MEF UNIVERSITY

AD CLICK PREDICTION USING MACHINE LEARNING ALGORITHMS

Capstone Project

Nazlı Tuğçe Uncu

İSTANBUL, 2021

MEF UNIVERSITY

AD CLICK PREDICTION USING MACHINE LEARNING ALGORITHMS

Capstone Project

Nazlı Tuğçe Uncu

Advisor: Asst. Prof. Dr. Hande Küçükaydın

İSTANBUL, 2021

MEF UNIVERSITY

Name of the project: Click Prediction in Practice
Name/Last Name of the Student: Nazlı Tuğçe Uncu
Date of Thesis Defense: 25/01/2021

I hereby state that the graduation project prepared by Nazlı Tuğçe Uncu has been completed under my supervision. I accept this work as a "Graduation Project".

25/01/2021
Asst. Prof. Dr. Hande Küçükaydın

I hereby state that I have examined this graduation project by Nazlı Tuğçe Uncu which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

25/01/2021

Prof. Dr. Ozgur Ozluk

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of _____ and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Asst. Prof. Dr. Hande Küçükaydın	……
2. ……..	……

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Name	Date	Signature
Nazlı Tuğçe Uncu	25.01.2021	

EXECUTIVE SUMMARY

AD CLICK PREDICTION USING MACHINE LEARNING ALGORITHMS

Nazlı Tuğçe Uncu

Advisor: Asst. Prof. Dr. Hande Küçükaydın

JANUARY, 2021, 28 pages

Online advertising has a great potential to boost business' revenue. One of the key metrics that defines the success of online ad campaigns is click through rate (CTR) which indicates the total number of clicks received in relation to the total impression. Therefore, the click prediction systems, which have the aim of increasing the click through rates of online advertising campaigns by predicting the clicks, have become essential for businesses. For this reason, predicting whether an advertisement will receive a click from the user or not attracts the attention of researchers from the both industry and academia.

In this capstone project, the click prediction is studied by using Avazu's click logs dataset. The effects of having high cardinality categorical features and imbalanced data are examined during data preprocessing phase and then relevant features are selected to be used in modeling. The methods that are used for this classification problem are decision trees, random forest, *k*-nearest neighbor, extreme gradient boosting, and logistic regression. According to the results of the study, extreme gradient boosting shows the best performance.

Key Words: Click Prediction, Decision Trees, Random Forest, *k*-Nearest Neighbor, Extreme Gradient Boosting, Logistic Regression

ÖZET

MAKİNE ÖĞRENİMİ ALGORİTMALARI İLE REKLAM TIKLAMA TAHMİNLEME

Nazlı Tuğçe Uncu

Proje Danışmanı: Asst. Prof. Dr. Hande Küçükaydın

OCAK, 2021, 28 sayfa

Çevrimiçi reklamcılık, işletmelerin gelirlerini arttırmak için önemli bir potansiyele sahiptir. Çevrimiçi reklam kampanyalarının başarısını ölçen en önemli metriklerden biri, kampanyanın toplam gösterilme sayısının tıklama sayısına oranını gösteren tıklama oranıdır (TO). Bu nedenle reklamların tıklama oranlarını arttırmayı amaçlayan tıklama tahminleme sistemleri işletmeler için önemli olmaya başlamıştır. Yine aynı nedenlerle, reklamın kullanıcılar tarafından tıklanıp tıklanmayacağı, hem endüstriden hem de akademiden araştırmacıların ilgisini çekmektedir.

Bu projede Avazu şirketinin tıklama kayıtları veri kümesi kullanılarak tıklama tahminlemesi yapılmıştır. Tahminlemeden önce, veri kümesinin ön işleme sürecinde, veri kümesindeki yüksek kardinaliteye sahip kategorik öznitelikler ve dengesiz veri dağılımı için kullanılan yöntemler ve sonrasında seçilen öznitelikler ayrıntılı bir şekilde incelenmiştir. Tıklama tahmini için karar ağacı, rastgele orman, *k*-en yakın komşuluk, ekstrem gradyan arttırma ve lojistik regresyon sınıflandırma algoritmaları kullanılmıştır. Çalışmanın sonucuna göre en yüksek performansı gösteren algoritma, ekstrem gradyan artırma olmuştur.

Anahtar Kelimeler: Tıklama Tahminleme, Karar Ağacı, Rastgele Orman, *k*-En Yakın Komşuluk, Ekstrem Gradyan Artırma, Lojistik Regresyon

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Display advertising is an online advertising format in which banner ads such as text, image, video, audio, and motion appear in specifically designated areas of a website or an app. Display advertising spending reached 161 billion U.S. dollars in 2019 worldwide (Guttmann, 2020). This dramatic increase in online advertisement motivates to better estimate the Click Through Rate (CTR) in order to determine whether spending money on digital advertising is worth or not. A CTR is the ratio of the number of times that an ad is clicked to the number of times it is displayed. The number of times an ad is displayed is also called impressions. A higher CTR means that the ad becomes successful in generating an interest. High CTR may be affected by different variables such as banner position, display size, the website showing the ad and so on. In addition, in many online advertising systems such as Google Ads, FB Ads, the ad ranking strategy depends on the product of CTR and bid, where the bid shows how much businesses are willing to pay for a specific action, i.e. the ad click. If the businesses know the expected CTR before they release the advertisement campaign, they can maximize the revenue and maintain a desirable user experience.

To determine ad displacements for each user/device/platform combination is a challenge for the advertising system. Predicting ad clicks attracts lots of attention from both academia and industry. There are a few public datasets available and over the past years, some articles have been published on 'CTR prediction' that aims to make improvements in predicting whether the user will click the advertisement.

1.1. Literature Review on CTR Prediction

Whether the user will click the advertisement that is displayed to them or not is a classification problem. There are many proposed models in this field such as logistic regression, tree-based models, factorization machine-based models, and deep learning-based models.

For CTR prediction, it is important to pay attention to feature interactions behind users' clicks. For example, Guo et. al. (2017) suggest an interaction between app category and time-stamp, since it finds out that food delivery apps are often installed at meal-time.

Due to the nature of CTR prediction datasets, most of the studies focus on feature engineering.

He et al. (2014) use a massive volume of Facebook's ads data for predicting clicks on ads at Facebook (FB) and point out that the best performing model is a combination of decision trees and logistic regression. Although the article does not explicitly explain the features, it does underline that historical features which depend on previous interaction of a user, have more explanatory power than contextual features. This exclusively depends on current information regarding the context in which an ad is to be shown. Overall, the article sums up that combination of decision trees and logistic regression exceeds either of these methods on its own by over 3%, which is an improvement that has a significant impact on the overall system performance.

Field-aware Factorization Machines (FFMs) won two competitions for CTR prediction hosted by Avazu and Criteo and have been considered one of the best performing models (Zhuang et al., 2016). According to the team who won the contest, the feature engineering and the ensemble methods are the keys for their solution which outperformed the other FFMs based solutions (4 Idiots' Approach for Click-through Rate Prediction).

