International Journal of Artificial Intelligence and Applications (IJAIA), Vol.10, No.4, July 2019

A Novel Feature Engineering Framework in Digital Advertising Platform

Saeid SOHEILY-KHAH and Yiming WU

SKYLADS Research Team, Paris, France

Abstract. Digital advertising is growing massively all over the world, and, nowadays, is the best way to reach potential customers, where they spend the vast majority of their time on the Internet. While an advertisement is an announcement online about something such as a product or service, predicting the probability that a user do any action on the ads, is critical to many web applications. Due to over billions daily active users, and millions daily active advertisers, a typical model should provide predictions on billions events per day. So, the main challenge lies in the large design space to address issues of scale, where we need to rely on a subset of well-designed features. In this paper, we propose a novel feature engineering framework, specialized in feature selection using the efficient statistical approaches, which significantly outperform the state-of-the-art ones. To justify our claim, a large dataset of a running marketing campaign is used to evaluate the efficiency of the proposed approaches, where the results illustrate their benefits.

Keywords: Digital Advertising, Ad Event Prediction, Feature Engineering, Feature Selection, Statistical Test, Classification, Big Data.

1 Introduction

Digital advertising, which has only been around for two decades, is one of the most effective manners for all sizes companies and businesses to expand their reach, find new clients, and diversify their revenue streams. In the case, an ad event prediction system is critical to many web applications including recommender systems, web search, sponsored search, and display advertising [1,2,3,4,5], and is a hot research direction in computational advertising [6,7].

The event prediction is defined to estimate the ratio of events such as videos, clicks or conversions to impressions of advertisements that will be displayed. The impression, which also called an ad (advertisement) view, refers to the point in which an advertisement is viewed once by a user, or displayed once on a web page. In general, ads are announcements online about something such as a product or service, and the principal components in a marketer's paid advertising campaigns. They are sold on a 'Pay-Per-Click' (PPC) basis or even 'Pay-Per-Acquisition' (PPA), meaning the company only pays for ad clicks, conversions or any other pre-defined actions, not ad views. Hence, the Click-Through Rate (CTR) and the Conversion Rate (CVR) are very important indicators to measure the effectiveness of advertising display, and to maximize the expected value, one needs to predict

DOI: 10.5121/ijaia.2019.10403

the likelihood that a given ad will be an event, in the accurate way possible. As result, the ad prediction systems are essential to predict the probabilities of a user doing an action on the ad or not, and the performance of prediction model has a direct impact on the final advertiser and publisher revenues and plays a key role in the advertising systems. However, due to the information of advertising properties, user properties, and context environment, the ad event prediction is very fancy, challenging and complicated, and is a large-scale learning problem. It can influence ads pricing, and pricing impacts the advertisers return on their investments and revenue for the publishers.

Nowadays, almost all web applications, in multi-billion dollar digital advertising and marketing industry, relied heavily on the ability of learned models to predict ad event rates accurately, quickly and reliably [8,9,10]. Hence, even very small amount of improvement in ad event prediction accuracy would yield greater revenues in the hundreds of millions of dollars. While, with over billions daily active users (e.g. 3.8 billion internet users, 5 billion unique mobile users, and around 2.8 billion active social media users) and over millions active advertisers (e.g. more than 5 million active advertiser on Facebook, 8 million business profile, with more than 1 million active advertisers globally on Instagram), a typical industrial model should provide predictions on billions of events per day. Therefore, one of the main challenges in digital advertising industry lies in the large design space to address issues of scale. In the case, we need to rely on a set of well-designed features, for an efficient ad event prediction system. However, to capture the underlying data patterns, selecting, encoding, and normalizing the proper features has also pushed the field.

In this research paper, we discuss in detail the machine learning methods for ad event prediction, propose a novel feature engineering framework and a dedicated approach to predict more effectively whether an ad will be an event or not. The novel enhanced framework mainly facilitates feature selection using the proposed statistical techniques, thereby enabling us to identify, a set of relevant features. In the next section, we review the state-of-the-art of different classification techniques which widely used for ad event prediction applications. Next, in Section 3, we discuss machine learning and data mining approaches for feature engineering including feature selection, feature encoding and feature normalizing (scaling). In Section 4, we characterize the proposed feature engineering framework which could be directly applicable in any ad event prediction system. The deeply conducted experiments and results obtained on a running marketing campaign are discussed in Section 5. Lastly, Section 6 concludes the paper.

The main contributions of this research work are to a) introduce an enhanced framework for ad event-prediction by analyzing the huge amount of real-world data, where the framework includes pipelines for data pre-processing, feature selection, encoding and scaling, as well as training and prediction process, b) propose two novel adjusted statistical measures for feature selection, called adjusted chi-squared

test and adjusted mutual information, and c) illustrate through deeply experimental studies that the introduced framework significantly outperform the alternative ones. Notice that, to simplify, in the remainder of the paper, we consider the events as clicks. Of course, all the design choices, experiments and results can however be directly extended to any other events such as conversions, videos, etc.

2 State-of-the-art

Over the last decade, researchers have proposed many models for digital advertising and marketing that are usually based on machine learning methods, and therefore, different classification techniques such as logistic regression, gradient boosting, naive Bayesian, neural network and random forest have been widely used for ad event prediction applications.

2.1 Logistic regression

In statistical modeling, regression analysis is a statistical process to estimate the relationships among features. It includes many techniques for modeling and analyzing several features, when the focus is on the relationship between a dependent feature and one or more independent features (or predictors). More explicitly, regression analysis helps one understand how the typical value of the dependent feature changes when any one of the independent features is varied, while the other independent features are held fixed. Regression is the one of the most basic and commonly used predictive analysis, where there are a variety of different regression approaches such as linear regression, nonlinear regression, logistic regression, nonparametric regression, etc.

In the literature, (logistic) regression models have been used by many researchers to solve the ad event prediction problems for advertising [11,12,13,8]. In the case, logistic regression predicts the possibility of an event as:

$$P(y = 1|\mathbf{x}) = h_\theta(\mathbf{x}) = \frac{1}{1 + exp(-\theta^T \mathbf{x})} \tag{1}$$

and the probability of non event as:

$$P(y = 0|\mathbf{x}) = 1 - h_\theta(\mathbf{x}) \tag{2}$$

where $\mathbf{x}$ stands for all the feature in the data, and y is the event label (1 stands for an event and 0 stands for no event.). Lastly, θ is the weight that logistic regression assigned to all the feature. Note that the θ will change through the time, until it reach a optimized point, when the cost function is minimized.

