

FACEBOOK AD PERFORMANCE PREDICTION SYSTEM

PROJECT REPORT

BY

EDWIN TAM WEI CHING - A0178396J

LIN JUN LIANG - A0178295M

THE INSTITUTE OF SYSTEMS SCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE

MASTER OF TECHNOLOGY IN KNOWLEDGE ENGINEERING



THE INSTITUTE OF SYSTEM SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

January 2020

Distribution List

Strictly for NUS ISS and Construct Digital only

Acknowledgements

We would like to express my deepest appreciation to all those who provided us the possibility and opportunity to complete this project and report. A special gratitude we give to our final year project supervisor, Dr Zhu Fang Ming and Lecturer, Mr. Sam Gu, whose contribution in stimulating suggestions and encouragement, helped us explore other methods to move forward the project and guiding the team in achieving the goal.

Furthermore, we would also like to acknowledge with much appreciation the crucial role of our project Sponsor, Construct Digital, Edwin & his team, who gave the permission to use all required equipment and the necessary materials to complete the project.

I have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.

Table of Contents

Distribution List	2
Acknowledgements	3
Executive Summary	6
1) Problem Definition	8
1.1) Problem Description	8
1.2) Project Scope & Objectives	9
1.3) Benefits & Cost	10
System Costs	11
1.4) Solution Outline	12
2.1) Knowledge / Data Acquisition	12
2.2) Knowledge / Data Modelling	14
3.0) System Design	16
3.1) Operational Context	17
3.2) Functional Description	18
3.3) User Interface Design	19
3.4) Knowledge / Data Structure Representation	19
3.5) Problem Solving Paradigm	20
3.6) Technical Architecture and Design	22
Analytics	24
3.7) Hardware and Software	28
4.0) Implementation and Results	28
4.1) Implementation	28
Phase 3 Results are Interesting	30
4.2) Results / Validation and Verification	34
Findings & Results Summary of System	35
5) Recommendations	36
5.1) Project Management	36
5.2) Results	37
5.3) Conclusion	37
6) References	38
7) Appendices	38
Project Proposal & Reports	38

Glossary 38

AdSet targeting Definition: 38

Executive Summary

Social media marketing is an important part of marketing for businesses today. As such, Facebook ads platform is one of the top social media platforms where businesses run advertisements on to promote their products and services.

However, it cost money to run ads on Facebook platform and very often, budget for running the ads are limited and marketers would want to achieve the best revenue on ads spend (ROAS) for their marketing campaigns. In most campaigns higher click throughs usually results in better performance, henceforth, it is the interest of the marketer to want to be able to gauge the performance of the ads before launching the ads into the platform.

Hence, the goal of the final system is to help the marketers to predict the Cost-Per-Click (CPC) of an ad, before the ad goes live in Facebook's ads platform.

The intelligent system we developed takes a social media advertisement (image and text) to predict its performance. To do so, the system is trained based on extracted data from Facebook ads platform and go through a series of features extraction, transformation, image objects recognition and natural language processing steps to prepare the data for modelling.

Despite the abundance of ad performance data, marketers only have access to simple reports on the ads platforms currently in the market and marketers tend to rely on their gut feel and past experience to predict the "goodness" of an ad.

The closest tools are likely Programmatic Advertising platforms or Creative Management Platforms (CMP e.g. Thunder). However, these platforms are primarily used to run ad campaigns. Any intelligence is dependent on the features provided by the individual platform and they are relatively expensive.

The majority of the cost will be hosting and processing the data which is discussed in detail in the sections below. As the current solutions run on a local environment, cost is minimal, but we do foresee the storage size would increase further as more data is available when there are more campaigns running and the data would be used to train and further improve the model's performance.

The bulk of time went into extraction and transformation of features from Interests, Work Titles, Countries, Objects in Ad Image, TFIDF for Ad Text and the biggest performance boost came from text mining the ads which seems to be a feature engineering area that could be develop further in future.

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.

1) Problem Definition

It costs money to run advertisements on social media, in our case here, Facebook platform. But budgets are finite – this limits the display frequency of social media ads. Thus clients, with limited monies and limited showtimes, want the most bang for their buck. In this case, it's **click-throughs**: the more people who **click** on their ad and land on their website, the happier they will be. Otherwise, its good money tossed away – resulting in unhappy clients.



The goal of the final system is to help the marketers to predict the Cost-Per-Click (CPC) of an ad, before the ad goes live in Facebook's ads platform. The predicted CPC will provide marketers a good indicator on ad performance as well as the "goodness" of their media plans.

Thus, the goal is to build a system that can extract features from Facebook Ads, transform them (if necessary), and then predict the CPCs for each ad.

1.1) Problem Description

What gets clicks on a social media advertisement?

Two similar ads for the same product
Similar styling, target audience; yet the performance is different

<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"><div style="display: flex; justify-content: space-between; align-items: center;"><div>SAP (ID, PH, TH, ...)</div><div>...</div></div><div style="margin-top: 5px;"><small>Sponsored · 🌐</small></div><p>Find out why Huber's Butchery considers SAP Business One a perfect match to its expanding business. Download your free SAP Business One trial now.</p><div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"><div>A Solution That Grows With Your Business</div><div>Sign Up</div></div><div style="margin-top: 5px;"><small>https://www.sea-sap.com/mobile-app-...</small></div></div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"><div>CTR: 1.2% CPC: \$0.45</div><div style="text-align: center;"><div style="color: green; font-size: 2em;">↑</div><div style="color: green; font-weight: bold;">1.9X 2X</div></div></div>	<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"><div style="display: flex; justify-content: space-between; align-items: center;"><div>SAP (ID, PH, TH, ...)</div><div>...</div></div><div style="margin-top: 5px;"><small>Sponsored · 🌐</small></div><p>You can now manage your business anytime, anywhere. Get the free SAP Business One Demo and join others who've made the right choice.</p><div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"><div>Manage Your Business On the Go</div><div>Sign Up</div></div><div style="margin-top: 5px;"><small>https://www.sea-sap.com/mobile-app-...</small></div></div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"><div>CTR: 0.62% CPC: \$0.93</div><div></div></div>
---	--

CTR: Click-Through Rate
CPC: Cost Per Click

Today, if you ask any marketer, you won't get the same two answers. It could be the audience, the copy, the image, the timing and cadence, or...any combination of these factors and a toss of a polyhedron dice. As such, social media marketing teams depend on their gut and past experiences when planning new social media ad campaigns on Facebook, LinkedIn, or

elsewhere. More importantly, Marketers can't find out the performance until the advertisement runs for at least 1 to 2 weeks.

If performance is great, money is well spent. If not, it's wasted – as the team pivots their advertising tactics – and a client is left unfulfilled.

With the abundance of ad performance data available in Facebook, we can use machine learning methods to “guess” the performance of the ads before campaigns go live. This gives us an indication of the goodness of an advertisement and whether it needs to change before it goes live.

Hence this Facebook Ad Performance Prediction System.

1.2) Project Scope & Objectives

We want to develop an intelligent system that takes a social media advertisement (image and/or text) to predict its performance. Performance is defined as Cost-Per-Click (CPC). Different industries would have different performance benchmark.

In a nutshell, this system should achieve these:

1. Extract data from Social Media Platforms

Ads and ad performance data is contained on social media platforms. They are our data source for this system. Accomplished using Supermetrics.

2. Feature Extraction & 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.

IS Problems:

- a. Label Encoding of ad parameters
Parsing and 1-hot encoding of interests, work titles, and countries
- b. Object recognition in images
Achieved via ImageAI which supports object detection, video detection and object tracking using RetinaNet, YOLOv3 and TinyYOLOv3 trained on COCO dataset
- c. Text Mining of ad body copy
Frequentist approach using single tokens with stop words removal. Calculates Term Frequency-Inverse Document Frequency (TF-IDF) statistic for use in model

3. 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 R^2 , 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).

IS Problem Type: Supervised learning. Linear Regression on Log Transformed CPC
Machine learning with XGBoost Linear Ensemble

This sponsor's objective is to develop a proof of concept, and if results of the performance are shown to be viable, the system will be bookended with end user UI elements, trialed on ad data from other platforms, and integrated into the agency's marketing intelligence stack by sponsor's development and data science teams. For this iteration, the usual UI frills will be minimal.

Here's an example of how the system works.

It should "read" a social media ad as is from some place (e.g. a folder). User to upload ad parameters to the system. The system extracts ad features. It runs the ad parameters and ad features to present a range of likely performance.

For example: given a Facebook ad, this system should tell us that the ad should get a CPC of \$4.70 (range: \$4.10 to \$5.30). The results are compared to post-campaign results to determine the accuracy of the prediction.

1.3) Benefits & Cost

Is it worth building an intelligent ad prediction system? Are there tools that already do this?