Pan et al. (2018) state that although FFMs have been among the best performing methods for the datasets in which all features are categorical and the huge number of parameters of FFM models causes memory problems and thus inefficiency for the real-world production system. Therefore, they propose a new model which is called Field-weighted Factorization Machines (FwFMs). The article clearly explains why FwFMs is the best answer to the CTR prediction dataset's 'multi-field categorical data' and also discusses FwFMs structure and experiments. In the end, the research proves that when using the same number of parameters, FwFMs can achieve consistently better performance than FFMs.

In addition, Vasiloudis et al. (2019) propose a new Gradient Boosted Trees (GBT) based algorithm called "Block-distributed Gradient Boosted Trees". Although the GBT algorithm already has a proven success in the field of CTR prediction and of learning-to-rank due to their accuracy and scalability, it has some drawbacks when it comes to high dimensional data with millions of features. While distributed GBT algorithm use row distribution method, block-distribution GBT algorithm involves both the row and column

dimensions. Block-distributed GBT algorithm achieves the training time reduction for high dimensional data. This study also uses the Avazu dataset in order to evaluate the hypothesis of the study. The same dataset is converted into binary classification form with one million features.

On the other hand, deep neural network based models have also been proposed for CTR prediction in the last few years. Zhou et al. (2017) build a new model named Deep Interest Network (DIN) that predicts a user's action (to click or not to click) by taking into account the user behaviours which has higher relevance to the given ad while calculating the vector of user interest. The study outperforms other deep learning based models in which user features are calculated as a fixed-vector regardless of the given ad's context.

Another deep learning model proposed for CTR prediction is DeepFM which is a factorization based neural network. In the study, Guo et al. (2017) state that DeepFM integrates the architecture of factorization machines, which models low-order feature interactions, and deep neural networks, which models high-order feature interactions. Therefore, DeepFM learns both low and high feature interactions. It does not need pre-training and feature engineering as well. The effectiveness and efficiency of the model outperforms the state-of-art models given in the article.

2. ABOUT THE DATA

The dataset used in the project is provided by Avazu, a leading advertising platform. The dataset is publicly available online at https://www.kaggle.com/c/avazu-ctrprediction/data. It contains rich information on the ads that were displayed for a period of 11 days. Avazu provides 'train' & 'test' files for participants. The training file includes 10 days of click data ordered chronologically, while the testing file includes 1 day of click data for testing the model predictions.

The training data includes around 40 Million rows and 6 GB of uncompressed data and it contains 23 different features. The target feature is the 'click' which takes binary values. The target takes the value 0, if an add is not clicked and 1, otherwise. 9 features of the dataset are anonymous and all of them are categorical. The only non-categorical feature is 'hour' which is of datetime data type. Table 1 displays the features of the dataset and their definitions.

Table 1: Features of the data

Feature	Definition
id	Ad identifier
click	Response variable, integer (binary)
hour	DateTime Format
C1	Anonymized categorical variable
banner_pos	The display position of the ad in the screen
site_id	The unique identifier of the site that the ad was displayed.
site_domain	The domain information of the website
site_category	The category information of the website
app_id	The unique identifier of the app that the ad was displayed

app_domain	The domain information of the app that the ad was displayed
app_category	The category information of the app that the ad was displayed
device_id	The unique identifier of the device that the ad was clicked
device_ip	The ipv4 address of the device on which the ad was clicked
device_model	The model of the device
device_type	The type of the device
device_conn_type	The connection type of the device
C14-C21	Anonymized categorical variables

The feature definitions are obtained from the Avazu's Kaggle competition website. The features of this dataset can be classified into following categories:

- *Target feature:* click
- *Site features:* site_id, site_domain, site_category
- *App features:* app_id, app_domain, app_category
- *Device features:* device_id, device_ip, device_model, device_type, device_conn_type
- *Anonymized categorical features:* C1, C14-C21

2.1. Feature Engineering

As it is mentioned, the training set contains over 40 millions of records and 1 million of them are randomly sampled to process them locally. The new sampled dataset contains values from 21/10/14 00:00:00 to 30/10/2014 23:00:00.

There is no outlier detection/removal, since all our features are categorical except 'click' and 'hour'. Also, no missing value is found in the dataset.

Two new columns are generated from the 'hour' column, where one shows the hour of the day which consists of 24 unique categorical values from 0 to 24 and the other one indicates the day of the week and includes 7 unique categorical values.

After generating these new columns, the total number of features in the dataset becomes 26, including the target feature ‘click’.

2.2. Exploratory Data Analysis

Some of the categorical features have high amounts of distinct values. Table 2 shows the number of distinct values that features can take.

Table 2: The number of distinct values that categorical features can take

Feature	Number of Distinct Values
id	1000000
C1	7
banner_pos	7
site_id	2675
site_domain	2886
site_category	21
app_id	3138
app_domain	194
app_category	26
device_id	150102
device_ip	554787
device_model	5166
device_type	5
device_conn_type	4
C14	2257
C15	8
C16	9
C17	420
C18	4

C19	66
C20	165
C21	60
hour_of_day	24
day_of_week	7

The click percentage of the dataset is approximately 16% as can be seen in Figure 1. Therefore, the dataset displays some imbalance, since the majority of data instances are not clicked.

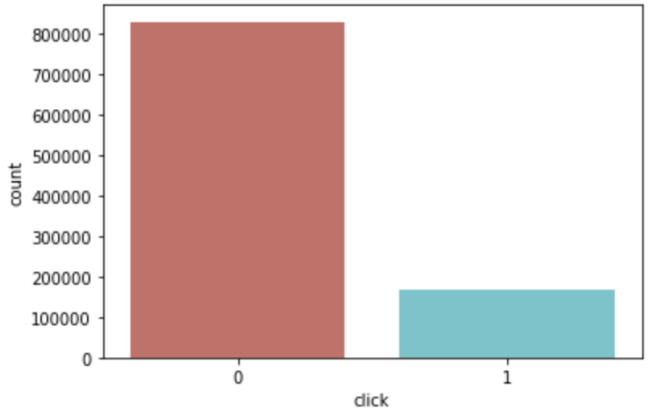

Figure 1: Click Distribution

Figure 2 plots the change in the number of clicks in time. There is a similar pattern in the change; that is to say, the number of clicks increase towards noon and then decrease during the nighttime. However, on October 22, 2014 and on October 28, 2014, there are peak points.