2.2 Gradient boosting

One of the most powerful machine learning techniques, for different regression and classification problems, is gradient boosting. It produces a prediction model in the form of hybrid weak models, typically decision trees [14]. The boosting notion came out of the idea of whether a weak learning model can be changed to become better. The gradient boosting builds the model in a stage-wise manner like other boosting approaches do, and generalizes them by allowing the optimization of a loss function. The term 'gradient boosting' represents 'gradient descent' plus 'boosting', where the learning procedure sequentially fits novel models to provide a more accurate response estimation. In a nutshell, the principle idea behind it, is to construct the novel base-learners to have maximal correlation with the negative gradient of the loss function, associated with the whole hybrid model. Mathematically, it is a sequence of ensemble models, where each ensemble models can be regarded as a tree based model. Gradient boosting technique, practically, is widely used in many prediction applications due to its easy use, efficiency, accuracy and feasibility [8,15], as well as the learning applications [16,17].

2.3 Bayesian classifier

Bayesian classifiers, which refer to any classifier based on Bayesian probability or a naive Bayes classifier, are statistical approaches that predict class membership probabilities. They work based on the Bayes' rules (alternatively Bayes' law or Bayes' theorem), and the features are assumed to be conditionally independent. However, practically, even in spite of assumption of the features dependencies, they provide satisfying results. In general, Bayesian classifiers are easy to implement and fast to evaluate, where they just need a small number of training data to estimate their parameters. The main disadvantage is that Bayesian classifiers make a very strong assumption on the shape of data distribution. In addition, they can not learn interactions between the features, and suffer from zero conditional probability problem (division by zero) [18], where one simple solution would be to add some implicit examples. Furthermore, computing the probabilities for continuous features is not possible by the traditional method of frequency counts. Nevertheless, some studies have found that, with an appropriate pre-processing, Bayesian classifiers can be comparable in performance with other classification algorithms [10,19,20,21]. In the last decade, there is a very interesting research work by Microsoft team where they used a Bayesian online learning algorithm for the event prediction [10].

2.4 Neural network

A neural network is a network or circuit of neurons (or nodes), or according to what is considered acceptable today, an artificial neural network, composed of artificial

neurons. The neural networks are modeled based on the same analogy to the human brain working, and are a kind of artificial intelligence based approaches for ad event prediction problems [22]. Neural networks algorithms benefit from their learning procedures to learn the relationship between inputs and outputs by adjusting the network weights and biases, where the weight refers to strength of connections between two units (i.e. nodes) at different layers. Thus, they are able to predict the accurate class label of the input data.

In the litterateurs, the neural networks also have been used by many researchers in the ad event prediction scope. In [23], authors extended Convolutional Neural Network (CNN) for click prediction, however they are biased towards interactions between the neighboring features. Most up to date, authors, in [24], proposed a factorization machine-supported neural network algorithm, to investigate potential of training neural networks to predict ad clicks based on the categorical features. However, it is limited by the capability of factorization machines. In general, among deep learning frameworks, Feed Forward Neural Networks (FFNN) and Probabilistic Neural Networks (PNN) are claimed to be the most competitive algorithms for predicting ad events.

2.5 Random forest

Random forest [25] is an ensemble learning approach for regression, classification and such other tasks as estimating the feature importance, which operates by constructing a plenty of decision trees. Particularly, random forest technique is a combination of decision trees that all together produce predictions and deep intuitions into the data structure. While in standard decision trees, each node is split using the best split among all the features based on the Gini index, in random forest, each node is split among a small subset of randomly selected input features to grow the tree at each node. Therefore, in random forest, each tree will learn the data independently, the predicted result will be generated from the voting of the trees. For each tree during its learning process, the tree splits on one feature which can minimize the Gini index, defined as:

$$G = \sum_{m=1}^{M} \sum_{k=0}^{1} p_{mk}(1 - p_{mk}) \tag{3}$$

and with considering the depth of tree as $|T|$, it can be formulated as:

$$G = \sum_{m=1}^{M} \sum_{k=0}^{1} p_{mk}(1 - p_{mk}) + \alpha|T| \tag{4}$$

where m stands for m^{th} leaf, k stands for the k^{th} class, and α is a regularization parameter. The p_{mk} is the fraction of samples in class k, in leaf m. Here, for each

split, the aim is to minimize the Gini index, and the predicted result of an observation is the most commonly occurring class in the leaf which the observation is in.

This strategy yields to perform very well in comparison with many other classifiers such as support vector machine, neural network and nearest neighbor. Indeed, it makes them robust against the over-fitting problem as well as an effective tool for classification and prediction [26,27]. However, applying decision trees and random forests to display advertising, has additional challenges due to having categorical features with very large cardinality and the sparse nature of the data, in the literature, many researchers have used them in predicting ad events [28,29,30].

Nevertheless, one of the most vital and necessary steps in any event prediction system is to mine and extract features that are highly correlated with the estimated task. Moreover, many experiment studies are conducted to show that the feature engineering improves the accuracy of ad event prediction systems. The traditional event prediction models mainly depend on the design of features, while the features are artificially selected, encoded and processed. In addition, many successful solutions in both academia and industry rely on manually constructing the synthetic combinatorial features [31,32]. Because, the data sometimes has a complex mapping relationship, and taking into account the interactions between the features is vital. In the next section, we discuss about state-of-the-art of the feature engineering approaches, which can be considered as the core problem to digital advertising and marketing, prior to introduce our proposed approach in feature engineering and event prediction.

3 Feature engineering

Feature engineering is the fundamental to the application of machine learning, data analysis and mining as well as mostly all artificial intelligence tasks, and generally, is difficult, costly and expensive. In any artificial intelligence or machine learning algorithm (e.g. predictive and classification models), the features in the data are vital, and they dramatically influence the results we are going to achieve. Therefore, the quantity and quality of the features (and data) have huge influence on whether the model is good or not. In the following, we discuss the data pre-processing process and feature engineering in more detail and we present the most well-used methods in the case.

3.1 Feature selection

Feature selection is the process of finding a subset of useful features and removing irrelevant features to use in the model construction. It can be used for a) simplification of models to make them easier to expound, b) reduce training time consumption, c) avoid the curse of dimensionality and etc. In simple words, feature selection

determines the accuracy of a model and helps remove useless correlation in the data that might diminish the accuracy. In general, there are three types of feature selection algorithms: embedded methods, wrapper methods and filter methods.

Embedded methods Embedded approaches learn which features best contribute to model accuracy while it is being created. It means that some learning algorithms carry out the feature selection as part of their overall operation, such as random forest and decision tree, where we can evaluate the importance of features on a classification task. The most common kind of embedded feature selection approaches are regularization (or penalization) methods, which inset additional restrictions into the optimization of a predictive algorithm. Decision trees and random forests are the other commonly used techniques in embedded feature selection. But, how does random forest select features?

Random forest technique consists of hundreds of decision trees, where each of the trees built over a random extraction of features as well as data, and therefore, none of the trees will not encounter all the features. This characteristic guarantees that the trees be de-correlated and less prone to over-fitting. Each node of the tree is a question based on one or combination of features, where the tree divides into two groups. The more similar among themselves in one group and the different from the ones in the other group. Lastly, the importance of each feature is derived from how 'pure' each of the groups is. However, in the case, creating such a good model is challenging due to the time complexity of training models.

