRODUCTION

SOCIAL media advertising spending is increasing year over year as predicted by marketing leaders in the recent CMO survey [1]. Businesses today is aware of the importance of social media judging by the large amount of marketing budget invested in this channel [1]. However, digital advertising is costly and is not always as effective as what the marketers wanted it to be.

Social media ads publisher such as Facebook, generate a significant amount of revenue by showing a combination of images, textual or videos ads to users on their website and mobile application. They get paid every time when a desired action is performed by the target users due to the ads they've viewed or interacted with such as clicking an image or playing a video. There are also different cost models available such as cost-per-click (CPC), cost per action (CPA), cost-per-thousand-impressions (CPM) etc. In helping advertisers to make decision on how much to spend for their ads, such platforms typically have some built-in feature that provides general prediction on the performance to the advertisers as a rough estimate. However, such predictions and the actual data points and algorithms are usually not transparent to the advertisers besides knowing that it's based largely on the selected target audience segments and historical data provided by the platform itself. Other ad performance data is also available but only provided after the ad has run [2], which is usually too late, too costly.

.

Chen et. al. (2016) [3] mentioned that Features that are used to represent an ad are extremely important in a machine learning model and to make the CTR prediction model more accurate, many researchers use millions of features to describe a user's response record which is call an ad impression. CTR prediction is defined to estimate the ratio of clicks to impressions of advertisements that will be displayed and In the CPC model, the clickthrough rate (CTR) is an important indicator to measure the effectiveness of advertising display [4].

Visual features describe the visual appearance of an image ad at different levels and may have the power to influence the CTR of an image ad [3] and as result, affecting the CPC as well. The importance of visual features is also usually under estimated. Despite the abundance of ad performance data, marketers only have access to simple reports and they tend to rely on their gut feel and past experience to predict the “goodness” of an ad.

In this paper, we collaborate with Construct Digital, our project sponsor, on building a proof of concept prediction model for predicting CPC of an ad, based on the ad images, ad body text and ad targeting parameters. The prediction allows marketers to make informed decision even before the ad is live in the Facebook ad platform. We found that using XGBoost (Extreme Gradient Boosting) Linear Ensemble regression model works best for our set of data and features selected. As expected, the most important thing is to have the right features and indeed, the bulk of our time is spent on feature engineering and extraction.

II. DATASETS

Our datasets are provided by Construct Digital via their internal tool which extract post campaign data from Facebook Ad platform. The dataset used for training our models consist of 3,347 observations. Each observation has 41 metrics and 150 dimensions which contains key information such as the ad images, ad text, dates of campaigns and all ad targeting parameters. Out of all the observations, a total of 2,061 observations are being used eventually as we excluded non-English data because our main focus is on English ad only with ad images that are larger than thumbnail sized images. Although only 2,061 ads make the cut, we have a far larger image library that can be used during image object recognition phase.

III. SCOPE

Our system takes a social media advertisement (image and text) as an input and output the predicted performance to the marketer. Performance in our context is defined as Cost-Per-Click (CPC), which is the amount the advertisers need to pay for each click. Taking into account that different industries would have different performance benchmark.

A. *Extract data from Social Media Platform*

Ads and ad performance data is contained on social media platforms. They are our data source for this system. Accomplished using an internal tool used by Construct Digital Analytics team.

B. *Feature Extraction and Transformation*

The system should “recognize" image elements and text in the ad. Image elements could be people, car, etc. These elements are then used – in conjunction with ad Targeting Parameters – to train the system and predict performance on new ads. Accomplished using Python.

Ad parameters such as interest, work titles and countries are parsed and 1-hot encoded. Object recognition in images are achieved via ImageAI [5], a python library which supports object detection, video detection and object tracking using RetinaNet, YOLOv3 and TinyYOLOv3 trained on COCO dataset. Lastly, text Mining of ad body copy is performed using frequentist approach using single tokens with stop words removal. Calculates Term Frequency–Inverse Document Frequency (TF-IDF) statistic for use in model.

C. *Predict a close enough Cost-Per-Click (CPC)*

The final output of this system will be a predicted CPC. The model’s performance is based on R2, RMSE, and coverage within CPC tolerance of +/- $0.60 (e.g. if the system predicts a CPC of $1 for an ad; the actual answer should be within $0.40 to $1.60).