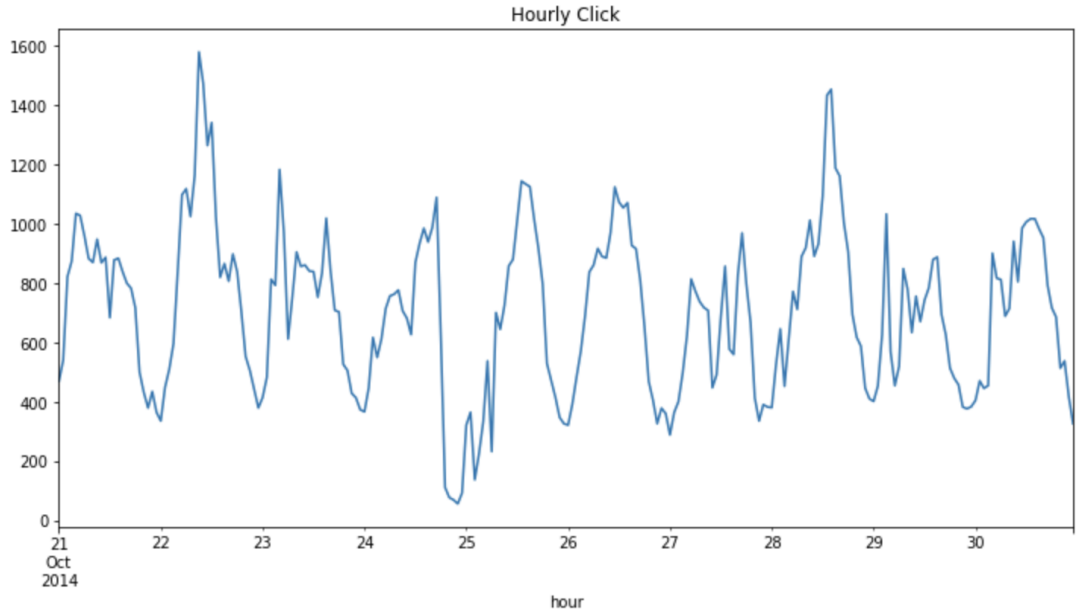

Figure 2: Number of clicks change in time

To understand if these peaks are related to the day of the week, Figure 3 which shows click trends by day of the week is plotted. According to this figure, on Tuesdays and Wednesdays, the number of clicks are the highest. The peak days, October 22, 2014 and October 28, 2014, are Wednesday and Tuesday, respectively.

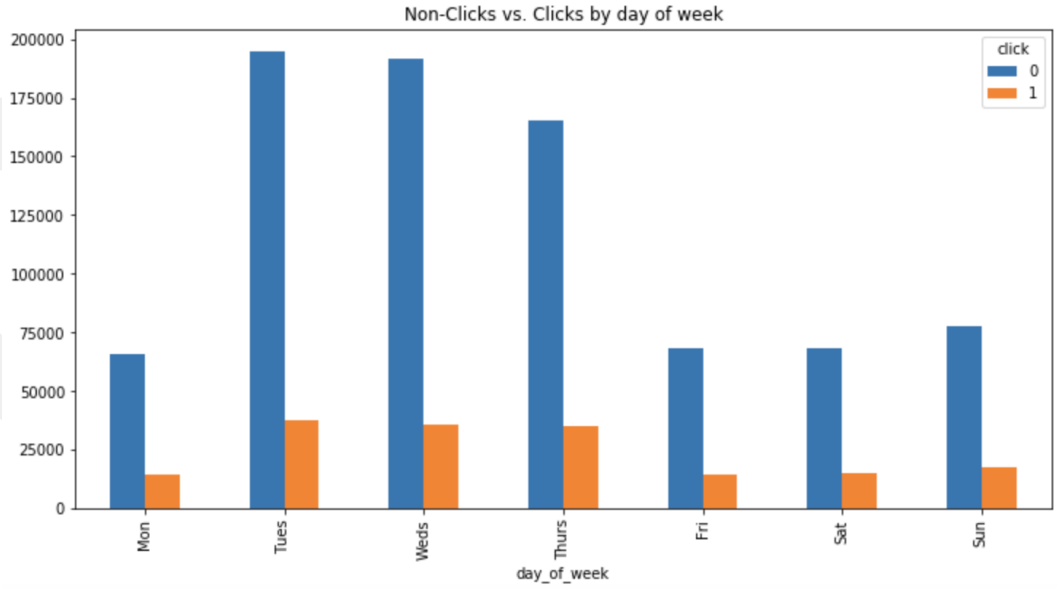

Figure 3: Click distribution by day of the week

On the other hand, the click ratio by day of the week is calculated as the ratio of total number of clicks to total number of impressions of all ads. Figure 4 below indicates that Mondays, Saturdays and Sundays have the highest click ratios.

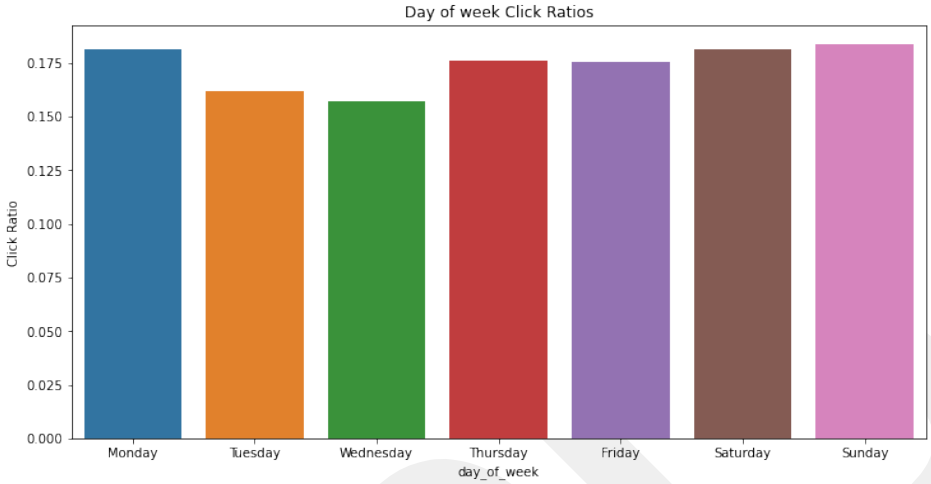

Figure 4: Click distribution by day of the week

Figure 5 represents hourly change of clicks and non-clicks. When we closely examine this figure, we see that the number of clicks decreases from evening to midnight. At 9 am and 1 pm, the data has the highest number of impressions.

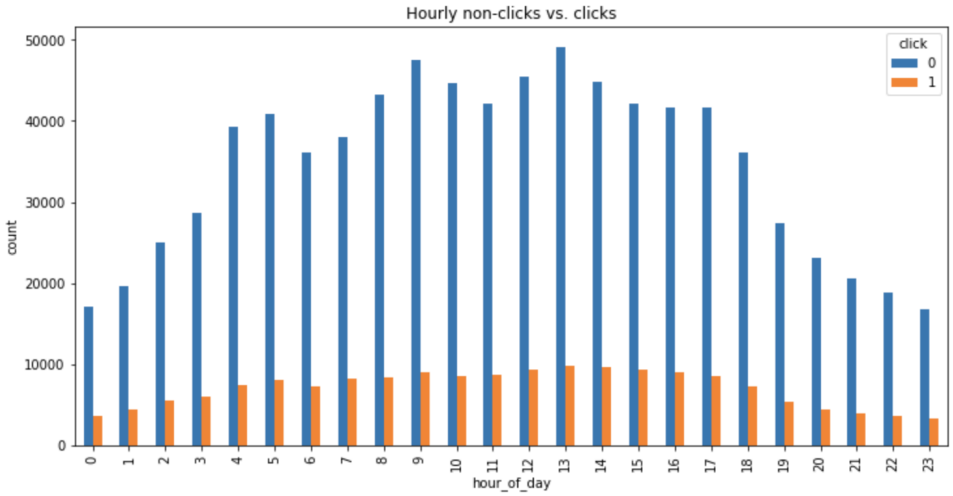

Figure 5: Hourly clicks vs non-clicks

On the other hand, around 1 am and 3 pm, the data has the highest click ratio as can be seen from Figure 6 below.