Wrapper methods Wrapper methods consider the selection of a subset of useful features as a search problem, where different combinations are constructed, evaluated and then compared to other ones. A predictive model used to assess the combination of features and assign a score based on the accuracy of the model, where the search process could be stochastic, methodical, or even heuristic. In the following, we discuss two well-known wrapper methods to select the best feature (or subset of features), in more details.

Genetic algorithm
Mathematically, feature selection is formulated as a combinatorial optimization problem, where the optimization function is the generalization performance of the predictive model, represented by the error on a selection data. As a complete selection of features would evaluate lots of different combinations, the process requires lots of computational work and even impracticable, if the number of features is big. Therefore, we need intelligent methods which allow to perform feature selection in practice.

One of the most advanced algorithms for feature selection is Genetic Algorithm (GA), which is a stochastic method for function optimization based on the mechan-

ics of natural genetics and biological evolution. GA, initially introduced in [33,34], is a type of optimization algorithm to find the optimal solution(s) of a given computational problem that maximizes or minimizes a particular function. It operates on a population of individuals to produce better and better approximations. In the case, each individual represents a predictive model, and the number of genes is the total number of features. Genes are binary values and represent the inclusion or not of particular features in the model. The main steps of the proposed feature selection algorithm using GA are as follows:

- Create and initialize the individuals in the population
- Fitness assignment: assign the fitness to each individual
- Selection: choose the individuals to recombine for the next generation
- Crossover: recombine the selected individuals to generate a new population
- Mutation: change randomly the value of some features in the offsprings
- Repeat the above step till termination criterion is satisfied

Generally, the genes of the individuals are usually initialized at random. To evaluate the fitness, we need to train the classification (predictive) model with the train data, and then evaluate its error with the test data. Obviously, a high error rate means a low fitness, and therefore, those individuals with greater fitness will have a greater probability of being selected for recombination. After that, selection operator chooses the individuals according to their fitness level, that will recombine for the next generation. Once the selection operator has chosen a pre-defined size of the population, the crossover operator recombines the selected individuals to generate a new population. In practice, the crossover operator picks two individuals at random and combines them to get offsprings (children) for the new population, until the new population has the same size than the old one. Since the above operator can generate offsprings that are very similar to the parents, it might cause a new generation with low diversity. The mutation operator solve this problem by changing the value of some features in the offsprings at random (randomly changing the state of the gene). The whole fitness assignment, selection, crossover and mutation process is repeated until a stopping criterion is satisfied. As a sequel, genetic algorithms can select the best subset of features for the predictive model, but they require a lot of computation.

In a nutshell, the most common advantages of the genetic algorithms are:
a) they usually perform better than traditional feature selection techniques,
b) they can manage data with huge number of features,
c) they don't need specific knowledge about the under study problem,
and
d) they can be easily paralelized in computer clusters.

However, they might be very expensive in computational terms, since evaluation of each individual requires building a classification (or predictive) model.

Sequential search
Sequential feature selection is one way of dimensionality reduction to avoid overfitting by reducing the complexity of the model. It learns which features are most informative at each time step, and then chooses the next feature depending on the already selected features. So, it automatically selects a subset of features that are most relevant to the problem to reduce the generalization error or to improve computational efficiency of the model by removing irrelevant features.

Feature subset selection is vital in a number of situations, such as: a) features may be expensive to obtain and (/or) train, b) we may want to extract meaningful rules from the classifier. Further more, fewer features means fewer parameters for optimization, mining and learning tasks, as well as reducing complexity and runtime. To do so, one need a search strategy to select candidate subsets and an appropriate objective function to evaluate the candidates. Since the exhaustive evaluation of feature subsets, for N number of features, involves 2^N combinations, a good search strategy is therefore needed to explore the optimal combination of features. In addition, the objective function evaluates the subsets and returns a measure of their goodness criteria, a feedback signal used by the search strategy to select new candidate subsets.

In simple words, given a feature set $\mathbf{x} = \{x_i | i = 1 : N\}$, we want to find the subset of $\mathbf{x}$ as $\mathbf{x}_{\text{subset}} = \{x_j | j = 1 : M\}$, w.r.t. $j < i$, which optimizes a pre-defined objective function, ideally the probability of correct classification. Therefore, the search strategy corresponds to the following steps:

- Initialize the algorithm with an empty set ($\mathbf{x}_{\text{subset}} = []$). so that the M, the size of the subset, is 0.
- If the feature $x_k, k \in 1 : N$, temporary added to $\mathbf{x}_{\text{subset}}$, maximizes the objective function, then add it to subset $\mathbf{x}_{\text{subset}}$ as an additional feature.
- Repeat the above step till there is no improvement in the objective function or the termination criterion is satisfied.

Of course, the objective function (or the termination criterion) could be any pre-defined function, which achieves the user goal.

Filter methods Filter feature selection approaches apply a statistical test to assign a goodness scoring value to each feature. The features are ranked by their goodness score, and then, either selected to removed from the data or to be kept. These filter selection methods are generally univariate and consider the feature independently (e.g. chi-square test), or in some cases, with regard to the dependent feature (e.g. correlation coefficient scores). In a nutshell, they are independent to the type of the predictive model, and therefore, the result of a filter method would be more general rather than a wrapper approach.

Typically, the filter approaches use one evaluation criteria from the intrinsic connections between the features to score a feature (or even a subset of features). The common evaluation criteria are correlation, mutual information, distance and consistency metrics.

3.2 Feature encoding

In machine learning, when we have categorical features, we often have a major issue: how to deal with categorical features? Practically, one can postpone the problem using a data mining or machine learning model which handle the categorical features (e.g. k-modes clustering), or deal with the problem (e.g. label encoding, one-hot encoding, binary encoding)

When we use such a learning model with categorical features, we mostly have three types of models: a) models handling categorical features accurately, b) models handling categorical features incorrectly, or c) models do not handling the categorical features at all.

Therefore, there is a need to deal with the following problem. Feature encoding points out to transforming a categorical feature into one or multiple numeric features. One can use any mathematical or logical approach to convert the categorical feature, and hence, there are many methods to encode the categorical features, such as: a) numeric encoding, which assigns an arbitrary number to each feature category, b) one-hot encoding which converts each categorical feature with m possible values into m binary features, with one and only one active, c) binary encoding to hash the cardinalities into binary values, d) likelihood encoding to encode the categorical features with the use of target (i.e. label) feature. From a mathematical point of view, it means a probability of the target, conditional on each category value, and e) feature hashing, where a one-way hash function convert data into a vector (or matrix) of features.