Let's have a look:

1. Facebook, LinkedIn, Twitter will not tell you 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.

2. Ad Creative feature mining is still relatively experimental

There are efforts to correlate ads and ad parameters to ad performance (see [Sources](#)). But intuitively, the ad has the largest effect on ad performance

3. Cost-effective commercial platforms are the exception

The closest tools are likely Programmatic Advertising platforms (e.g. DataXu) or Creative

Management Platforms (CMP e.g. Thunder). However, these platforms are:

- a. Primarily used to run ad campaigns. Any intelligence is by-the-way and subject to the whims of the platform.
- b. Relatively expensive. It 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)

Short answer: Yes (it's worth building), and No (there doesn't seem to be tools).

System Costs

The majority of the cost will be hosting and processing the data, which we will explain below. There is also an opportunity cost associated with ad biasedness – after all, the model is dependent on the data used.

Hosting and Processing the Data

The current solutions run on a local environment. Storage used for now is around 4.1GB which consist of ad images (creatives) and post campaign performance data extracted from Facebook Ads Platform using Supermetric, a 3rd party Facebook ad report extraction tool used by Construct Digital internally to extract reporting data from Facebook.

We foresee the storage size would increase further as more data is available when there are more campaigns running and the data would be used to train and further improve the model's performance.

Opportunity Cost of Ad Biasedness

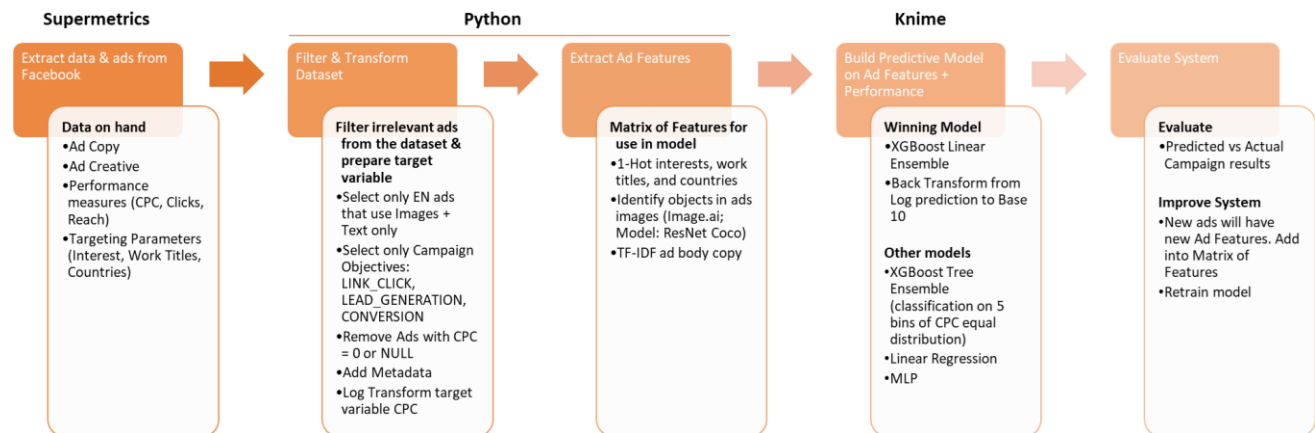
Our system is based on our available dataset. Of which, we have close to 3,000 rows of preprocessed data to use. However, this is based on work done for the agency's clients, which encompasses certain creative and industrial biases (e.g. enterprise technology ads tend to be rational; retail ads are more "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/cost can be mitigated over time and ads.

1.4) Solution Outline

As a reminder, the system will predict ads performance (CPC) given an ad image and ad text. It is trained on past ads and ad performance. This system helps marketers decide if they want to launch the Facebook campaigns as is or amend the ads before launch.

Solution Flowchart



The solution consists of extract + transform data; extract ad features; build predictive model; and evaluate/run. The solution is built using a combination of three sub-systems:

1. Supermetrics to extract Facebook Ad Data
2. Python to Clean & Transform the data
3. Knime to learn the model & predict CPCs for future ads

2.1) Knowledge / Data Acquisition

To understand usage and patterns, we did these things:

1. Interview stakeholders
2. Explore extracted data

Our stakeholder interviews and exploratory data analyses suggested two distinct types of knowledge models for this project:

1. What goes into making and launching a Facebook ad

Indicated by repeated expositions of “the ad must have a headline”, “we must also take care of targeting” etc.; as well as our exploratory data analysis of Facebook ad data.

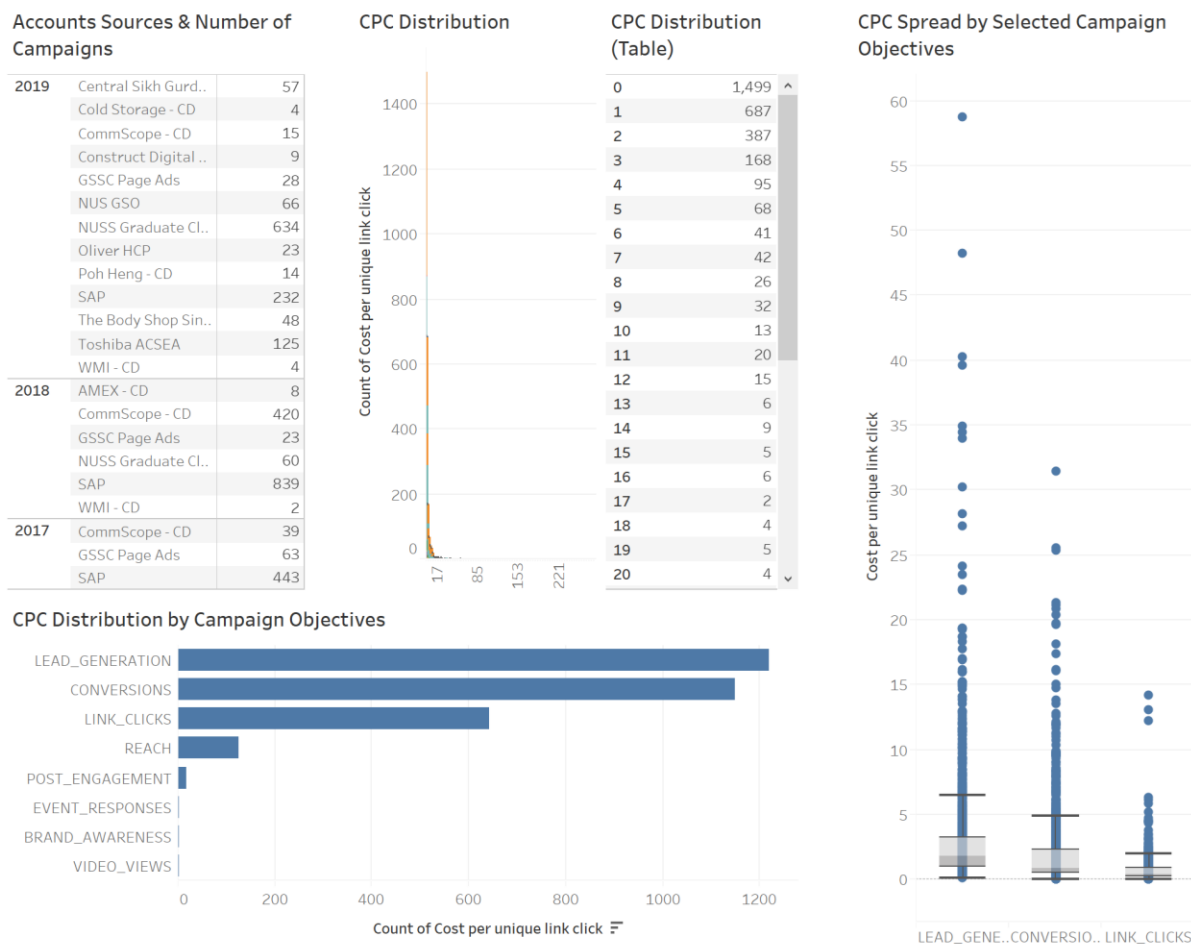
The relationships are largely “part-of”; hence we will model this knowledge as a composition tree.

2. Why an ad performs poorly (or well)

Not surprising considering that social media ad optimization is largely based on trial and error with some reverse-engineering taking place. In this case, the marketers are looking to hypothesize causes of ad performance. Hence, we will model this knowledge as a cause tree.

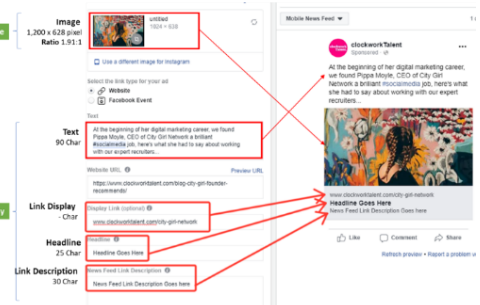
We expect to use knowledge model “1. What goes into making and launching a Facebook ad” more often as it defines the parts of an ad that we’ll use as inputs for our predictive model. See next section for both knowledge models.