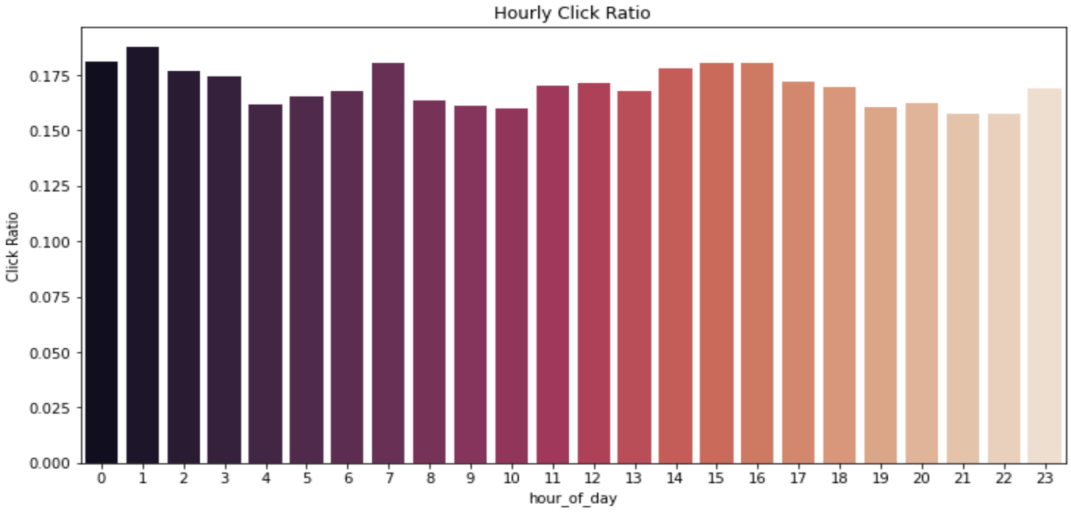

Figure 6: Click ratio change by time in hours

To understand the effect of banner position on the performance of the display ads, click distribution plot by banner position is plotted by Figure 7. The ads are displayed at seven different positions range from 0 to 7. We are not given the information of the exact placements of these positions on the page so the positions range from 0 to 7 are anonymous.

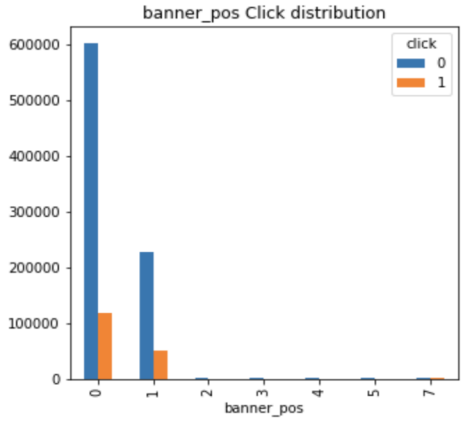

Figure 7: Click distribution by banner position

As can be seen from Figure 7, most of the ads are placed at position 0 and position 1. When we inspect the click ratios according to banners' positions, we see that position 7 has the highest click ratio although a very limited number of ads are displayed in this position.

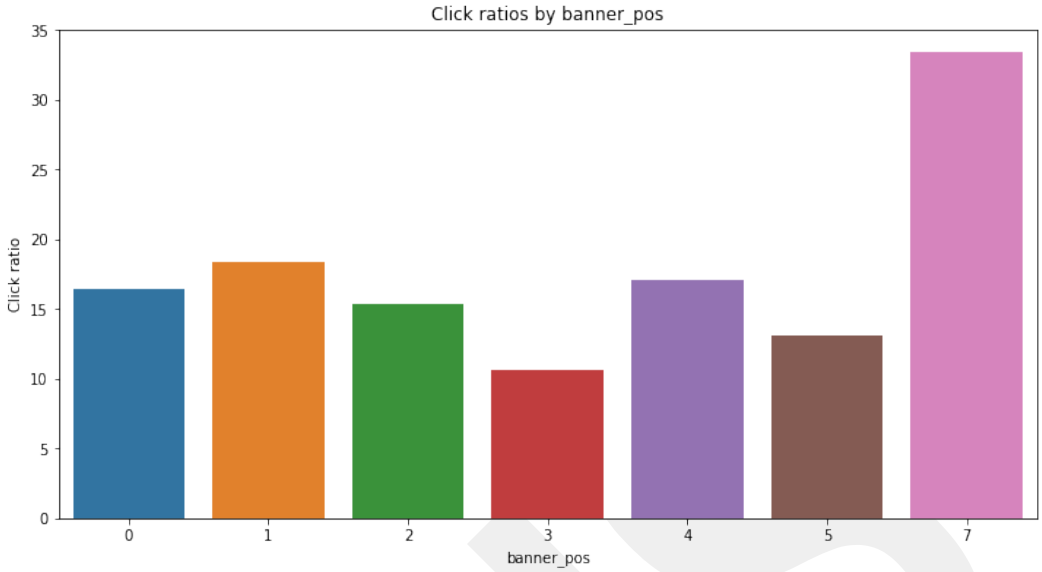

Figure 8: Click ratios by banner position

The anonymous categorical variable 'C1' has 7 different values. 1005 (an anonymous C1 value) has the highest number of impressions as shown in Figure 9.