3.3 Feature scaling

Most of the times, the data will contain features highly varying in units, scales and ranges. Since, most of the machine learning and data mining algorithms use Eucledian distance between two data points in their computations, this makes a problem. To suppress this effect, we need to bring all features to the same level of unit, scale or range. This can be attained by scaling. Therefore, feature scaler is a utility that converts a list of features into a normalized format suitable for feeding in data mining and learning algorithms. In practice, there are four common methods to perform feature scaling: a) min-max scaling to rescale the range of features in [0, 1] or [-1, 1], b) mean normalisation to normalize the values between -1 and 1, c) standardisation, which swaps the values by their Z scores, and d) unit vector,

where feature scaling is done in consideration of the entire feature vector to be of unit length.

Notice that, generally, in any algorithm that computes distance or assumes normality (such as nearest-neighbor), we need to scale the features, while feature scaling is not indispensable in modeling trees, since tree based models are not distance based models and can handle varying scales and ranges of features. As more examples, we can speed up the gradient descent method by feature scaling, and hence, it could be favorable in training a neural network, where doing a features scaling in naive Bayes algorithms may not have much effect.

4 The design choices

Algorithm 1 presents briefly the proposed feature engineering framework, where in the following, we explain in detail the different steps of proposed feature learning approach for ad event prediction system.

Algorithm 1 The proposed feature engineering framework

input: <`data`> raw data, `threshold`
output: $\texttt{features}_{\texttt{cat}}$ (categorical features), $\texttt{features}_{\texttt{num}}$ (numerical features)

function pre-processing(`data`)
    remove duplication data
    rebild (or drop) missing (or incomplete) values
    remove (redundant) features with zero (and low) variance
return `data`

function feature selection(`data`)
    run proposed adjusted chi-squared-test (or adjusted mutual information)
return $\texttt{features}_{\texttt{cat}}$, $\texttt{features}_{\texttt{num}}$

function feature encoder($\texttt{features}_{\texttt{cat}}$)
    for each feature i **in** $\texttt{features}_{\texttt{cat}}$
        `d_value` $\Leftarrow$ number of distinct values for feature i
        if `d_value`>`threshold`
            do string indexer
            do feature hasher
        else (<`threshold`)
            do one-hot-encoder
        end if
    end for
return encoded $\texttt{features}_{\texttt{cat}}$

function feature scaler($\texttt{features}_{\texttt{num}}$)
    do normalization
return normalized $\texttt{features}_{\texttt{num}}$

Before doing any data processing and learning algorithm, we need to know the environment of a digital advertising (marketing). Marketing is the set of activities a company uses to advertise and sell products, services, jobs, etc. It includes reaching existing customers with additional offers and attracting new customers. Marketing agencies utilize various strategies, from simple ads to offering discounts, to entice people to buy their goods. Thereby, the most important terms in a digital advertising market environment are:
a) advertisers who pay the money to get their advertisement shown; they run marketing campaigns and interpret the data,
b) demand side platform (DSP) which used to purchase advertising from a marketplace in an automated fashion,
c) publishers who get money for showing the ads on their websites,
d) sell-side platform (SSP),
e) display ads,
f) advertising agencies, often referred to creative agencies, who create, make plan, and handle advertising and sometimes other forms of promotion and marketing for their clients,
g) audience or online users who are in such away the prospective buyers, and
h) campaigns which are the efforts made, usually over a pre-defined time period, to advertise, and as result generate some events (e.g. sell something).
Notice that, many DSPs partner with third-party data providers to offer the advertisers as much information as possible. Additionally, many DSPs allow customers to import their own data from a data management platform (DMP).

In the case, there are lots of sensors to record data from advertisers, publishers, audience, online users and etc, over time, and therefore, there are plenty of recorded information, attributes and measures in an executed marketing campaign. For instance, the logs services enable advertisers to access the raw, event-level data generated through the online platform in different ways. However, we are not interested in all of them. Lots of recorded information and data are not useful or available for us, even they increase the complexity. So, at the first step, we prune the raw data, before doing any mining task.

4.1 Data cleaning and pre-processing

While unreliable data has a highly destructive effect on the performance, data cleaning is the process of detecting and refining (or deleting) corrupt, outlier, anomaly or inaccurate data from a dataset. It refers to identifying defective, inaccurate, erroneous, inconsistent or irrelevant parts of the data, prior to replacing, modifying, removing or even reclaiming the dirty or coarse data, as well as removing any duplicate data. In simple words, the data cleaning converts data from an original raw form into a more convenient format.

Typically, in the case of digital marketing, when we receive the data, there are lots of duplication values, because of lack of centralized and accurate data gathering, recording or perfect online report generator tools. Of course, knowing the source of duplication can help a lot in the cleaning process. However, in the more autonomous way, for data cleaning, even we can rely on the historical data. Another stage in data cleaning is rebuilding missing or incomplete data, where there are different solutions depending on the kind of problem such as time series analysis, machine learning, etc, and it is very difficult to provide a general solution. But, before doing the data cleaning task, we have to figure out the reason why data goes missing, whereas the missing values happen in different manners, such as at random or not at random. Missing at random means that the data trends to be missing is not relevant to the missing data, but it is related to some of the observed data. Additionally, some missing values have nothing to accomplish with their hypothetical values or with the values of other features (i.e. variables). In the other hand, the missing data could be not at random. For instance, people with high salaries generally do not want to reveal their incomes, or the females generally do not want to reveal their ages. Here the missing value in 'age' feature is impacted by the 'gender' feature. So, we have to be really careful before removing any missing data. Finally, when we figure out why the data is missing, we can decide to abandon missing values, or to fill.

As a summary, the most important key benefits of data cleaning in digital marketing are: a) having accurate view of customers, users and audiences; b) improving data integration; and c) increasing revenue and productivity.

The customers and online users are the exclusive sources of any analysis and mining task and are central to all business decision making. However, they are always changing. Their natures, behaviours, likes and dislikes, their habits, as well as their expectations are in a constant stage of change. Hence, we need to remain on top of those fluctuations in order to make smart decisions. Also, integrating data is important to gain a complete view of the customers, users and audiences, and when data is dirty and laden with duplicates and errors, data integration process will be difficult. In typical, multiple channels and multiple departments often collect duplicate data for the same user. With data cleaning, we can omit the useless and duplicate data and integrate it more effectively;

4.2 Feature selection

In feature selection, we rely on the filter methods, and we try to fit two proposed adjusted statistical measures (i.e. mutual information, chi square test) to the observed data, then we select the features with the highest statistics. Suppose we have a label variable (i.e. the event label) and some feature variables that describe the data. We calculate the statistical test between every feature variable and the label variable and observe the existence of a relationship between the variable and the

label. If the label variable is independent of the feature variable, we can discard that feature variable. In the following, we present the proposed statistical measures in detail.