Exploratory Data Analysis of Facebook Ads



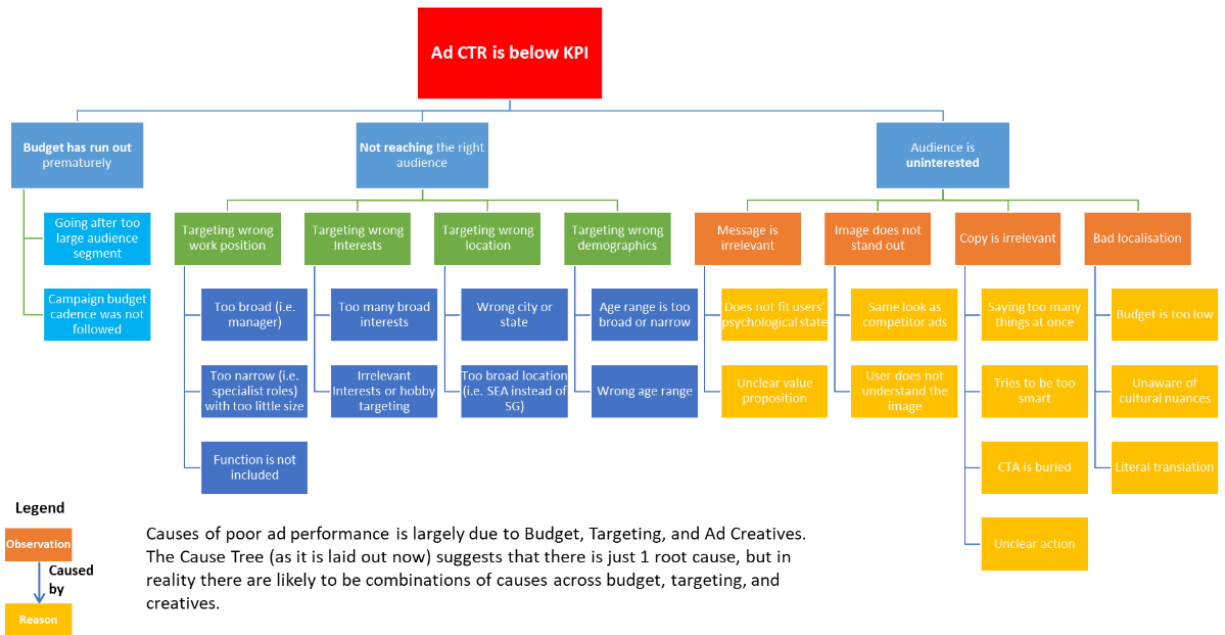
As we can see, the ads are mostly run for B2B type companies (e.g. SAP, Commscope). The CPCs are unequally distributed – both across Campaign Objectives and CPC values, and even within the most relevant campaign objectives.

This informs us to filter the dataset by objectives; and log transform CPC to “tighten” the distribution.



2. Why an ad performs poorly (or well)

Cause Tree



These knowledge models determined the key features that our target users (marketers and media planners) used in their Facebook campaign set up and planning. This helped to focus on designing a system that would fit into their daily workflow with minimal changes required if they were to use the system eventually. It also shows the interaction between the users and the system.

3.0) System Design

Operations

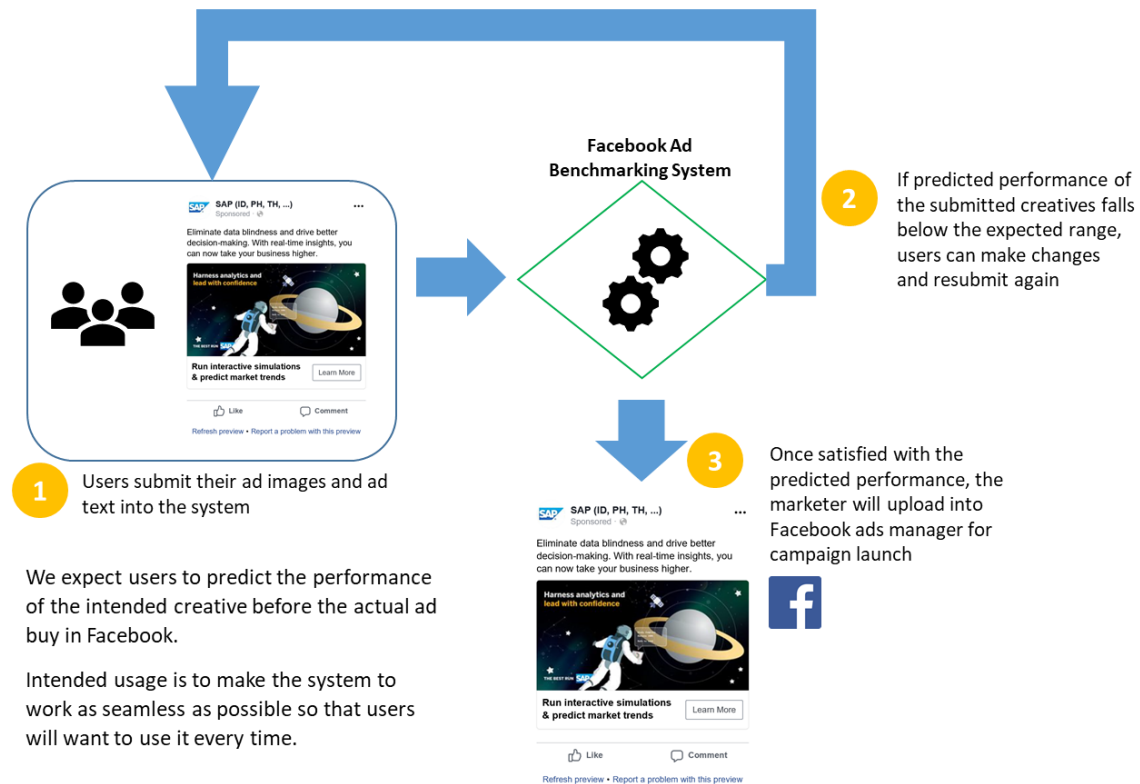
This system is seen as an experiment/prototype by the sponsor to fulfill a business problem. The IS techniques while new, are not cutting edge, hence the focus is on providing a tool to solve an impending need which are lacking right now.

User & System Roles

User	Role	Frequency of Use	Security/Access Features Used	Additional Notes
Digital media planners / in-house marketing executive	Plans & Executes ad campaigns for brand	At least 1 per day	<ol style="list-style-type: none">1. Upload Ads + parameters2. Get CPC prediction	Future versions: ad improvement pointers, current position in benchmarks, Predict on other ad performance metrics (e.g. CTR)
System	Advisor that helps users forecast ad performance	NA	<ol style="list-style-type: none">1. Access Facebook Ad manager2. Access historical database	Future versions: automatic link up with social media platforms; recommendation engine

Digital media planners / in-house marketers execute the media buys in Facebook by logging into the Facebook Ads Manager platform and upload the creatives ads, input the ad copy and select the placement and targeting criteria according to the media plan that was signed off by clients.

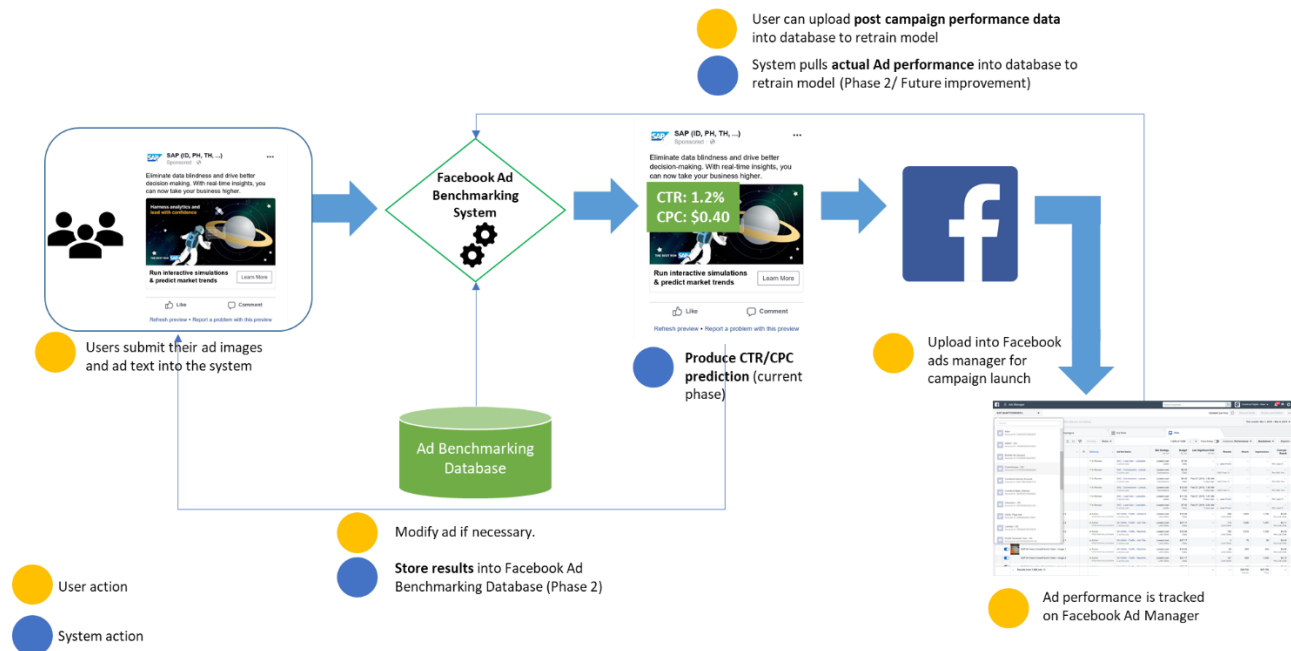
Users and System Interactions (Today's context)



3.1) Operational Context

Before using the system, it is advisable for users to have the creatives and ad copy ready to be uploaded to the system for prediction.