IV. Benefits

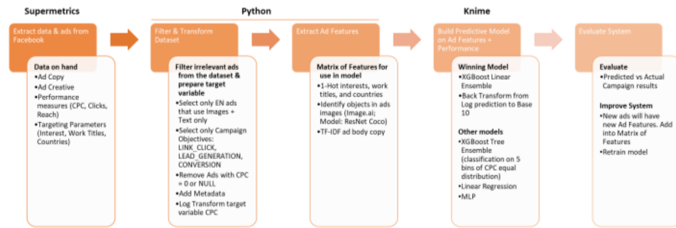
As highlighted that it is often costly to advertise on social media platform such as Facebook. Hence, a tool like this will help advertisers reduce wastage of ads and running cost.

Other existing social media platforms currently in the market that sell ads inventory besides Facebook are LinkedIn, Twitter, Google and other more. These platforms will not tell advertisers the future performance of an ad despite the abundance of ad performance data. Marketers only have access to simple reports on these platforms and metrics such as audience Reach. Hence marketers tend to rely on their “gut” and past experience to predict the “goodness” of an ad.

Ad Creative feature mining is still relatively experimental. There are efforts to correlate ads and ad parameters to ad performance [5] [6] [4]. But intuitively, the ad has the largest effect on ad performance. The closest tools are likely Programmatic Advertising platforms (e.g. DataXu) or Creative Management Platforms (CMP e.g. Thunder). However, these platforms are primarily used to run ad campaigns and any intelligence is dependent on the platform. It is also relatively expensive and may costs anywhere between SGD 5,000 to 100,000 per month to run a campaign (costs are based on regions, audience segments, and number of creatives).

V. Solution Outline

Our solution consists of extraction and transformation of data. Ad features are extracted , transformed before using it to build the predictive model for evaluation. The solution is built using a combination of the following three sub-systems.



Solution Flowchart (Zoom in to see detail)

A. Facebook Ad Data Extraction

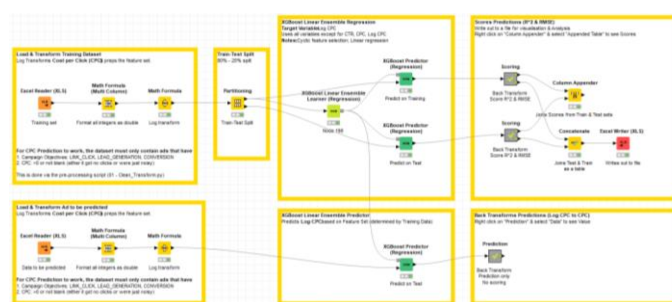
This is an internal tool build by Construct Digital using a commercial product, Supermetric. We have no affiliation with Supermetric and the data we worked on are already extracted by the analytics team in Construct digital. Hence, there is no need for us to recreate a new ad data extraction tool.

B. Python Code to Clean and Transform Data

Python codes for cleaning and transforming the data are developed as part of this solution. All our code uses free, opensource libraries that are available in the internet under the Creative Commons license.

C. KNIME Software to Build Predictive Models

For quicker model building and testing, we used KNIME, an open source analytics platform which has most of the machine learning algorithms and allows us to write custom python codes when required, as our main modelling tool besides python.


KNIME Data Model Workflow (Zoom in to see detail)

VI. Methods

There is no need for meta-controls as the System infers from an input. It doesn’t have to handle conflicting examples. The hook in to Facebook Ad

Manager will only happen later after the models are trained and prepped. Hence the remaining components requiring algorithmic or inferential mechanisms are in Basic Feature Transformation, Image extraction, Ad Text processing, and Prediction Model.

A. *Basic Feature Transformation*

We used 1-hot encoding on Targeting parameters and inputs such as dictionary of interest, work titles, and countries. Using the inputs to create a 1-hot matrix of inputs against each ad.

B. *Image Extraction*

Extract image features based on category labels to be inputted. The inputs are images of fixed size (1,200 X 628 px; padded if necessary). We passed this image through a Convolutional Network. The architecture is based on 17 convolution layers. The first convolution layer uses 5 X 5 convolution kernels. Following first layer, there are four groups, and each has four layers with 3 X 3 kernels.