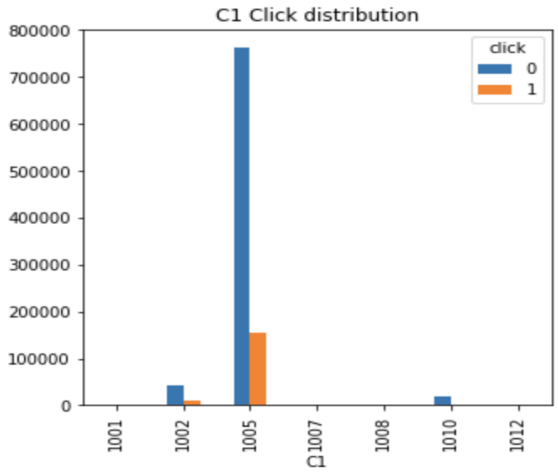

Figure 9: Click and Impression distribution of C1

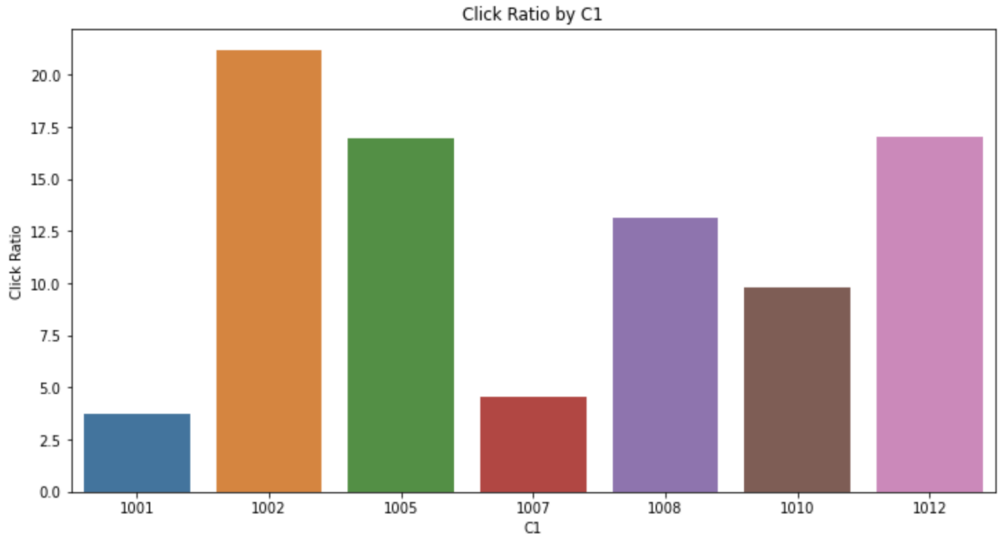

Figure 10: Click ratios by C1

In addition, as can be seen from Figure 10, the attribute values 1002, 1012 and 1005 have the highest click ratio values.

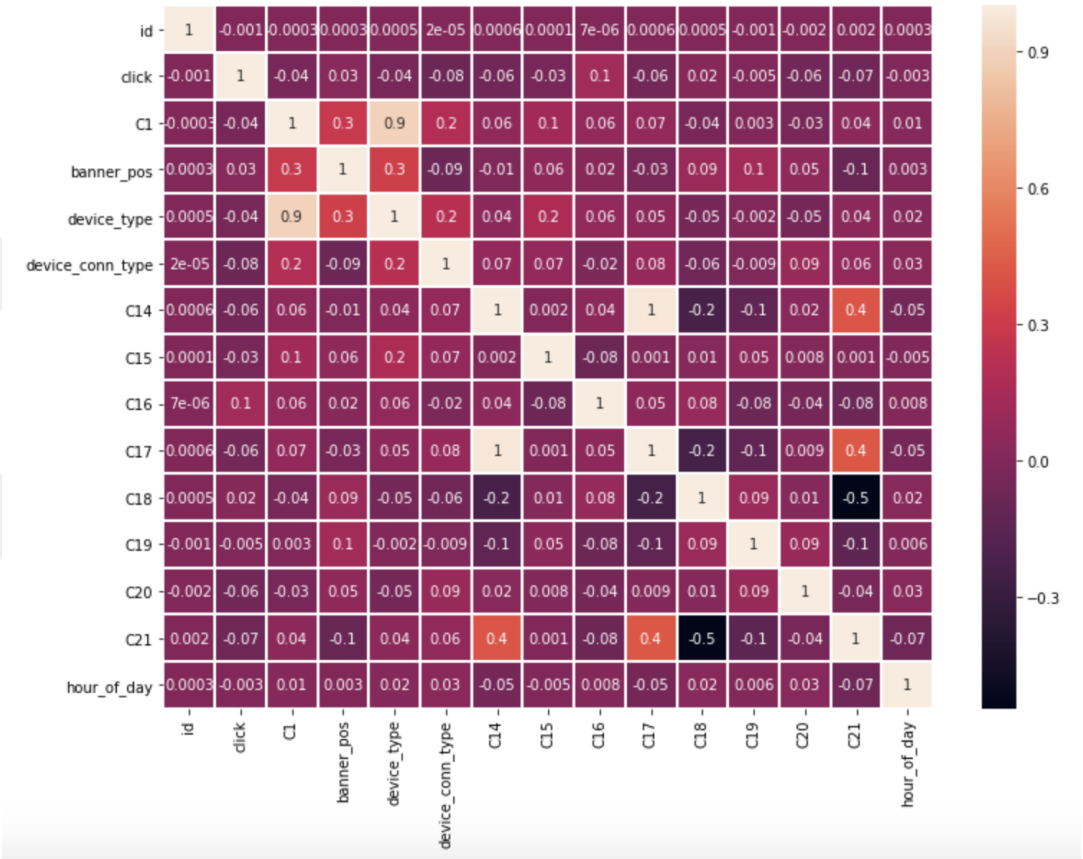

Figure 11: Correlation matrix

Figure 11 shows the correlation matrix which indicates correlation coefficients between the variables. The correlation between C14 and C17 is 1. This means that they are highly correlated with each other. So, they can be treated as the same feature. On the other hand, C21 and C18 negatively correlated with each other with -0.5 coefficient. This is the second-highest correlation in the dataset between the features. In addition, the correlation coefficient between id and other variables is quite low when comparing other correlation combinations. The correlation between C15 and id is 0.0001 which is the lowest correlation coefficient in the dataset.

3. PROJECT DEFINITION

3.1. Problem Statement

Due to the massive growth in online advertising spendings, decreasing misclassification rate in spendings has become significant in the industry. Since CTR is one of the most important factors that affects revenue and spendings, click prediction has become compelling for the studies that aim to predict if an ad will be clicked or not under the given conditions. Higher accuracy in the prediction of whether a given ad will be clicked can help businesses set the right set of attributes and reduce the cost of the advertising accordingly.

3.2. Project Objectives

The aim of this project is to predict the clicks on a binary scale, 1 for click and 0 for no click, by using machine learning algorithms. Since the dataset that includes advertisement click logs consists of high cardinality categorical features, in order to achieve the optimum performance and accuracy in prediction, this research concentrates on trying different encoding techniques and also the feature selection accordingly in order to reduce the high dimensionality and also on finding the best performing algorithm.

3.2. Project Scope

The public advertisement dataset that consists of click logs provided by Avazu is used in this project. The dataset contains a lot of anonymous categorical variables that make this dataset hard to interpret. Having anonymous features set serious boundaries for feature engineering and feature selection. He et al. (2014) indicate that historical features which show the previous interaction of user have more explanatory power than contextual features. On the Avazu dataset, all the non-anonymous features are contextual features. Therefore, it is hard to interpret the dataset for the purpose of feature engineering and selection. In addition to anonymity, there is a lot high-cardinality features that cause a serious challenge in terms of performance for many classification and regression algorithms which require numerical inputs. To avoid these issues, two different encoding techniques are used to convert categorical features and feature selection is performed based on the encoding technique used in the dataset.

For click prediction, five different classification methods including decision trees, random forest, *k*-nearest neighbor, extreme gradient boosting, logistic regression are used and their results are compared.