Users are encouraged to provide the campaign performance results back into the system for the purposes of validating the prediction and retraining the model once they have new data from the campaigns.



3.2) Functional Description

All systems comprise of functions that are interconnected via function calls to provide the expected functionalities for the users. We adopted a user-story format that helps to contextualize what the system should be capable of doing in different business scenarios.

1) “As a marketer, I can input my ads image so that I get to know the potential CTR and CPC for the ad image I provided”

System functions:

1. Allow the user to upload the image
2. Produce a prediction result based on the upload

2) “As a marketer, I can manually add on more images with actual campaign performance so that I can get better prediction from the system”

System functions:

1. Allow user to enter the location of the image and the actual performance
2. Extract features from new images provided by user
3. Use new features to re-train existing models
4. New models to be redeployed

3) “As a marketer, I want to bulk upload campaign performance data from Facebook ads manager so that I don’t have to manually do so.”

External function: The in-house system scrapes data from Facebook, which is not perform by our system, and upload the information into our cloud storage.

System functions:

1. Connect to cloud storage and locate new data daily
2. Extract features from new images provided by user
3. Use new features to re-train existing models
4. New models to be redeployed

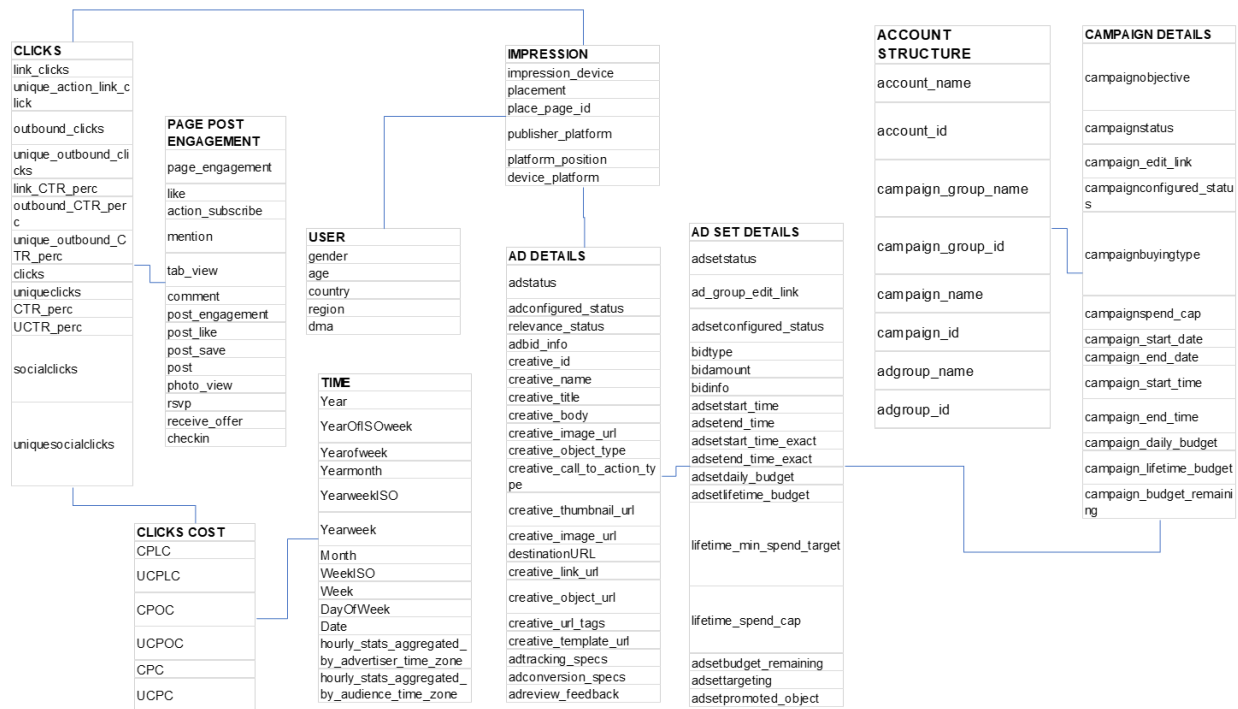
3.3) User Interface Design

User interface design is currently not applicable.

Our final solution is run using Python code and KNIME directly.

3.4) Knowledge / Data Structure Representation

This is how we’d represent an ad’s features if they were stored in a database.



3.5) Problem Solving Paradigm

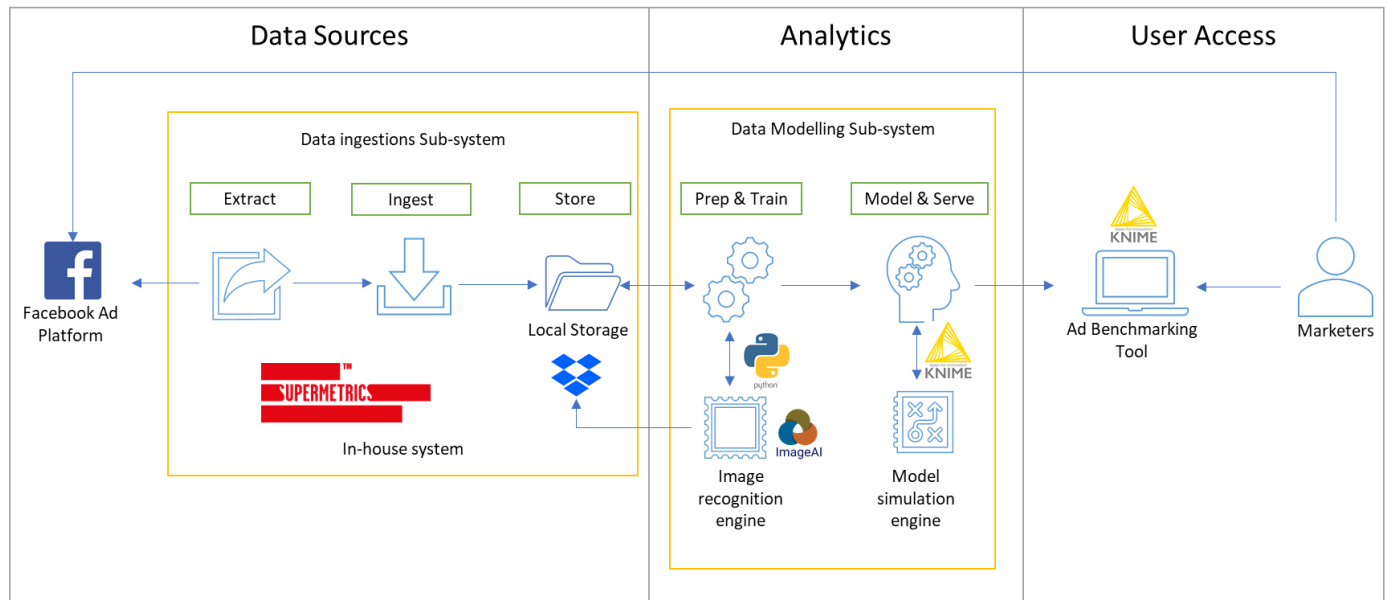
Let's go from the simplest to the most complex.

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.