Adjusted chi-squared test A very popular feature selection method is chi-squared test (χ^2-test). In statistics, the chi-squared test is applied to test the independence of two events (i.e. features), where two events X and Y are defined to be independent, if $P(XY) = P(X)P(Y)$ or, equivalently, $P(X|Y) = P(X)$ and $P(Y|X) = P(Y)$. The formula for the χ^2 is defined as:

$$\chi^2_{df} = \sum \frac{(O_i - E_i)^2}{E_i} \tag{5}$$

where the subscript df is the degree of freedom, O is the observed value and E the expected value. The degrees of freedom (df) is equal to:

$$df = (r - 1) \times (c - 1) \tag{6}$$

where r is the number of levels for one categorical feature, and c is the number of levels for the other categorical feature. After taking the following chi-square statistic, we need to find p-value in the chi-squared table, and decide whether to accept or reject the null hypothesis (H_0). The p-value is the probability of observing a sample statistic as extreme as the test statistic, and the null hypothesis is the case that two categorical features are independent. Generally, small p-values reject the null hypothesis, and very large p-values means that the null hypothesis should not be rejected. As result, the chi-squared test gives a p-value, which tells if the test results are significant or not. In order to perform a chi-squared hypothesis test and get the p-value, one need a) the degree of freedom and b) the level of significance (α), while the default value is 0.05 (5%).

Like all non-parametric data, the chi-squared test is robust with respect to the distribution of the data [35]. However, it has difficulty of interpretation when there are large numbers of categories in the features, and tendency to produce relatively very low p-values, even for insignificant features. Furthermore, chi-squared test is sensitive to sample size, which is why several approaches to handle large data have been developed [36]. When the cardinality is low, it would be a little difficult to get a null hypothesis rejection, whereas a higher cardinality will be more intended to result a rejection.

In feature selection, we usually calculate the p-values of each feature, then choose those ones which are smaller than the 'preset' threshold. Normally, we use $\alpha = 0.95$, which stands for a threshold of 5% significance. For those features within this threshold, smaller p-value stands for better feature. However as mentioned before, higher cardinality will always cause a lower p-value. This means that the features with higher cardinality, *e.g. user identification*, or *site URLs*, are always

having lower p-values, and in turn, to be a better feature, which may not always be true.

In order to find a more reliable measure other than simply using p-value from chi-squared test, we proposed a new measure by adding a regularization term on the p-values (pv) of features, called 'adjusted p-value' (p_{adj}). The new proposed statistical measure, p_{adj}, is defined as:

$$p_{\text{adj}} = \frac{\chi^2_{1-pv,df} - \chi^2_{1-\alpha,df}}{\chi^2_{1-\alpha,df}} \tag{7}$$

where α is the level of significance, and df is the degrees of freedom. By using this quantity, we are penalizing on the features with higher cardinality. Simply to say, we are trying to see how further by percentage the critical value corresponding to pv, the $\chi^2_{1-pv,df}$, is crossing the critical value $\chi^2_{1-\alpha,df}$, corresponding to a given significance level $1-\alpha$. Note that, the $\chi^2_{1-\alpha,df}$ will be very big for high cardinality features due to higher degree of freedom, and it is regarded as a penalization term.

The penalization could also be softer, if we take the logarithm of the critical value $\chi^2_{1-\alpha,df}$. In the case, the adjusted p-value with soft penalization, $\hat{p}_{\text{adj}}$, can be formulated as:

$$\hat{p}_{\text{adj}} = \frac{\chi^2_{1-pv,df} - \chi^2_{1-\alpha,df}}{\log(\chi^2_{1-\alpha,df})} \tag{8}$$

For the two proposed above measures, higher value stands for better feature.

Adjusted mutual information Similar to the Chi-square test, the Mutual Information (MI) is a statistic quanitity which measures how much a categorical variable tells another (mutual dependence between the two variables). The mutual information has two main properties; a) it can measure any kind of relationship between random variables, including nonlinear relationships [37], and b) it is invariant under the transformations in the feature space that are invertible and differentiable [38]. Therefore, it has been addressed in various kinds of studies with respect to feature selection [39,40]. It is formulated as:

$$I(X;Y) = \sum_{x \in X, y \in Y} P(x,y) \log\Big(\frac{P(x,y)}{P(x)P(y)}\Big) \tag{9}$$

where $I(X;Y)$ stands for the 'mutual information' between two discrete variables X and Y, $P(x,y)$ is the joint probability of X and Y, and $P(x)$ and $P(y)$ are the marginal probability distribution of X and Y, respectively. The MI measure is a non-negative value, and it is easy to deduce that if X is completely statistically independent from Y, we will get $P(x,y) = P(x)P(y)$ $s.t.$ $x \in X, y \in Y$, which indicates a MI value of 0. The MI is bigger than 0, if X is not independent from Y.

In simple words, the mutual information measures how much knowing one of the variables reduces uncertainty about the other one. For example, if X and Y are independent, then knowing X does not give any information about Y and vice versa, so their mutual information is zero.

In the case of feature selection using the label column (Y), if MI is equal to 0, then X is considered as a 'bad' single feature, while the bigger MI value suggests more information provided from the feature X, which should be remained in the classification (predictive) model. Furthermore, when it comes to optimal feature subset, one can maximize the mutual information between the subset of selected features $\mathbf{x}_{\text{subset}}$ and the label variable Y, as:

$$\zeta = arg \max_{\text{subset}} I(\mathbf{x}_{\text{subset}}; Y) \tag{10}$$

where $|\text{subset}| = k$, and k is the number of features to select. The quantity ζ is called 'joint mutual information', and its maximizing is an NP-hard optimization problem.

However, the mutual information is subject to a Chi-square distribution, that means we can convert the mutual information to a p-value as a new quantity. This new adjusted measure, called MI_{adj}, will be more robust than the standard mutual information. Also, we can rule out those features that is not significant based on the calculated p-value. Normally:

$$2N \times \text{MI} \sim \chi^2_{df} \tag{11}$$

where the N stands for the number of data samples, and similar as before, *df* is degree of freedom. So simply to say, our proposed new filtering rule (MI_{adj}) is defined by:

$$2N \times I(X;Y) - \chi_{0.95,df(X,Y)} > 0 \tag{12}$$

The bigger MI_{adj}, the better is. Some features will be ruled out if their new adjusted measures are negative, which indicate the mutual information are not significant comparing to their degrees of freedom.

4.3 Feature encoding

After selecting the proper features, we need to format the data, which can be accepted by the training model. Practically, we do the training and prediction process tasks using Spark in Yarn, because we have nearly forty million records to analyze on a weekly basis. Specifically, we use the StringIndexer, which encodes the features by the rank of their occurrence times, for high cardinality features above a predefined threshold, and one hot encoder for the features whose unique levels less then the predefined threshold. We also hash the high cardinality features to

ensure that we are formatting the data without loosing too much information. In a nutshell, in our ad event prediction case, there are some extremely high cardinality features like *user ids*, or *page urls* with millions levels on weekly basis. To keep most of the useful information without facing the risk of explosion of feature numbers at the meantime, it would be preferable to hash them rather than doing one-hot encoding.

4.4 Feature scaling

In the last step of our proposed feature engineering framework, using the max-min scaling method, we normalize the features, if needed.

5 Experimental study

In this section, we first describe the dataset used to conduct our experiments, then specify the validation process, prior to present and discuss the results that we obtained.