The network is pre-trained on a training dataset of category labels and the output are feature vector of the image. We would like to do this in an unsupervised learning way in the future as we don't want to limit the things in the ads

C. *Image Object Recognition*

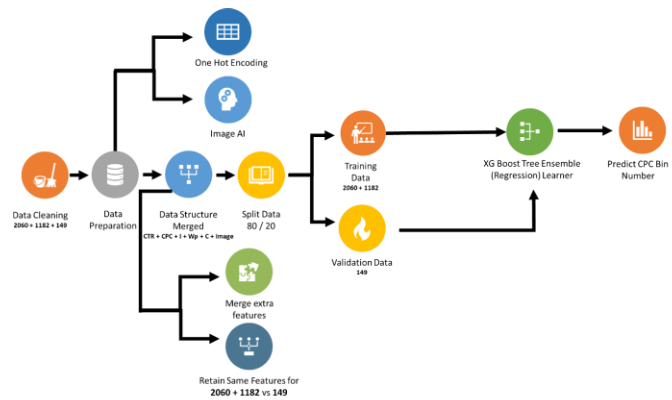
The objects appearing in an ad is detected using ImageAI, an opensource python library built to empower developers, researchers and students to build applications and systems with self-contained Deep Learning and Computer Vision capabilities using simple and few lines of code. The library can detect up to 80 common objects which is sufficient for us to validate the feasibility of this solution.

D. *Ad Text Processing*

Frequentist method using TF-IDF. Input required are the Ad body copy. We tokenize all copy, remove stop words, and find the TF-IDF for each token and the outputs is a probability matrix of terms against each ad.

E. *Prediction Model*

The label/target to work toward is CPC and its Log transformed to reduce spread. The inputs are the feature vector of image, TF-IDF of ad text mined, basic feature vector of interests, work title, countries and model are train on: Log Transformed CPC by passing all features into a XGBoosted linear ensemble (regression) and the output is a Log CPC (Back transformed by 10^CPC)



Process Flow

VII. CHALLENGES

During out initial run of our first few models, we encountered high feature dimensionality & sparsity after extraction. We realized that we had 510 Interest Features; 203 Work Positions; and 52 Image Features to process. As there are possible contextual overlaps (e.g. Basketball and tennis are ball sports), we tried to reduce dimensionality within each Feature Class to get better results – but to no avail.

We also started hypothesizing whether individual feature classes (e.g. Interests) do well in predicting ad performance. In addition to building the mixed input model, we used linear regression on individual feature classes against ad performance. As a side effect, we used R^2 to determine the amount of dimensionality reduction to do on feature sets.

Our models aren't very accurate despite having tried individual linear regression models and mixed input models. The best feature combination only answers 0.2465 of variance.

The interim results were not promising. As a result, we explored the following methods and techniques.

A. *Changing features*

We increased dataset size. Originally, we had 2,000+ workable ads, by including this year's ads – it increased to 3,000+ workable ads. Next, we focused on ads with the follow objectives: LINK_CLICKS, LEAD_GENERATION, CONVERSIONS. This removes noise caused by other types of ads as Facebook optimizes to set objectives (e.g. REACH).

Thirdly, we removed rows which had CPC values = 0 or were blank. This implies that there was no budget set against the ad, or they performed so badly that they didn't get a single click – which meant that we can't calculate CPCs.

Fourthly, we included ad body text as a feature set. Its processing has been described earlier.

Lastly, as CPC distribution is right skewed with most of the CPCs falling between $0.1 to $4 (90%). Hence, we log transformed CPCs to reduce spread, and expect the model to do badly when predicting high CPCs (i.e. greater than $12)

B. *Simplifying our model*

We also start to simplify our models and trial different approaches. Instead of using mixed input models, we stuck with basic models that were boosted or ensembled. In addition, we believed that a classification model (since the spread was so skewed) would be more meaningful.

Then we categorized CPCs into equally distributed bins for classification prediction which is done alongside regression predictions.

Eventually simplifying the model and started to trial on ensemble and boosted models - XGBoost (Extreme Gradient Boost) Linear Ensemble (regression) and XGBoost Tree Ensembles for regression and classification respectively.

VIII. RESULTS

We trialed 2: classification or boosted linear ensembles. The classifier gave 57% accuracy on the categorized CPCs (5 bins of equal CPC distribution); while linear ensembles provided an R^2 of .493 on the test dataset. The latter represents a 1.8X increase in performance over the best during our initial phase.