Basic Feature Transformation NA 1-hot encoding on Targeting parameters	Image Extraction Supervised Learning Extract image features based on category labels to be inputted.
Input: Dictionary of interest, work titles, and countries Create a 1-hot matrix of inputs against each ad.	Input: image of fixed size (1,200 X 628 px; padded if necessary) <ol style="list-style-type: none"> 1. Pass 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. 2. The network is pre-trained on a training dataset of category labels Output: Feature vector of the image Note: 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
Ad Text Processing Frequentist using TF-IDF	Prediction Model Supervised Learning Label to work towards: CPC (Log transformed to reduce spread)
Input: Ad body copy Tokenize all copy, remove stop words, and find the TF-IDF for each token. Outputs a probability matrix of terms against each ad.	Input: Feature vector of image + TF-IDF of ad text mine + Basic feature vector of interests, work title, countries Train on: Log Transformed CPC Pass all features into a XGBoosted linear ensemble (regression). Output: Log CPC (Back transformed by 10^{CPC})

3.6) Technical Architecture and Design



High level view of the technical architecture of the overall system

Above shows an overview of the system architecture and the sub-systems, segmented by

3 key areas:

- 1) Data source
- 2) Analytics
- 3) User Access

5 sub areas:

- 1) Extract
- 2) Ingest
- 3) Store
- 4) Preparation & Training
- 5) Model and Train

2 sub-systems:

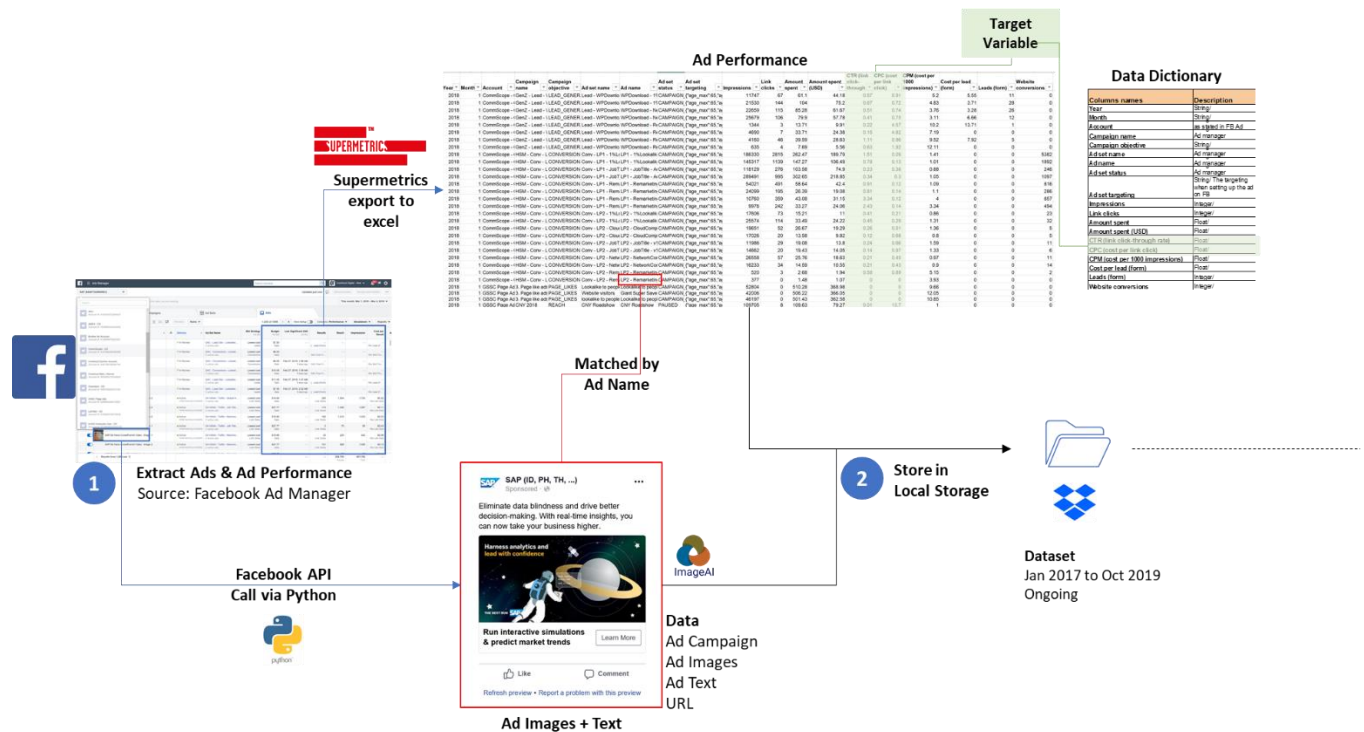
- 1) Data ingestions
- 2) Data modelling

Data Source

In this area, there is a data ingestion sub-system used by Construct Digital which bring together all structured, unstructured and semi-structured data from Facebook's ads platform and store it into a local Storage. Right now, the data is stored inside Dropbox's account of Construct Digital and are sync to our local file system.

The local storage in our computer is the interface between data ingestion and modelling sub-systems. When there are new data extracted from Facebook Ads platform, it will be sync across our local storage via Dropbox.

Data Ingestion Sub-system



Technical architecture of the data ingestion sub-system

The data ingestion sub-system is divided into 3 parts.

1) Facebook Campaign Data Extraction (out of scope)

Supermetrics: Uses data connectors that connects to Facebook's Ads API and customized web scraper to extract information not provided by Facebook's API. The data extraction and ingestion will not be part of the scope for now since it's managed in-house.

Data freshness is on an on-demand basis. Replication of the data for data modelling sub-system will be done as and when is required.

2) Image recognition / Object detection (in Scope)

Python: Using ImageAI library and its pretrained image recognition models and together with other pretrained models from Kaggle.com

3) Image objects mapping (In scope)

Python: The detected objects from each ad images are then transposed , cleaned and map back to the Facebook campaign performance dataset as part of the data cleaning process.

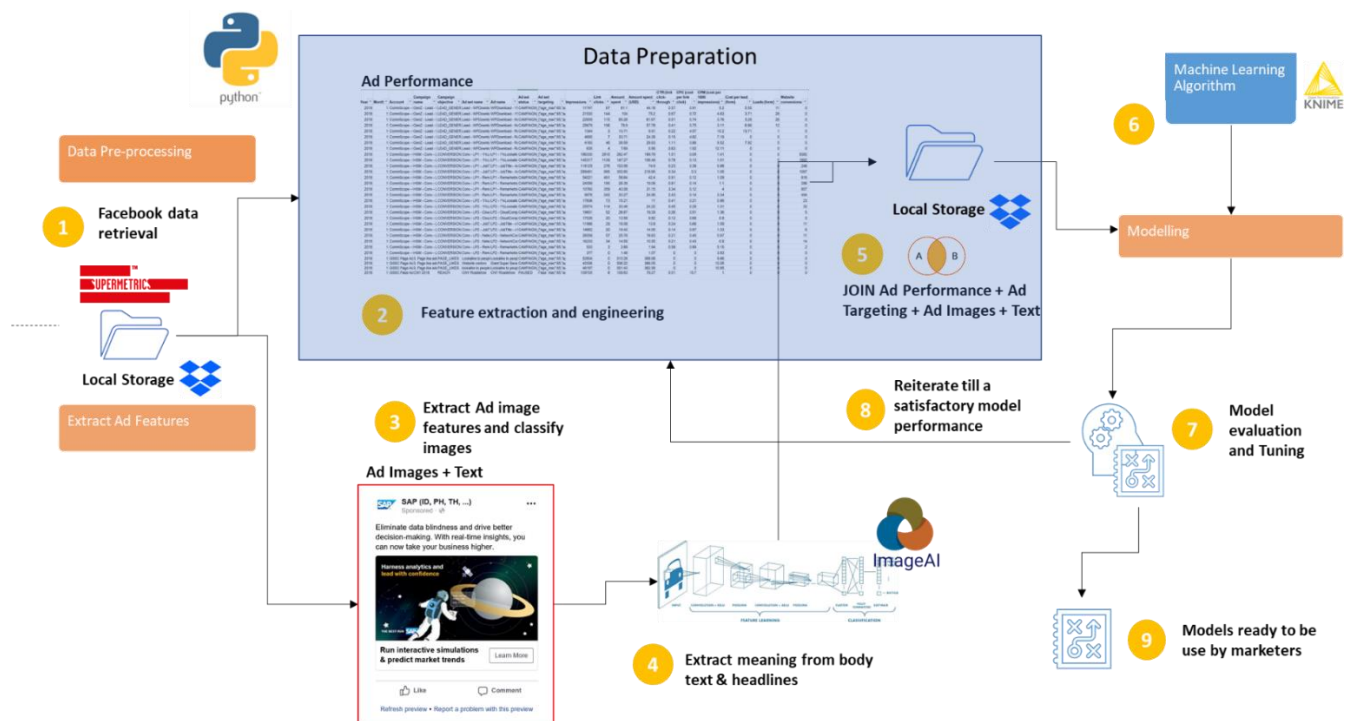
4) Ad Copy Text Mining (In scope)

Python: Tokenizes, removes stop words, and then calculates inverse term frequencies. They are mapped back to the Facebook Campaign performance dataset.