5.1 Data description

To clarify our claim in ad event prediction system, we used a large real-world dataset of a running marketing campaign. The dataset is a private activity report from MediaMath digital advertising platform, is very huge, and the entire dataset is stored on cloud storage (i.e. Amazon S3) of Amazon Web Services (AWS). It comprises over 40 millions of recorded ads data (on weekly basis), each one with more than 80 pieces of information, which can be categorized in two main group: a) attributes which can be considered as features, and b) event labels for machine learning and mining tasks.

Attributes (features) The input features for machine learning algorithms, such as *user id*, *site url*, *browser*, *date*, *time*, *location*, *advertiser*, *publisher* and *channel*. Notice, that it is highly probable that one generates new features from the existence ones. For instance, from *start-time* and *stop-time*, we can produce the *duration* feature, or from *date*, we can generate the new feature *day of week*, which will be more meaningful in the advertising campaign.

Measures (labels) Measures are the target variables data which acts as labels in machine learning and data mining algorithms, such as *impressions*, *clicks*, *conversions*, *videos*, and *spend*. Note that, the mentioned measures (i.e. labels) are needed for supervised learning algorithms, while in non-supervised algorithm one can ignore them.

5.2 Validation process

In our experimental studies, we here compare the proposed feature engineering framework in the ad event prediction system with the state-of-the-art and the well-used feature engineering based event prediction methods. To do so, for our comparisons, we rely on the accuracy, recall or true positive rate, precision or positive predictive value, F_1-score, which is a harmonic mean of precision and recall, as well as the area under precision-recall curve (AUC-PR), which are commonly used in the literature, to evaluate each method.

The accuracy measure lies in [0, 100] in percentage, and true positive rate (recall), positive predictive value (precision), and F_1-score lie within a range of [0, 1]. The higher index, the better the agreement is. In the other side, precision-recall is a useful measure of success of prediction when the classes are very imbalanced, and the precision-recall curve shows the trade-off between precision and recall for different threshold. A high area under the curve represents both high recall and high precision. High scores illustrate that the predictor (or classifier) is returning accurate results (high precision), as well as returning a majority of all positive results (high recall). Notice that, the Precision-Recall (PR) summarizes such a curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight.

In the experiments, for all approaches, the parameters as well as the training and testing sets are formed by k-fold cross validation in the ratio of 80% and 20% of the data, respectively. For example, for the random forest algorithm two parameters are tuned: number of trees and minimum sample leaf size. Finally, the results reported hereinafter are averaged after 10 repetitions of the corresponding algorithm.

5.3 Experimental results

In the context of event prediction, the accuracy, recall, precision, F_1-score, as well as the area under PR curve (AUC-PR), for the various tested methods, are reported in the Table 1. Many papers in the literature have shown that the random forest are among the most efficient methods to be considered, as far as heterogeneous multivariate data are concerned [41,42]. Hence, we build our proposed event prediction algorithm on the basis of a Random Forest (RF) classifier. To facilitate the big data analysis task, we do the data pre-processing, training, and ad event prediction by running spark jobs on Yarn. The results in bold correspond to the best assessment values.

Table 1 shows the comparison of performances of a simple classifier versus the case of doing a feature engineering before running the classifier, on historical data of a running campaign. Here, in the context of feature selection, in addition to the filter methods, i.e. standard chi-squared test (χ^2) and standard mutual information (MI), we compare our proposed statistical model with two well-known wrapper

Table 1: Comparison of performances based on the different feature engineering methods

	Accuracy	Recall	Percision	F_1-score	AUC-PR
RF	99.61	0.050	0.013	0.022	0.005
RF + Feature Eng. (χ^2)	99.92	0.000	0.000	0.000	0.005
RF + Feature Eng. (χ^2-p_{adj})	**99.94**	**0.051**	**0.044**	**0.047**	**0.010**
RF + Feature Eng. (χ^2-$\hat{p}_{\text{adj}}$)	99.92	0.000	0.000	0.000	0.006
RF + Feature Eng. (MI)	99.92	0.000	0.000	0.000	0.006
RF + Feature Eng. (MI_{adj})	99.91	0.008	0.023	0.012	0.006
RF + Feature Eng. (GA)	99.92	0.000	0.000	0.000	0.004
RF + Feature Eng. (SS)	99.92	0.000	0.000	0.000	0.003

approaches, i.e. genetic algorithm (GA) and sequential search (SS). Notice that, in filter methods, we consider only top 20 features. Of course, one can simply find the best number of features using a k-fold cross validation technique.

As has been pointed out in Table 1, while in feature selection using the standard statistical approaches (i.e. χ^2 and MI) and the wrapper methods (i.e. GA and SS), the random forest classifier can not provide any good results for accuracy, recall, precision, F_1-score and the area under precision-recall curve, using our proposed statistical measures (i.e. χ^2-p_{adj} and MI_{adj}), we generally outperform the results. Also, it is plain to see that the RF classifier with considering feature selection based on the proposed χ^2-p_{adj} has the best results, and outperforms significantly the precision, recall, F_1-score as well as the area under precision-recall curve. However, for this case, the soft penalized version of χ^2-p_{adj} (i.e. χ^2-$\hat{p}_{\text{adj}}$) does not provide very good result, but still it is better than the standard χ^2 and is comparable with the standard mutual information measure. Furthermore, as demonstrated, using the proposed adjusted version of mutual information in feature selection process, provides better results rather than using the standard mutual information measure. Lastly, the worst result belongs to the case of using genetic algorithm (GA) and sequential search (SS) in the feature engineering process.

To verify our claim and consolidate the comparative results, we use a Wilcoxon signed rank test, which is a non-parametric statistical hypothesis test to effectively determine whether our proposed adjusted statistic measures are significantly outperform the classifier (using the alternative quantities) or not. Tables 2 presents the two-sided p-value for the hypothesis test, while the results in bold indicate the significantly different ones. The p-value is the probability of observing a test statistic more extreme than the observed value under the null hypothesis. Notice that, the null hypothesis is strongly rejected when the p-values are lower than a pre-defined criterion, almost always set to 0.05. It means that the differences between the two tested classifiers are significant and the uniform hypothesis is accepted as p-values are greater than 0.05. Based on the p-values, we can justify that using the proposed adjusted measures in feature selection, the classifier leads to significantly better re-

sults than the others. Note that the difference between the pairs of classifiers results follows a symmetric distribution around zero and to be more precise, the reported p-values are computed from all the individual results after some repetitions of the corresponding algorithm.