4. METHODOLOGY

The dataset sampled from Avazu's train data has 1 million rows and 24 features. Other than the 'hour' feature which is a DateTime value and 'click' which is the target feature and has integer data type, all of the features in the dataset are categorical. Some of the categorical features have a high amount of distinct values which is also called 'high cardinality' (Moeyersoms and Martens, 2016). For example, 'device_ip' has around 554,000 distinct values which make this feature (very) high cardinal. For statistical analysis, categorical variables are usually converted into numerical format (Cerda and Varoquaux, 2019).

4.1. Categorical Feature Encoding

In order to convert categorical features, we try feature hashing and one-hot encoding, respectively. As a result, two different encoded datasets are created.

4.1.1 Feature Hashing

In order to overcome the problems that high cardinality causes, an alternative encoding technique that deals with large-scale datasets is needed. Based on the literature review on Avazu dataset, Feature Hashing is a common technique that is used to convert categorical features since it tries to compress feature representations (Seger, 2018). For example, the winner of the Avazu CTR Prediction Kaggle Challenge ("4 Idiots' Approach for Click-through Rate Prediction", n.d.), also the top 5th solution (Efimov, 2015), and the research paper that proposes field-weighted Factorization Machines for Click-Through Rate Prediction in Display Advertising use feature hashing (hashing trick) on categorical features (Pan et al., 2018). Feature hashing or the hashing trick uses a hash function to reduce original high-dimensional space into low-dimensional space. In this method, a hash function maps the features to hash keys and aggregates features' counts (Caragea et al., 2012).

As a result, feature hashing technique is used on the dataset in order to convert categorical features.

4.1.2 One-Hot Encoding

One-hot encoding is a popular encoding technique that creates a vector representation of categorical variables. It uses dummy variables to encode categorical features (Liu, 2019). On the other hand, the one-hot encoding technique is not feasible when high cardinality features exist on the data because high cardinality creates high-dimensional feature vectors (Seger, 2018). High dimensionality feature vectors lead to memory and computability concerns for machine learning models (Seger, 2018). Given the fact that this project is being conducted from a local computer which has limited memory and performance, applying one-hot encoding to this dataset without feature selection would cause serious performance issues. Therefore, in this study, one-hot encoding technique is only used when feature selection is applied on the dataset.

4.2 Undersampling for Imbalanced Data

Another issue that should be handled during the preprocessing is imbalanced data. The ratio of the number of ads that are not clicked to the number of ads that are clicked is 4.9. Imbalanced data can lead to poor accuracy and recall on the minority class which is generally the class of interest (Pozzolo et al., 2015). Undersampling is a method for handling the imbalance data. Undersampling balances a dataset by removing observations from the majority class at random (Brownlee, 2016). Since the dataset on hand is imbalanced, the undersampling method is used in this project as a preprocessing. To note that, without undersampling, the recall, precision and F1 scores of models are extremely low. To improve the evaluation scores, undersampling is applied to the dataset first and then categorical features are converted via encoding techniques.

4.3 Feature Selection

Feature selection is a process to choose the features that are most relevant to the machine learning problem. It helps the model to achieve its maximum performance with less measurement effort (Liu and Motoda, 1998).

Before selecting the relevant features, the dataset is sampled again to 100,000 rows to be able to process it easily. For encoding, the first approach is to apply feature hashing technique to the data in order to convert categorical values. Since feature hashing is a fast dimensionality reduction technique, we select and drop manually only a few features first.

Features named 'hour', 'id', 'device_ip' are removed. The ‘id’ feature which is a unique ad identifier, ‘device_ip’ which identifies the device are dropped since they are non-informative in the context of click prediction. On the other hand, since ‘hour_of_the_day’ and ‘day_of_the_week’ features are already generated from the ‘hour’ column, the ‘hour’ feature is also dropped. Then categorical features are converted via feature hashing which performs well when dealing with large-scaled datasets. Since the initial purpose is to see the dataset’s learning capability before making any further feature selection, feature hashing technique gives the best fitting solution to the problem.

To compare encoding techniques performance with each other, one-hot encoding is also applied to the same dataset. However, to apply one-hot encoding, more features need to be dropped. Otherwise, high dimensionality would cause some problems regarding the computation.

The following features are removed from the dataset due to being non-informative or having high-cardinality: 'hour', 'id', 'device_ip', 'device_conn_type', 'site_id', 'site_domain', 'app_id', 'app_domain', 'device_id', 'device_model', 'C14', 'C17', 'C19', 'C20', 'C21'. Although C1, C15, C16 and C18 are anonymous categorical features, they are not removed from the dataset since they do not have high-cardinality. The reduced dataset includes features ‘click’ (target feature), ‘C1’, ‘banner_pos’, ‘site_category’, ‘app_category’, ‘device_type’, C15, C16, C18 ‘hour_of_day’ and ‘hour_of_day’.

4.4 Prediction Methods

The machine learning models in this project include Extreme Gradient Boosting (xgboost), Random Forest Classifier, Decision Tree, *k*-Nearest Neighbor and Logistic Regression.

Decision tree is a supervised learning method that is used to solve classification and regression problems. It is a tree-like graph that illustrates all of the possible decision alternatives, their probabilities and outcomes in a sequential diagram (Liu, 2019). Decision trees are easy to interpret and require little data preparation. However, they are prone to overfitting (Kotu and Deshpande, 2015). Starting from the root node, the training data is split into subsets and this process continues recursively until the subset has the same class label or splitting no longer improves the class purity of the subset (Liu, 2019).

Random forest is an ensemble method that contains multiple decision trees. The algorithm creates decision trees from a randomly selected subset of training data and calculates the output by averaging the predictions of each tree. Random forest overcomes decision tree's overfitting issue by averaging the results of decision trees (Reinders et al., 2019).

The *k*-nearest neighbor (KNN) method is another popular supervised classification algorithm (Rajaguru and Prabhakar, 2017). In order to classify an unknown object from a test set, KNN method finds *k*-nearest neighbors of the unknown object and then uses the class labels of these neighbors to make a summarized prediction (Bao et al., 2004). KNN is a simple and effective method. However, its high classification complexity and high memory requirement are significant drawbacks (Dwulit and Szymański, 2012).

Logistic regression is a robust and flexible method to solve binary classification problems (Seufert, 2014). It uses the logistic sigmoid function to transform its output to return values between 0 and 1 (Liu, 2019). Since its output returns a probability value, the estimated probabilities which are greater than 0.5 can be mapped into the positive class labeled as 1; other probabilities can be mapped into the negative class labeled as 0 (Géron, 2017).

Gradient Boosting is a very popular implementation of boosting method. Boosting method is an ensemble method that trains weak learners sequentially such that at each iteration weak learners try to correct the errors of its predecessor model (Géron, 2017). In gradient boosting, each learner is trained according to the residuals of its predecessor model. On the other hand, Extreme Gradient Boosting (xgboost) is a popular implementation of gradient boosting framework designed for efficiency and scalability (Chen and He, 2020). Unlike gradient boosting trees which are built sequentially, extreme gradient boosting methods build trees in parallel. It reduces overfitting and controls model complexity (Quinto, 2020).