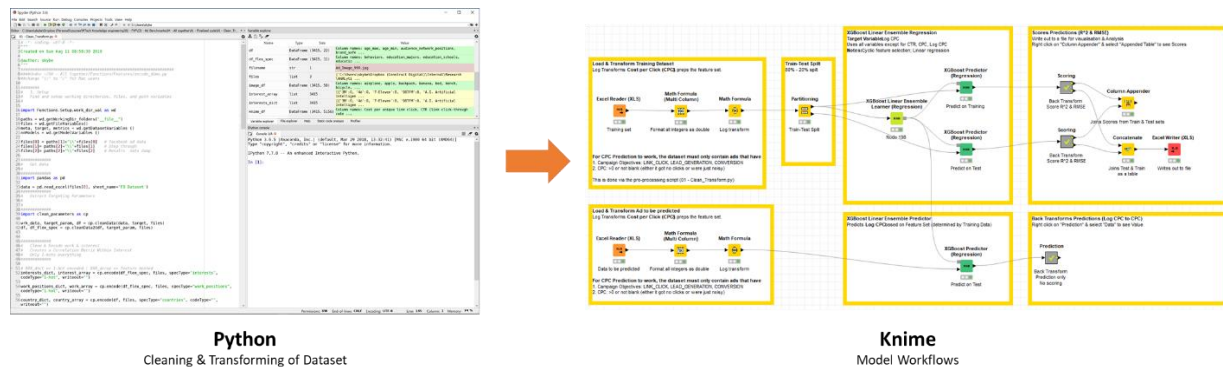
Analytics

The analytics area forms the data modelling sub-system, where computing power will be used extensively. We are aware this may be a bottleneck of the system when the data volume increases over time. As of development phase, time to extract features and training the models are less critical, but we are looking to scale the processing node into parallel processing environment to achieve faster prediction if the need arises.

Data Preparation Sub-system



Technical architecture of the data modelling sub-system



Cleaning & Modelling Sub-systems/Workflows

In the data preparation sub-system, data will be extracted from local storage and data cleaning and pre-processing happens here.

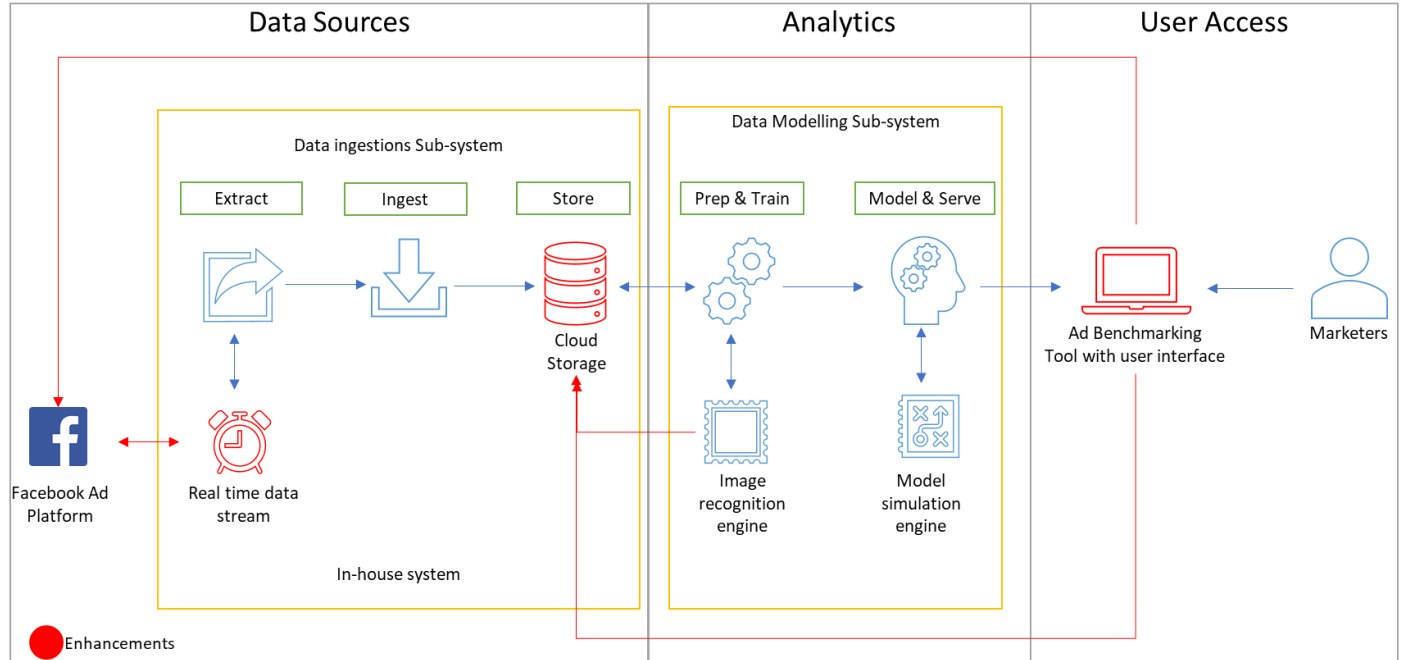
1. Data is extracted from Facebook ads platform using in-house tools by Sponsor
2. Perform feature engineering
 - a. Targeting parameters (Interest, work titles, countries)
 - b. Ads images recognition and mapping
 - c. Extracting text from the ad text and process using TF-IDF techniques
3. Combine processed features with ad performance data (i.e. matrix of features + ad performance). Split the dataset into 80% – 20% (training – validation dataset).
4. Prediction Model
 - a. Test with machine learning algorithms (once off to select a model that works)
 - b. Select model
 - c. Train model; Learn from dataset
5. Prepare for use by Marketers

For quicker mode building and testing, we have switch to the use of 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.

User Access

Lastly, this is the area whereby marketers interact with the system. To query the system with ads images and get predictions . As of now, we have set up KNIME in the marketer's PC, with some training, the tool helps to serve the needs, which is to get a prediction of the performance of the ads provided.

Future Enhancements to the System



High level view of the technical architecture of the overall system's future enhancements

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

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.

3.7) Hardware and Software

The system right now is run locally.

There really isn't a need to reinvent existing components. The sponsor wants to create a prototype to test their hypothesis – and more importantly find out what doesn't work.

4.0) Implementation and Results

The follow subsection will touch on what are the implementation we've tested and the results from our validation.

4.1) Implementation

During our initial implementation, we faced several challenges:

1. **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.

2. **Would 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.

3. **Our models aren't very accurate**

We have tried individual linear regression models and mixed input models. However, the accuracy rates are dismal as shown below. The best feature combination only answers 0.2465 of variance.

Model	CPC (Features)	Results
		CTR (Features)
Linear Regression (Interest)	0.1146 (48)	0.1463 (59)
Linear Regression (Work)	0.1318 (18)	0.0149 (28)
Linear Regression (Location)	0.0949 (36)	0.09492 (36)
Linear Regression (Images)	0.01499	0.01499
Linear Regression (Interest + Work)	0.1162 (59 + 18)	0.1457 (59 + 18)
Linear Regression (Work + Country)	0.2219 (28 + 36)	0.1071 (28 + 36)
Linear Regression (Interest + Country)	0.2605 (59 + 36)	0.1986 (59 + 36)
Linear Regression (Interest + Work + Country)	0.2465 (48 + 28 + 36)	0.2643 (59 + 18 + 36)
Mixed Input Model (see "Combine all feature categories")	0.01017 (ic1)	0.057944 (iwc1)

As outlined in Phase 2, the interim results were not promising. As a result, we explored the following methods and techniques:

a. Change our features.

1. Increase dataset size. Originally, we had 2,000+ workable ads, by including this year's ads – it increased to 3,000+ workable ads.
2. 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).
3. 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.
4. Include ad body text as a feature set.
5. 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. 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.

1. Categorized CPCs into equally distributed bins for classification prediction. Done alongside regression predictions.
2. Simplify models. Trial ensemble and boosted models - XGBoost Linear Ensemble (regression) and XGBoost Tree Ensembles for regression and classification respectively.

Phase 3 Results are Interesting

There were 2 approaches: classify or boosted linear ensembles. The classifier gave 57% accuracy on the categorized CPCs; 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 Phase 2.