Appended table - 0:218 - Column Appender (Joins Scores from Train & Test...)

File Hilite Navigation View

Table "Scores" - Rows: 6 | Spec - Columns: 2 | Properties | Flow Variables

Row ID	CPC_Pr...	CPC_Pr...
R^2	0.648	0.493
mean absolute error	0.819	1.152
mean squared error	4.309	9.382
root mean squared er...	2.076	3.063
mean signed difference	-0.296	-0.531
mean absolute perce...	0.462	0.618

Results of XGBoost Linear Ensemble (Regression)

Our R^2 isn't great (just better than initial models). However, during our test and validation phase, the marketers gave feedback that they can tolerate some inaccuracy in the predictions. Currently, this threshold is set at +/- $0.60.

Upon graphing the predicted vs actual CPCs, we find that the model is suitable for use 65% of the time. In addition, it also highlights that the increase in RMSE is likely due to the lack of data for larger CPCs. This result is encouraging – particularly in light of our initial results.

CPCs and imprecision Thresholds (Regression)

There is general agreement amongst the Marketing team that this proof of concept has value in bringing rigor to their media planning activities and as a product for client use.

We can imagine that the eventual system will be part of the Construct Digital's Marketing Intelligence suite of products. To get there, we estimate that it would take about 1 year to develop and trial the components in the recommendations. Primary investments would go to building a robust data pipeline to extract, store and transform ad data from different ad platforms (preferably cloud based); getting alignment and test usage across internal stakeholders; and mitigating data access restrictions by ad platforms.

This could probably be done by a 2-man team of data scientists supported by an ever-changing roster of interns.

IX. Validation

We took 10 marketing campaigns spread out in 2019. From these campaigns, we extracted the ads and targeting parameters (interests, work titles, countries). These parameters were chucked into the system, and predicted CPCs for each campaign and then verify against the actual CPC.

As the system is not user friendly, it is helmed and used by the analytics department. However, intel was provided to marketing department for the test. Results from the 10 ads are shown below:

Campaign ID	Actual CPC	Predicted CPC	Difference (Actual – Predicted)
112	23.51	11.98257018	-11.5274
118	12.33	12.37465873	0.044659
119	13.51	6.964873139	-6.54513
133	8.95	5.308738358	-3.64126
134	5.66	5.869095793	0.209096
3409	0.14	0.254500537	0.114501
3410	0.22	0.245758997	0.025759
3411	0.19	0.214330371	0.02433
3412	0.3	0.26010577	-0.03989
3413	0.25	0.214330371	-0.03567

This test dataset has the following statistics:

1. RMSE: 4.34
2. R^2: 0.84
3. About 70% of this test's results is within +/- $0.60 threshold

The test actually outperforms the model's R^2 while doing worse at RMSE. And as expected, campaigns that actually get high CPCs woefully under-predicted. While the rest of the campaigns did OK.

X. Findings

This appears to be a feature engineering problem. The bulk of time went into extraction and transformation of features from Interests, Work Titles, Countries, Objects in Ad Image, TF-IDF for Ad Text. We had 5,145 features, all of which we needed. As a comparison, the current version (with text data) had better performance than earlier versions.

As most of the features are categorical (e.g. either Interest A was set as a parameter or not), we needed to 1-hot encode them for use in the models.

Interestingly enough, there was no need to reduce feature dimensionality (we did so earlier via feature hashing and PCA, but the results were not ideal). Instead – using One Hot Encoding forces the model to take into account each individual feature's impact thus generalizing the model and improving its RMSE & R^2 values.

Campaign objectives do matter as Facebook optimizes against these set objectives. When predicting on CPCs, we only used ads with the objective of LINK_CLICK, LEAD_GENERATION, and CONVERSION. The other objectives are simply noise.

CPC distribution and skew matters. CPC is heavily skewed towards the right, with the majority of performance data falling between 0.01 to $4. Thus, the model works quite well in predicting values up to $12. But after that the model becomes unreliable.

The performance of this predictor system is at R^2 (0.49) & RMSE ($3.063). However, on speaking with the marketing team, they were actually OK if the system was able to predict CPCs within an offset of +/- $0.60. As such the model works about 65% of the time.