Table 2: P-values: Wilcoxon test

	RF + Featrue Engineering						
	(χ^2)	(χ^2-p_{adj})	(χ^2-$\hat{p}_{\text{adj}}$)	(MI)	(MI_{adj})	(GA)	(SS)
RF	**0.034**	**0.004**	0.073	0.073	0.798	**0.036**	**0.036**
RF+Feat. Eng. (χ^2)		**0.011**	0.157	0.157	**0.011**	0.157	0.157
RF+Feat. Eng. (χ^2-p_{adj})			**0.011**	**0.011**	**0.011**	**0.011**	**0.011**
RF+Feat. Eng. (χ^2-$\hat{p}_{\text{adj}}$)				1.000	**0.026**	0.157	0.157
RF+Feat. Eng. (MI)					**0.026**	0.157	0.157
RF+Feat. Eng. (MI_{adj})						**0.011**	**0.011**
RF+Feat. Eng. (GA)							0.157

As can be seen from Table 2, with regard to the p-values of Wilcoxon test, the random forest classifier with considering feature selection method using the proposed χ^2-p_{adj} measure, brings a significant improvement compared to the other methods. In addition, the proposed MI_{adj} is almost performing significantly different than the other approaches.

Table 3: Feature selection: top-k extracted features using different statistical test (rank sorted)

χ^2	χ^2-p_{adj}	χ^2-$\hat{p}_{\text{adj}}$	MI	MI_{adj}
os_id	id_vintage	mm_uuid	mm_uuid	publisher_id
category_id	os_id	contextual_data	contextual_data	site_id
contextual_data	fold_position	page_url	page_url	site_url
week_hour	advertiser_id	user_agent	site_id	main_page
concept_id	conn_speed_id	strategy_id	strategy_id	supply_source_id
overlapped_b_pxl	watermark	creative_id	site_url	exchange_id
week_hour_part	browser_name	site_id	publisher_id	category_id
site_id	browser_language	zip_code_id	creative_id	creative_id
norm_rr	week_hour	ip_address	user_agent	channel_type
page_url	width	site_url	main_page	concept_id
strategy_id	cross_device_flag	overlapped_b_pxl	zip_code_id	campaign_id
app_id	browser_version	aib_recencies	concept_id	browser_name
creative_id	week_hour_part	main_page	campaign_id	browser_version
browser_version	device_id	city_code_id	city_code_id	model_name
zip_code_id	app_id	concept_id	supply_source_id	homebiz_type_id
form_factor	city	campaign_id	exchange_id	os_id
supply_source_id	week_part_hour_part	channel_type	category_id	os_name
prebid_viewability	category_id	homebiz_type_id	model_name	brand_name
brand_name	norm_rr	os_id	norm_rr	form_factor
exchange_id	dma_id	form_factor	channel_type	norm_rr

To become closely acquainted with selected features, Table 3 shows the top twenty selected features using different feature selection models (i.e. statistical test) which we consider in our experimental study. In the term of time consumption, all the proposed adjusted statistical test (i.e. χ^2-p_{adj}, χ^2-$\hat{p}_{\text{adj}}$, MI_{adj}), have more or less the same time complexity with the standard ones (i.e. χ^2, MI), without any significant difference. Furthermore, Table 4 illustrates the best subset of features using different wrapper methods (i.e. GA and SS). Provided results using feature engineering with GA are obtained by considering the population size equals to 100, crossover and mutation probability equals to 0.2, and after 10 iteratons.

Table 4: Feature selection: best subset of features using different wrapper methods

Method	Selected features
Genetic algorithm	`publisher_id`, `category_id`, `day_of_week`, `week_hour_part`, `mm_creative_size` `aib_recencies`
Sequential search	`publisher_id`, `category_id`, `advertiser_id`, `dma_id`, `aib_recencies`, `day_of_week`, `aib_pixel_ids`

Lastly, to have a closer look at the ability of the proposed statistical quantities, Figure 1 shows the comparison of the area under PR curve for different feature engineering methods on the basis of a random forest classifier. Note that, we consider the area under PR curve, since it is more reliable rather than ROC curve, because of the unbalanced nature of data. The higher values are the better performance.

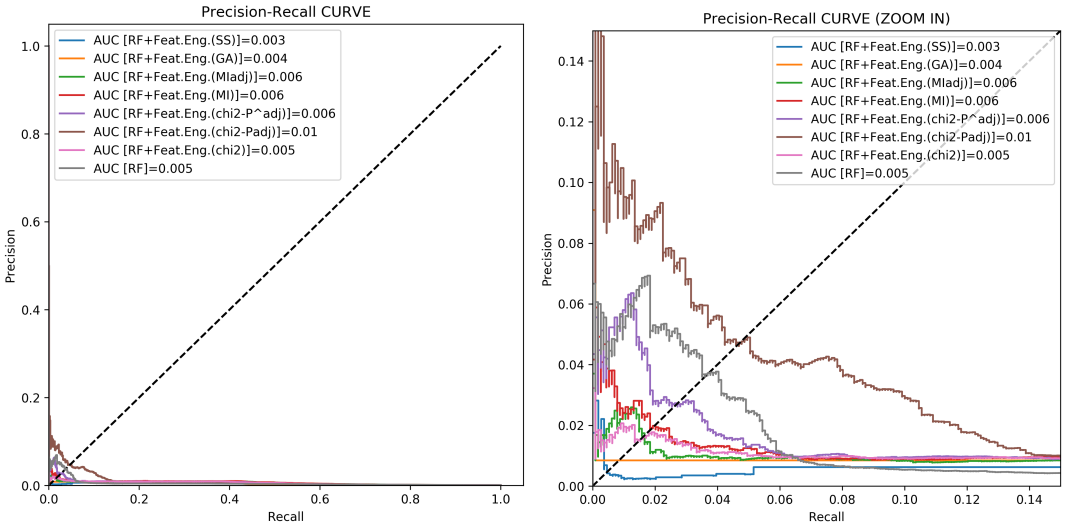

Fig. 1: Comparison of AUC-PR curve based of different feature engineering methods

6 Conclusion

This paper proposes a novel feature engineering framework for ad event prediction in digital advertising platform which has been applied on big data. In this case, we introduce two new statistical measures which can be used for feature selection: i) the adjusted chi-squared test and ii) the adjusted mutual information. While the standard statistical models (i.e. chi-squared test and mutual information) have some problems such as difficulty of interpretation and sensitivity to sample size, the proposed statistical ones overcome these difficulties. Furthermore, in feature encoding step before training the model, we used a practical and reliable pipeline to encode very large (categorical) data. The efficiency is analyzed on a large historical data from a running campaign. The results illustrate the benefits of the proposed adjusted chi-squared test and the adjusted mutual information, which outperform the alternative ones with respect to different metrics, i.e. accuracy, precision, recall, F_1-score and the area under precision-recall curve. Lastly, to determine that the proposed approaches is significantly better than the other described methods, a Wilcoxon signed-rank test is used. While, in this paper, we focus on the single features, the idea of combined features can be a proper proposition to gain better result. Hence, investigate the combination of some features to generate more useful features, to further increase the prediction performance of the imbalanced case, which is typical in the context of digital advertising, can be an interesting suggestion for future works.