After trying different encoding techniques and applying the models on the datasets, we try to improve the performance of the best performing model by implementing necessary hyperparameter tuning steps.

5. RESULTS

To predict the clicks on a binary scale, a random sample from Avazu dataset (the original training file) is split into train and test sets. The train set is trained with Decision Tree, Random Forest Classifier, kNN, Extreme Gradient Boosting (xgboost) and Logistic Regression models. Then, the test set is used to measure the predictive performance of the model. The evaluation metrics are calculated for the test set.

The dataset's categorical features are encoded using two different encoding techniques. As a result, two different datasets are used to train the models:

i) Dataset with features using feature hashing

ii) Dataset with features converted using one-hot encoding

As it is mentioned before, the features 'hour', 'id' and 'device_ip' are removed from the dataset whose categorical features converted using feature hashing. In addition, the following features are removed from the one-hot encoded dataset: 'hour', 'id','device_ip', 'device_conn_type', 'site_id', 'site_domain', 'app_id', 'app_domain', 'device_id', 'device_model', 'C14', 'C15',' C16', 'C17', 'C18', 'C19', 'C20', 'C21'.

5.1 Results of the models applied on the hashed dataset

The models are applied to the dataset which is encoded using feature hashing. The accuracy, recall, precision and F1 scores of each model are given in Table 3.

Table 3: Evaluation metrics for the models applied on the hashed dataset

	Decision Tree	Random Forest Classifier	*k*-Nearest Neighbor	Extreme Gradient Boosting	Logistic Regression
Accuracy	0.57	0.63	0.59	**0.64**	0.57
Recall	0.56	0.62	0.61	**0.69**	0.62
Precision	0.57	**0.63**	0.59	**0.63**	0.56
F1 Score	0.57	0.62	0.60	**0.66**	0.59

According to results given in Table 3, the best performing algorithm is extreme gradient boosting with 0.64 accuracy score. It is important to state that Random Forest algorithm with 0.63 accuracy score can be considered as performing similarly to extreme gradient boosting. On the other hand, logistic regression and decision tree methods has the lowest scores.

5.2 Results of the models applied on the one-hot encoded dataset

The models are also applied on the second dataset which is encoded using one-hot encoding technique. The evaluation metrics of the models are given in Table 4.

Table 4: Evaluation metrics for the models applied on the one-hot encoded dataset

	Decision Tree	Random Forest	K-Nearest Neighbor	Extreme Gradient Boosting	Logistic Regression
Accuracy	0.59	0.60	0.57	**0.62**	0.61
Recall	0.56	0.61	0.59	0.65	**0.68**
Precision	0.60	0.59	0.57	**0.61**	0.59
F1 Score	0.58	0.60	0.58	**0.63**	**0.63**

Although extreme gradient boosting has slightly higher accuracy score than the others, it is fair to say that accuracy scores of decision tree, random forest, extreme gradient boosting and logistic regression algorithms are almost the same. Thus it can be concluded that all models mentioned perform very similarly in terms of the accuracy. In addition, in terms of accuracy, precision and F1 score metrics, extreme gradient boosting performs slightly better than the others, but logistic regression algorithm has a little higher recall score than the other algorithms. High recall models are likely to return most of the relevant results, but these results do not have be accurate (Vermeulen, 2018). Although having higher recall is good, we need to analyze both recall and precision, as well as accuracy when deciding the best performing model. Therefore, after evaluating all of the evaluation metrics together, it can be concluded that extreme gradient boosting can be

considered as the model which slightly outperforms the others, similar to the results shown in Table 3.

When comparing the results given in Table 3 and Table 4, it can be observed that the models which are trained on the hashed dataset outperforms the models trained on the one-hot encoded dataset except the logistic regression model. Overall, extreme gradient boosting under feature hashing method performs slightly better in terms of evaluation scores.

5.3 Results of grid search applied on the best performing algorithm

In order to find the best combination of parameters of the extreme gradient boosting algorithm, Grid Search is applied to the Xgboost model (which uses feature hashing method). The results obtained are given in Table 5:

Table 5: Evaluation metrics for the Xgboost after hyperparameter tuning

Accuracy	0.67
Recall	0.72
Precision	0.68
F1 Score	0.69
AUC	**0.67**

As can be seen from Table 5, hyperparameters tuning does not bring dramatic increase in the results.

Feature importance resulting from Xgboost is given in Figure 12. The most important features are device_model and site_domain, whereas C15 and device_type are the least important features.

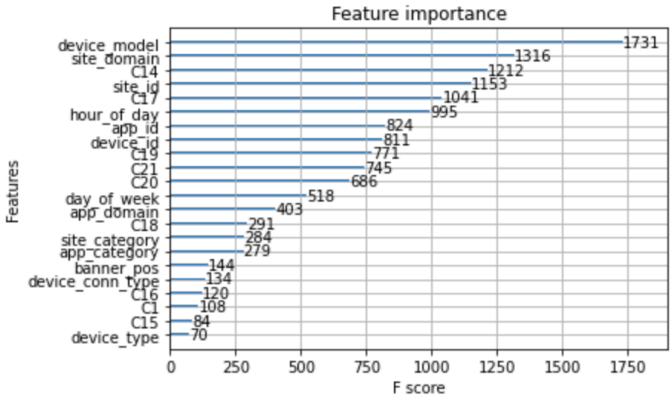

Figure 12: Feature Importance from Extreme Gradient Boosting

When comparing the results with the results obtained in similar studies, the area under the ROC curve (AUC) value of the best performing model in this research is quite low. To measure the predictive performance of the model, most of the studies in the literature that use click logs datasets employ AUC (Guo et al., 2018). For example, the AUC metric of Field-weighted Factorization Machines (FwFMs) for CTR prediction study that uses Criteo and Oath's click logs and the AUC of another benchmark study on sequential click prediction with RNNs are higher than the AUC in our study (Pan et al., 2018; Zhang et al., 2014). On the other hand, Ramanathan (2019) deals with the problem in a similar way and obtains a higher accuracy and better AUC results which are 0.83 and 0.84 for xgboost model, respectively. However, the recall and the precision values of the same model are significantly lower than the ones' in our project. High accuracy score with low recall and precision values may indicate overfitting and Ramathan (2019) also mentions that high accuracy but low precision and recall scores result from an imbalance dataset.

One possible reason for the poor results could be the feature engineering part. This study does not generate any user-associated feature to be used in the models. However, as it is indicated before, features that indicate a user's previous interaction with ads are significantly higher importance in click prediction (He et al., 2014). In addition, the machine learning algorithms used in this study do not properly model the interaction between features. However, it is crucial to learn complex feature interactions behind user click behaviors. Overall, having complete access to all the features with no anonymous values, generating proper user-related features and also taking into account the feature interactions can significantly improve the results.