XI. Recommendations

The biggest performance boost came from text mining the ads. That seems to be a feature engineering area that we could develop further. In addition, the marketing team has highlighted several improvements a follow:

1. Predict on Clicks instead of CPC. Use estimated Reach from Facebook as a variable

2. Make the distinction between text used in ad headlines over body copy.

3. Refine the text mining approach. Use phrases or co-occurrences instead of just tokens

4. Partition the dataset by campaign objectives and run the model accordingly. We expect to see quite different results.

5. Include a simulation module such that end users can input parameters without help

As our dataset grows larger, we expect that the current model will miss out on objects in images and rare targeting parameters – after all, the current training dataset only uses a subset of Facebook's available options. Hence the model will need to be retrained regularly. This should be done in such a way that it does not involve any human intervention.

XII. Future enhancements

The enhancement and changes that we believe will make the system more robust is highlighted below.

1. A real time data streaming function could be used to replace existing data extraction processing. This new function will keep the models up to date by listening to changes to campaign performance and refreshed the data in the data lake in real time.

2. Current local storage to be replaced by cloud storage where there can be enough processing power and storage for the new data and also for the use of custom image training to complement the pretrained models.

3. Ad benching marking tool will incorporate a user interface that is also linked to Facebook Ads platform via API for immediate activation of campaign directly without the need to login into Facebook ads platform separately.

XIII. Conclusion

Opportunity Cost of Ad Biasedness is present since our system is based on our available dataset. Of which, we have close to 3,000 rows of preprocessed data to use. However, this is based on work done for Construct Digital's clients, which encompasses certain creative and industrial biases (e.g. enterprise technology ads tend to be rational; retail ads are more "fun").

In addition, the CPC distribution of the dataset is skewed towards lower values. The system is trained on these ads. Hence, we expect the system to mis-predict novel ads or ads that are not well represented for certain industries. This problem can be mitigated over time and ads.

While we do not expect to compete with the ad platforms, which no doubt makes use of far greater volumes of features and ads to build far more accurate prediction models, this project has certainly shed some light on how marketers can perhaps "game" Facebook to get the most out of their marketing dollars.

The challenges we faced in this project – trialing of models, with many features to crunch – simply highlights that takes great effort and imagination to prepare the dataset for use. More importantly, simplicity rules over mixed models. It certainly drives home the message that features matter, and 80% of our time is spent on data wrangling and munging! Regardless, when taken as whole, the project is a step ahead for the sponsor's ambitions to create a suite of marketing intelligence tools.

XIV. References

[1] P. CHRISTINE MOORMAN, "Why Social Media Performance Lags Even As Spending Soars," [Online]. Available: https://cmosurvey.org/2019/07/why-social-media-performance-lags-even-as-spending-soars/. [Accessed 18 12 2019].

[2] Facebook, "About Estimated Daily Results," Facebook, [Online]. Available: https://www.facebook.com/business/help/1438142206453359?id=561906377587030. [Accessed 18 Dec 2019].

[3] J. Chen, B. Sun, H. Li, H. Lu and X.-S. Hua, "Deep CTR Prediction in Display Advertising," in *the 2016 ACM*, 2016.

[4] Q. Wang, F. Liu, S. Xing and X. Zhao, "A New Approach for Advertising CTR Prediction Based on Deep," *Computational and Mathematical Methods in Medicine*, p. 11, 2018.

[5] Lihui Shi and Bo Li, "Predict the Click-Through Rate and Average Cost Per Click for Keywords Using Machine Learning Methodologies," in *2016 International Conference on Industrial Engineering and Operations Management*, Detroit, Michigan, USA, Sept 2016.

[6] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers and Joaquin Quiñonero Candela, "Practical Lessons from Predicting Clicks on Ads at Facebook," in *ADKDD'14*, New York, NY, USA, August 24 - 27 2014.

[7] Xiaoliang Ling, Weiwei Deng, Chen Gu, Hucheng Zhou, Cui Li and Feng Sun, "Model Ensemble for Click Prediction in Bing Search Ads," in *2017 International World Wide Web Conference Committee (IW3C2)*, April 2017.

[8] Lin Guo, Hui Ye, Wenbo Su, Henhuan Liu, Kai Sun and Hang Xiang, "Visualizing and Understanding Deep Neural Networks in CTR Prediction," *Proceedings of ACM SIGIR Workshop on eCommerce*, vol. SIGIR 2018 eCom, p. 7, 2018.