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	D CPC_Pr...	D 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)

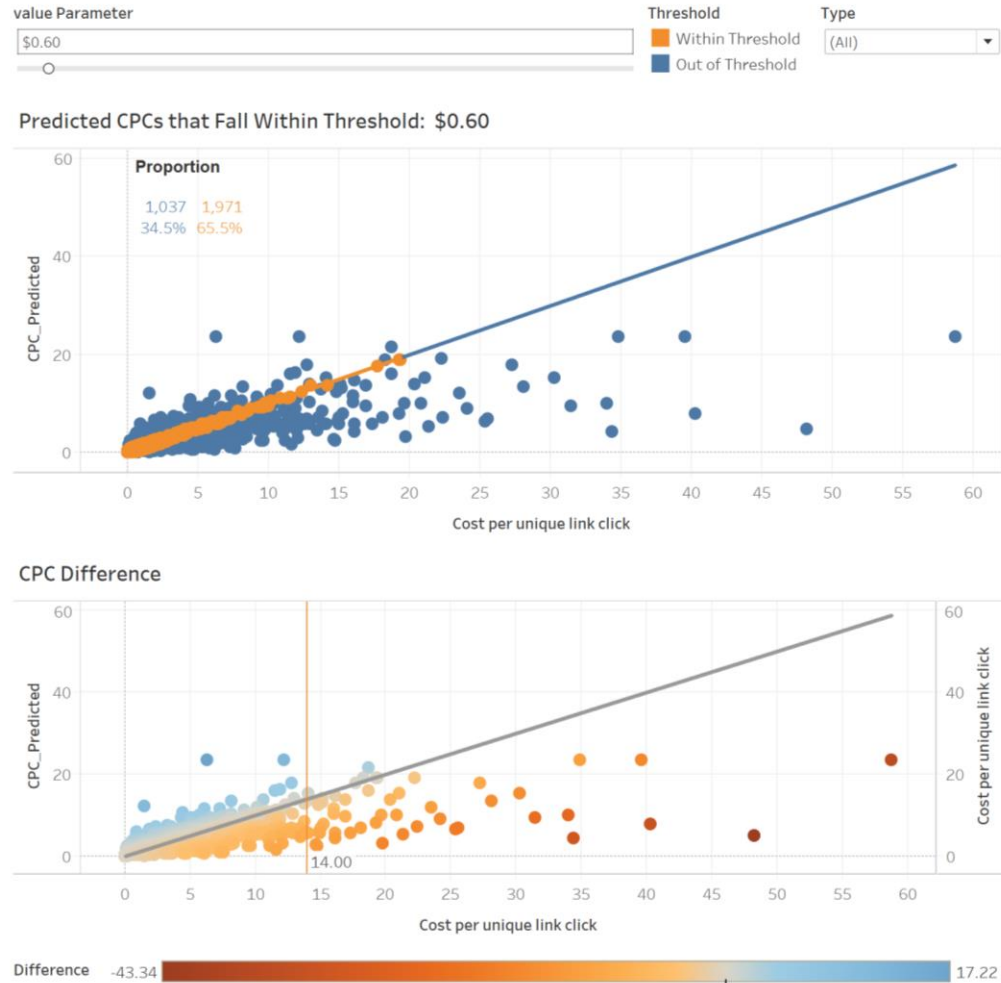
Our R^2 isn't great. However, during our test and validation phase, the marketers gave feedback that they can tolerate some inaccuracy in the predictions. Currently, this threshold is set at +/- \$0.60.

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 Phase 2's interim results.

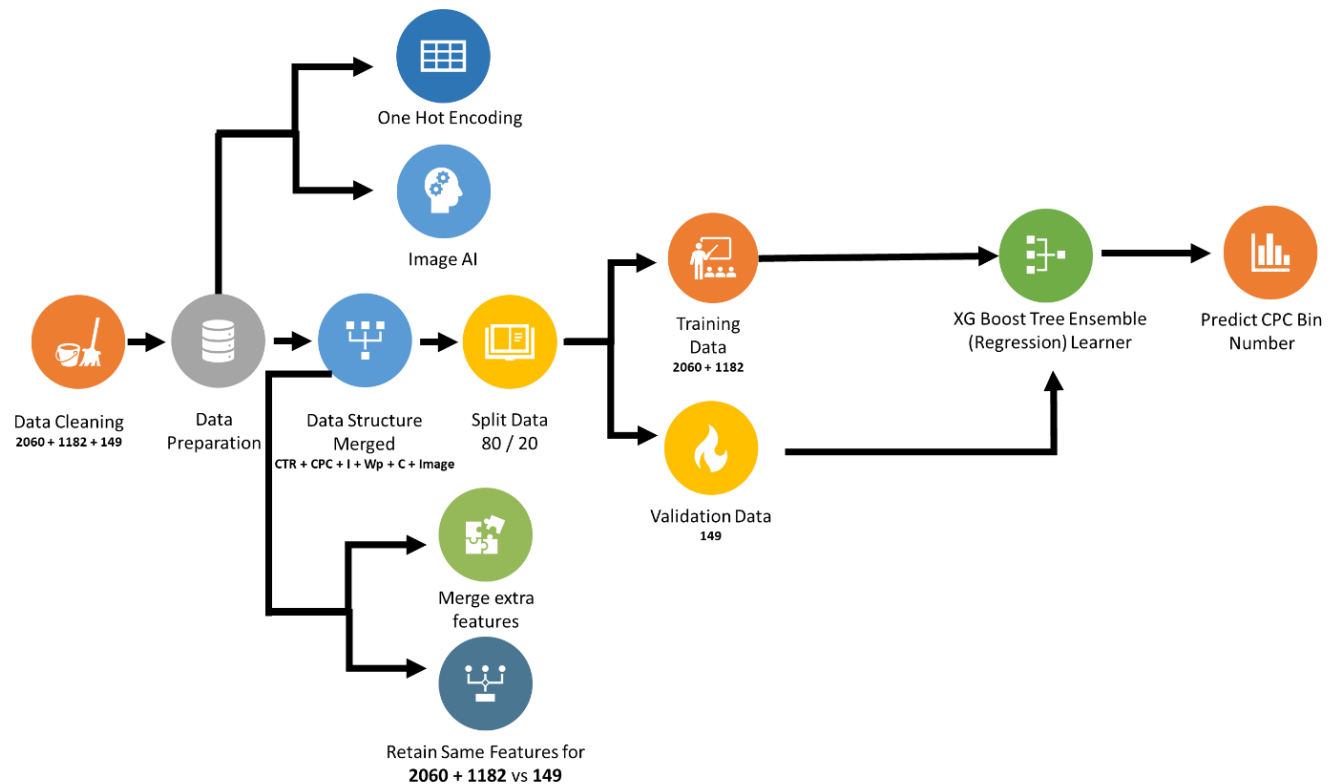
Distribution of CPC

0	1,469
1	632
2	352
3	157
4	90
5	65
6	39
7	42
8	26
9	31
10	13
11	20
12	15
13	6
14	8
15	5
16	6
17	2
18	4
19	5
20	2
21	2
22	2
23	1
24	1
25	2
27	1
28	1
30	1
31	1
34	3
39	1
40	1
48	1
58	1



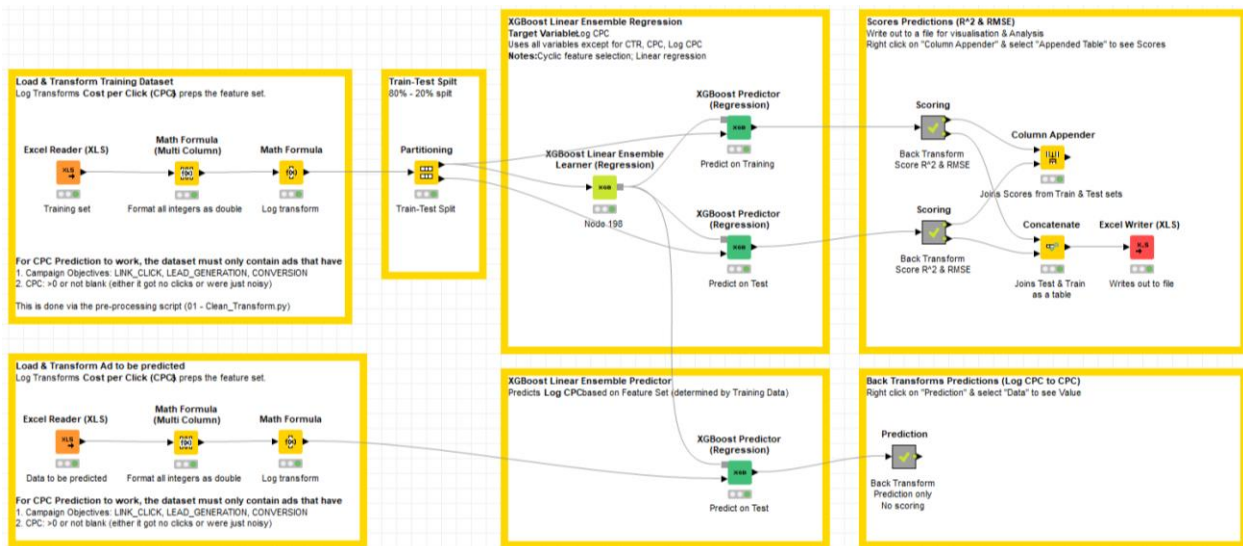
CPCs and imprecision Thresholds (Regression)

Thus, our process flow now looks like:



Process Flow

As mentioned in the section above that we have switched to using KNIME for our modelling steps due to the advantages stated. Below is the workflow of our KNIME setup



KNIME Data Model Workflow (Zoom in to 200% to see detail)

This workflow caters to both training of the model (XGBoost Linear Ensemble Learner (Regression)) and usage of model to predict CPCs for an ad. It will be stepped through during the demonstration.