5. CONCLUSION

The aim of this project is to study click prediction. From preparation of dataset to model application, each step is discussed in detail. First of all, feature engineering is performed and two new features are generated from 'hour' feature. Preparing and processing a dataset which consists of high cardinality categorical features is one of the key challenges in the project. To deal with high cardinality, feature hashing and one-hot encoding are used separately and consequently, two different encoded datasets are created. Decision tree, random forest, *k*-nearest neighbor, extreme gradient boosting and logistic regression methods are applied on these datasets and the models' predictive performance are measured accordingly.

The best performing model and encoding method turns out to be the extreme gradient boosting with feature hashing technique. However, it does not bring an improvement over the current click prediction studies.

This study uses a public dataset with anonymized values. The anonymized features which may indicate user-related values had been masked by Avazu company before the dataset was released for public use. Therefore, interpreting and evaluating the dataset and development of the models are done without knowing what anonymized values stand for. Therefore, this might have an effect on the overall performance.

This study also supports the importance of feature interaction and having user-related data which indicate the previous interaction of an ad or a user because without having these two important concepts, our models cannot be improved. Therefore, future work can be directed towards investigating feature interaction and feature engineering on historical user features.

REFERENCES

1. 4 Idiots' Approach for Click-through Rate Prediction (n.d.). NTU CSIE. Retrieved September 30, 2020 from https://www.csie.ntu.edu.tw/~r01922136/slides/kaggle-avazu.pdf
2. Bao, Y., Ishii, N., Du, X. (2004). Combining Multiple *k*-Nearest Neighbor Classifiers Using Different Distance Functions. In Yang Z.R., Yin H., Everson R.M. (Eds.) In *Lecture Notes in Computer Science*, 3177, 634-641. Springer, Berlin, Heidelberg https://doi.org/10.1007/978-3-540-28651-6_93
3. Brownlee, J. (2016). A Gentle Introduction to XGBoost for Applied Machine Learning. *Machine Learning Mastery* https://machinelearningmastery.com/xgboost-for-imbalanced-classification/
4. Caragea, C., Silvescu, A., Mitra, P. (2012). Protein sequence classification using feature hashing. *Proteome science*, 10(S1), 538-545. BioMed Central.
5. Cerda, P., Varoquaux, G. (2020) Encoding high-cardinality string categorical variables. *IEEE Transactions on Knowledge and Data Engineering,* arXiv:1907.01860*,* https://arxiv.org/abs/1907.01860
6. Chen, T., He, T. (2020, September 02). *xgboost: eXtreme Gradient Boosting*. cran.r-project
7. Dwulit, M., Szymański, Z. (2012). Dwulit's Hull as Means of Optimization of kNN Algorithm. *Human–Computer Systems Interaction: Backgrounds and Applications 2,* 345-358.
8. Efimov, D. (2015, June). Click Through Rate Prediction – Top-5 Solution for Avazu Contest. In *Thirteenth International Seminar* (p.76)
9. Géron, A. (2017). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.
10. Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 1725-1731. https://doi.org/10.24963/ijcai.2017/239

11. Guttmann, A. (2020). Global internet advertising spending 2007-2022, by format. *Statista* https://www.statista.com/statistics/276671/global-internet-advertising-expenditure-by-type/
12. He, X., Bowers, S., Candela, J. Q., Pan, J., Jin, O., Xu, T., Liu, B, Xu, T., Shi, Y., Atallah, A., & Herbrich (2014). Practical Lessons from Predicting Clicks on Ads at Facebook. *Proceedings of 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining - ADKDD14*, 1-9. https://doi.org/10.1145/2648584.2648589
13. Juan, Y., Zhuang, Y., Chin, W., & Lin, C. (2016). Field-aware Factorization Machines for CTR Prediction. *Proceedings of the 10th ACM Conference on Recommender Systems,* 43-50. https://doi.org/10.1145/2959100.2959134
14. Kotu, V., Deshpande, B., (2015). *Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner*. Morgan Kaufmann https://doi.org/10.1016/B978-0-12-801460-8.00004-5
15. Liu, Y. (2019). *Python Machine Learning By Example*. Birmingham, UK: Packt Publishing.
16. Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic Publ.
17. Moeyersoms, J., and Martens D. (2016) *Data Mining Tip: How to Use High-Cardinality Attributes in a Predictive Model.* KDnuggets www.kdnuggets.com/2016/08/include-high-cardinality-attributes-predictive-model.html
18. Qu, Y., Fang, B., Zhang, W., Tang, R., Niu, M., Guo, H., Yu, Y., & He, X. (2019). Product-Based Neural Networks for User Response Prediction over Multi-Field Categorical Data. *ACM Transactions on Information Systems*, *37*(1), 1–35. https://doi.org/10.1145/3233770
19. Quinto, B. (2020). *Next-Generation Machine Learning with Spark : Covers XGBoost, LightGBM, Spark NLP, Distributed Deep Learning with Keras, and More*. Apress L. P.
20. Pan, J., Xu, J., Ruiz, A. L., Zhao, W., Pan, S., Sun, Y., & Lu, Q. (2018). Field-weighted Factorization Machines for Click-Through Rate Prediction in Display

Advertising. *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW 18*. 1349-1357. doi: 10.1145/3178876.3186040

21. Pozzolo, A. D., Caelen, O., & Bontempi, G. (2015). When is Undersampling Effective in Unbalanced Classification Tasks? *Machine Learning and Knowledge Discovery in Databases Lecture Notes in Computer Science,* 200-215. doi:10.1007/978-3-319-23528-8_13
22. Ramanathan, M. (2019). An Ensemble Model for Click Through Rate Prediction(Unpublished Master's Thesis). San Jose State University, USA. https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1696&context=etd_projects
23. Reinders, C., Ackermann, H., Yang, M., Rosenhahn, B., (2019). Learning Convolutional Neural Networks for Object Detection with Very Little Training Data. *Multimodal Scene Understanding Algorithms, Applications and Deep Learning,* 65-100. Academic Press https://doi.org/10.1016/B978-0-12-817358-9.00010-X
24. Rajaguru, H., Prabhakar, S. (2017). *KNN Classifier and K-Means Clustering for Robust Classification of Epilepsy from EEG Signals. A Detailed Analysis.* Anchor Academic Publishing
25. Seger, C. (2018) *An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing* (Unpublished master's thesis) KTH Royal Institute of Technology, Sweden
26. Seufert, E. (2014). *Freemium economics*. Elsevier/Morgan Kaufmann.
27. Vermeulen, A. F. (2018). *Practical Data Science: A Guide to Building the Technology Stack for Turning Data Lakes into Business Assets*. Berkeley, CA: Apress.
28. Zhang,Y., Dai, H., Xu, Chang., Feng, J., Wang, C., Bian, J., Wang, B., Liu, T. (2014). Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* 28(1), 1369-1375
29. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., ... & Gai, K. (2018, July). Deep interest network for click-through rate prediction. In *Proceedings of the 24th*

ACM SIGKDD International Conference on Knowledge Discovery & Data Mining 1059-1068