References

1. H. Cheng and E. Cantú-Paz, "Personalized click prediction in sponsored search," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, (New York, NY, USA), pp. 351–360, ACM, 2010.
2. Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu, "Sequential click prediction for sponsored search with recurrent neural networks," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1369–1375, AAAI Press, 2014.
3. O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and scalable response prediction for display advertising," *ACM Trans. Intell. Syst. Technol.*, vol. 5, pp. 61:1–61:34, Dec. 2014.
4. A. Borisov, I. Markov, M. de Rijke, and P. Serdyukov, "A neural click model for web search," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 531–541, 2016.
5. H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, (New York, NY, USA), pp. 7–10, ACM, 2016.
6. C. Li, Y. Lu, Q. Mei, D. Wang, and S. Pandey, "Click-through prediction for advertising in twitter timeline," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 1959–1968, ACM, 2015.
7. J. Chen, B. Sun, H. Li, H. Lu, and X.-S. Hua, "Deep ctr prediction in display advertising," in *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, (New York, NY, USA), pp. 811–820, ACM, 2016.

8. M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: Estimating the click-through rate for new ads," in *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, (New York, NY, USA), pp. 521–530, ACM, 2007.
9. D. Agarwal, B. C. Chen, and P. Elango, "Spatio-temporal models for estimating click-through rate," in *WWW '09: Proceedings of the 18th international conference on World wide web*, (New York, NY, USA), pp. 21–30, ACM, 2009.
10. T. Graepel, J. Q. n. Candela, T. Borchert, and R. Herbrich, "Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, (USA), pp. 13–20, Omnipress, 2010.
11. O. Chapelle, "Modeling delayed feedback in display advertising," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, (New York, NY, USA), pp. 1097–1105, ACM, 2014.
12. M. J. Effendi and S. A. Ali, "Click through rate prediction for contextual advertisment using linear regression," *CoRR*, vol. abs/1701.08744, 2017.
13. H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica, "Ad click prediction: a view from the trenches," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.
14. J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, pp. 367–378, Feb. 2002.
15. I. Trofimov, A. Kornetova, and V. Topinskiy, "Using boosted trees for click-through rate prediction for sponsored search," in *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, ADKDD '12, (New York, NY, USA), pp. 2:1–2:6, ACM, 2012.
16. C. J. C. Burges, "From RankNet to LambdaRank to LambdaMART: An overview," tech. rep., Microsoft Research, 2010.
17. K. S. Dave and V. Varma, "Learning the click-through rate for rare/new ads from similar ads," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, (New York, NY, USA), pp. 897–898, ACM, 2010.
18. P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2, pp. 103–130, 1997.
19. R. Entezari-Maleki, A. Rezaei, and B. Minaei-Bidgoli, "Comparison of classification methods based on the type of attributes and sample size.," *JCIT*, vol. 4, no. 3, pp. 94–102, 2009.
20. M. Kukreja, S. A. Johnston, and P. Stafford, "Comparative study of classification algorithms for immunosignaturing data.," *BMC Bioinformatics*, vol. 13, p. 139, 2012.
21. A. C. Lorena, L. F. Jacintho, M. F. Siqueira, R. D. Giovanni, L. G. Lohmann, A. C. de Carvalho, and M. Yamamoto, "Comparing machine learning classifiers in potential distribution modelling," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5268 – 5275, 2011.
22. G. Zhou, C. Song, X. Zhu, X. Ma, Y. Yan, X. Dai, H. Zhu, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," *CoRR*, vol. abs/1706.06978, 2017.
23. Q. Liu, F. Yu, S. Wu, and L. Wang, "A convolutional click prediction model," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, (New York, NY, USA), pp. 1743–1746, ACM, 2015.
24. W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data: A case study on user response prediction," in *ECIR*, 2016.
25. L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
26. A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
27. S. Soheily-Khah, P. Marteau, and N. Béchet, "Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the iscx dataset," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, pp. 219–226, April 2018.

28. K. Dembczynski, W. Kotlowski, and D. Weiss, "Predicting ads âĂŹ click-through rate with decision rules," in *WWW2008, Beijing, China*, 2008.
29. I. Trofimov, A. Kornetova, and V. Topinskiy, "Using boosted trees for click-through rate prediction for sponsored search," in *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, ADKDD '12, (New York, NY, USA), pp. 2:1–2:6, ACM, 2012.
30. L. Shi and B. Li, "Predict the click-through rate and average cost per click for keywords using machine learning methodologies," in *Proceedings of the International Conference on Industrial Engineering and Operations ManagementDetroit, Michigan, USA*, 2016.
31. Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao, "Deep crossing: Web-scale modeling without manually crafted combinatorial features," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 255–262, ACM, 2016.
32. S. Soheily-Khah and Y. Wu, "Ensemble learning using frequent itemset mining for anomaly detection," in *International Conference on Artificial Intelligence, Soft Computing and Applications (AIAA 2018)*, 2018.
33. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.
34. M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.
35. M. L. McHugh, "The chi-square test of independence," *B. M.*, vol. 23, pp. 143–149, 2013.
36. D. Bergh, "Sample size and chi-squared test of fit: A comparison between a random sample approach and a chi-square value adjustment method using swedish adolescent data.," *In Pacific Rim Objective Measurement Symposium Conference Proceedings*, pp. 197–211, 2015.
37. T. M. Cover and J. A. Thomas, *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006.
38. S. Kullback, *Information Theory And Statistics*. Dover Pubns, 1997.
39. S. Cang and H. Yu, "Mutual information based input feature selection for classification problems," *Decision Support Systems*, vol. 54, no. 1, pp. 691 – 698, 2012.
40. J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, pp. 175–186, Jan. 2014.
41. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, Jan. 2014.
42. M. Wainberg, B. Alipanahi, and B. J. Frey, "Are random forests truly the best classifiers?," *J. Mach. Learn. Res.*, vol. 17, pp. 3837–3841, Jan. 2016.

Authors

Saeid SOHEILY KHAH graduated in (software) computer engineering, and he received master degree in artificial intelligence and robotics. Then he received his second master degree in information analysis and management from Skarbek university, Warsaw, Poland, in 2013. In May 2013, he joined to the LIG (Laboratoire d'Informatique de Grenoble) at Université Grenoble Alpes as a doctoral researcher. He successfully defended his dissertation and got his Ph.D in Oct 2016. Instantly, he joined to the IRISA at Université Bretagne Sud as a postdoctoral researcher.

Lastly, in Oct 2017, he joined Skylads (artificial intelligence and big data analytics company) as a research scientist. His research interests are machine learning, data mining, cyber security system, digital advertising and artificial intelligence.

Yiming WU received his B.S.E.E. degree from the Northwestern Polytechnical University, Xi'an in China. He received his Ph.D. degree in Electrical Engineering from University of Technology of Belfort-Montbéliard, Belfort, France, 2016. He joined Skylads as a data scientist in 2018, and his research has addressed topics on machine learning, artificial intelligence and digital advertising.