4.2) Results / Validation and Verification

Validation Plan: Take 10 marketing campaigns spread out in 2019. Get the ads and targeting parameters (interests, work titles, countries). These parameters were chunked into the system, and predicted CPCs for each campaign. 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 #	Cost per unique link click (Actual CPC)	Predicted CPC	Difference
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.

Findings & Results Summary of System

1. This is a feature engineering problem. The bulk of time went into extraction and transformation of features from Interests, Work Titles, Countries, Objects in Ad Image, TFIDF 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.

	Phase 3	Phase 2 (Linear Regression)
R²	0.493 (1.9X better!)	0.2605
RMSE	\$3.063	\$0.003
Features	Interest, Work, Country, Objects, Text (5,145)	Interest, Country (112 features)

2. 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 via feature hashing and PCA, but the results were horrid). 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 & R² values.
3. 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.
4. 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.
5. The performance of this predictor system is at R² (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.

5) 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:

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.

5.1) Project Management

This project is run on the typical waterfall methodology , a linear approach. As much as we would like to run this project on a sprint basis, like what are doing in our day to day , we find that it is unnecessary to do so as we are a 2-man team so it’s easy to communicate and coordinate among us. Hence, our project plan follows the traditional waterfall methodology which makes more sense since the scope if going to be fixed more or less and not huge amount of change are expected from this project.

We used a shared google sheet to keep track of the key milestone of the project and review it on a weekly basis to make sure we aren’t deviating too far from the planned milestones. We also have a weekly meeting every Wednesday either via skype or meet up to align and update each other on individual progress.

The 1st overrun timeline was due to the change of scope. Our initial sponsor had left the co shortly and as a team of 5 , we struggle to find another project of a similar scope, hence the initial team was broken into 2 teams. Which resulted a slight delay of 3 days.

In Phase 1 , we had underestimated the part on defining the system workflow as it turns out to taking more time than we expected and exceeded by 4 days.

Coming to Phase 2, we actually plan up the bulk of time for implementation and testing of the models. Despite that , we still overran by 3 days.

In the final Phase 3, after the last presentation in Phase 2, we realized we need to work a lot more on refining and coming up with a better model. Testing and implementing the models took us extra 60 days than what was planned.

The experts we have are the employees from Construct Digital. The media planners and marketers took time off their work and help us test and execute the campaigns and extract the data. Lucky for us, Construct Digital is a company of 1 of our team mates, Edwin, hence, it's so much easy to coordinate and reach out to them and setting up meetings and follow ups. In total, the experts spent 8 hours of meeting with us and 4 hours of giving us feedback and help us test out our final solution / system.

Key lesson learnt from this type of project is that we should always cater more time for modelling and feature engineering tasks.

5.2) Results

Most of the System's results have been discussed at length in sections **4.0) Implementation and 4.2) Results / Validation and Verification**. More importantly, there is tacit 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 **5) 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.

5.3) Conclusion

Fascinating. 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 faced in this project – trialing of models, 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. That in itself is gratifying.

6) References

Sources

1. [A Factorization-Machine based Neural Network for CTR Prediction](#)
2. [Advertising CTR Prediction Based on Deep Neural Network](#)
3. [CNN features are also great at unsupervised classification](#)
4. [Deep CTR Prediction in Display Advertising](#)
5. [Discriminative Unsupervised Feature Learning with CNN](#)
6. [Feature Extraction and Image Recognition with Convolutional Neural Networks](#)
7. [Model Ensemble for Click Prediction in Bing Search Ads](#)
8. [Practical Lessons from Predicting Clicks on Ads at Facebook](#)
9. [Predict the CTR and CPC for Keywords](#)
10. [Visualizing and Understanding Deep Neural Networks in CTR Prediction](#)

7) Appendices

Project Proposal & Reports

1. [Project Plan & Milestones](#)
2. [Formal Project Proposal and Project Plan v2](#)
3. [System Design Report 1](#)
4. [System Design Report 2](#)

Glossary

1. **Creative**
Industry jargon. Refers to the entire ad (images and text). Interchangeable with ad.
2. **Click-Through-Rate (CTR)**
Campaign success metric. Calculated as total clicks / total impressions on ad. Expressed as a %.
3. **Cost-Per-Click (CPC)**
Campaign success metric. Calculated as total \$ spent / total clicks on ad. Expressed as a %.
4. **Impression**
Number of times an ad has been viewed. No distinction between unique or repeated views.
5. **Overall campaign performance**
See Cost-per-Click or Click-Through-Rate.
6. **Lead**
People who have signed up for more information from the client. They have not bought anything yet.

AdSet targeting Definition:

age_max	Target group maximum age
age_min	Target group minimum age
flexible_spec	OR-AND-EXCLUSION OR – Union

	<p>AND – Intersection</p> <p>EXCLUSION – Exclude</p> <p>E.g. (segment 1 or segment2 or segment3) and (segment 4 or segment 5) and segment 6</p> <p>flexible_spec=</p> <pre>[{ 'segment_type':[segment1, segment2], 'segment_type':[segment3] }, { 'segment_type':[segment4, segment5] }, { 'segment_type':[segment6] }]</pre> <p>So, if len(value in flexible_spec) > 1 , AND condition is used</p>
geo_locations	<p>geographical targeting field from country, region, city or zip</p> <p>OUTPUT: {"countries":["SG"],"location_types":["home"]}</p>
publisher_platforms	<p>Deliver ads on specific placements</p> <p>OUTPUT: Facebook, Instagram, Audience Network, Messenger</p>
facebook_positions	<p>Deliver ads on specific placements</p>

	<p>OUTPUT: feed, right_hand_column, instant_article, marketplace, and story.</p> <p>If you select story, you must use Facebook feed or Instagram story and for device_platforms, you must use mobile since Facebook Stories is mobile-only.</p>
device_platforms	<p>Deliver ads on specific placements</p> <p>OUTPUT: ["mobile", "desktop"]</p>
custom_audiences	<p>Array of audience IDs or audience objects. 'id' field only: [123, 456] or [{'id': 123}, {'id': 456}]</p>
genders	<p>0,1,2</p> <p>All, Men , Women (Not sure which is matched to which)</p>
<p>locales</p> <p>(How do I get the reference for each different number?)</p> <p>http://fbdevwiki.com/wiki/Locales#Languages_and_Codes_Previously_Proposed</p>	<p>-Target people with language other than common language for a location.</p> <p>-Provide an ID for the language, such as 5 for German.</p> <p>- E.g of output should be</p> <pre> { "key": 51, "name": "English (Upside Down)" }</pre> <p>-OUTPUT:[25], given are Publish code</p>
excluded_custom_audiences	<p>Array of audience IDs or audience objects. 'id' field only: [123, 456] or [{'id': 123}, {'id': 456}]</p>
instagram_positions	<p>Deliver ads on specific placements</p> <p>OUTPUT: stream, story</p>
friends_of_connections	<p>Target friends of people connected to your object. Connections are fans of your Page and people who interacted with your app.</p>

	OUTPUT: [{"id":"478136539192440","name":"Giant Super Saver's Club"}]
wireless_carrier	Allowed value is Wifi. Target mobile users currently on wifi networks.
targeting_optimization	<p>Targeting Expansion</p> <p>Enable Facebook to expand interests when this increases results at a lower cost per result. Targeting expansion does not create lookalike audiences. Expanding interests does not change your targeting specifications for location, demographic targeting, such as age or gender, or exclusions.</p> <p>Note that campaigns with supported objectives are not enabled by default.</p> <p>To opt in, set the targeting_optimization parameter to expansion_all.</p> <p>To opt out, set targeting_optimization parameter to none.</p> <p>-OUTPUT: "expansion_all" or "none"</p>
exclusions	Related to flexible_spec, to filter out from flexible_spec
audience_network_positions	<p>Deliver ads on specific placements</p> <p>OUTPUT: classic - feed, instream_video - when doing something, rewarded_video</p>
user_os	<p>For Brand Awareness</p> <p>OUTPUT: ['Android']</p>
excluded_connections	Array of Facebook IDs. Target people who are not fans of your page, have not authorized your Canvas App or have not logged into your mobile app with Facebook Login. You can also target people who did not RSVP to a future event. If you have global page and you want to exclude page fans, you can only target people who are not fans of this global page, instead of people who are not fans of your local page. To set this, you need to be an admin of the Page or event, or developer of the app being advertised. For a 'Page Likes' campaign, you must exclude your page from targeting.

	OUTPUT: [{ 'id':123}, { 'id':456}, 789]
messenger_positions	OUTPUT: messenger_home, story
interests	Area of interest
connections	<p>Array of Facebook IDs. Target fans of your Page, people who RSVP'd to your Event, people who logged into your app with Facebook Login, or authorized your Canvas app. To set this, you must be an admin of the page or event, or developer of the app you're advertising. You can't target on past events.</p> <p>OUTPUT: [{ 'id':123}, { 'id':456}, 789]</p>