Low Level Design (LLD)

User Response Prediction System using Machine Learning Techniques

Vivek Limbad, Manali Kadam, Rueben Patil, Siddhanth Ghag

Pursuing Masters in Data Science & Business Analytics.

Written By	**Vivek Limbad**
Document Version	0.1
Last Revised Date	7 – November -2021

Contents

Abstract

It is necessary to predict profitable users who can click target ads (i.e., activity Targeting) in the advertising trade. The task selects the potential users that can connect the ads by analyzing users' clicking/web browsing data and displaying the foremost relevant ads to them. This project presents an associate empirical study of the exploitation of different web of things techniques to predict whether or not an advertisement is going to be clicked or not. We tend to perform click prediction on a binary scale, one for click and zero for no click. We tend to use clicks information from advertizing.csv provided as a region of Kaggle competition as our information set. We tend to perform feature choice to get rid of options that don't facilitate improved classifier accuracy. We tend to examine information manually and conjointly use feature choice capability.

1. Introduction

Internet showcasing has taken over traditional advertising methodologies in the ongoing past. Organizations like to advertise their items on websites and web-based life stages. Be that as it may, focusing on the correct crowd is a test in online advertising. Burning through millions to show the advertisement to the group of spectators that isn't probably going to purchase your items can be expensive. In This project, we will work with the advertising information of a showcasing agency to build up an AI calculation that predicts if a specific client will tap on an advertisement. The information consists of **10** factors: **'Daily Time Spent on Site'**, **'Age', 'Area Income', 'Daily Internet Usage', 'Ad Topic Line', 'City', 'Male', 'Country', Timestamp'** and **'Clicked on Ad'**. The fundamental variable we are keen on is 'Clicked on Ad.' This variable can have two possible results: 0 and 1, where 0 alludes to the situation where a client didn't tap the advertisement, while one alludes to the situation where a client taps the advertisement. We will check whether we can utilize the other nine factors to foresee the worth 'Clicked on Ad' factor precisely. Likewise, we will play out some exploratory information investigation to perceive how 'Daily Time Spent on Site' in combination with 'Ad Topic Line' influences the client's decision to tap on the ad.

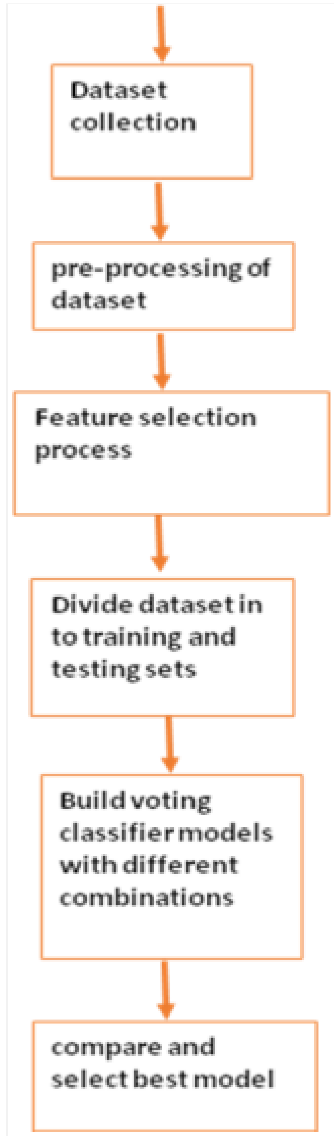
1.1 What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture



3 . Architecture Description

3.1 Data Collection

The dataset for this article can be downloaded from this Kaggle link. Unzip the downloaded zip file and place the “advertising.csv” file in your local drive. This is the file that we are going to use to train our machine learning model.

3.2 Data Pre-processing

You may have noticed that **"Ad Topic Line," "City,"** and **"Country"** are categorical columns. Let plot all the unique Values for these columns. Values for these columns.

	Ad Topic Line	City	Country
count	1000	1000	1000
unique	1000	969	237
top	Extended interactive model	Lisamouth	France
freq	1	3	9

As we can see from the table above that all the values in column "Ad Topic Line" is unique, while the "City" column contains **969** unique values out of **1000** and there are too many individual elements within these two categorical columns, and it is generally difficult to perform a prediction without the existence of a data pattern. Because of that, they will be omitted from further analysis, and the third categorical variable, i.e., "Country," has a unique element (France) that repeats nine times. Additionally, we can decide on countries with the highest number of visitors.

The table shows the **20** most represented countries in our Data Frame, and we have already seen, there are **237** different unique countries in our dataset, and no single country is too dominant. A large number of individual elements will not allow a machine learning model to exist easily valuable relationships. For that variable will be excluded too

col_0	count
Country	
France	9
Czech Republic	9
Afghanistan	8
Australia	8
Turkey	8
South Africa	8
Senegal	8
Peru	8
Micronesia	8
Greece	8
Cyprus	8
Liberia	8
Albania	7
Bosnia and Herzegovina	7
Taiwan	7
Bahamas	7
Burundi	7
Cambodia	7
Venezuela	7
Fiji	7

Next, we will analyze the **'Timestamp'** category. It represents the exact time when a user clicked on the advertisement. We will expand this category to **4** new types: month, day of the month, day of the week, and hour. In this way, we will get new variables that an ML model will process and find possible dependencies and correlations. Since we have created new variables, we will exclude the original variable "Timestamp" from the table. The "Day of the week" variable contains values from **0 to 6**, where each number represents a specific day of the week (from Monday to Sunday)

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Clicked on Ad	Month	Day	Hour	Weekday
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	0	3	27	0	6
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	0	4	4	1	0
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	0	3	13	20	6
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	0	1	10	2	6
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	0	6	3	3	4

3.3 Feature Extraction and Selection

The data scientist's data has several features that may or may not be relevant to the topic of interest. Also, it may not be in a suitable format. The first and foremost task to the data scientist is to extract the appropriate collection of attributes that preferably suits the learning algorithm. Before processing, it needs to be transformed to prevent relapse problems like overfitting and underfitting as presented. The following Table 1 shows the list of features present in the dataset.

Features	Description
Daily Time Spent on Site	User time spent on the website in minutes.
Age	User age in years
Area Income	Avg. Income of geographical area of user
Daily Internet Usage	Avg. minutes a day consumer is on the user.
Ad Topic Line	The headline of the advertisement
City	City of user
Male	Whether or not the user was male
Country	Country of user
Timestamp	Time at which user clicked on Ad or closed window
Clicked on Ad	0 or 1 indicated clicking on Ad

Table 1: List of features

The proposed ad-click prediction model is based on human features. To adapt to this, certain human-related features like Frequent Time Spent on Website, Lifetime, field Revenue, Frequent Internet Usage, and Gender are alone considered in this model. These attributes are extricated from the dataset to develop the prototype efficiently. Some features such as Advertisement Topic Line, City, Country, Time-stamp are not human, so they are ignored from consideration. The features that are taken into consideration are shown in Table 2. All extracted attributes have been indoctrinated into a convenient form to make study easy.

Features	**Description**
Daily Time Spent on Site	User time spent on the website in minutes.
Age	User age in years
Area Income	Avg. Income of geographical area of user
Daily Internet Usage	Avg. minutes a day consumer is on the user.
Male	Whether or not the user was male
Clicked on Ad	0 or 1 indicated clicking on Ad

Table 2: Features taken into consideration

3.4 Train and take a look at knowledge Sets

Once the dataset is processed, we want to divide it into 2 components that's coaching and take a look at set. We'll Take and use the train_test_split operate for that and every one variable except ‘Clicked on Ad’ are the input values x for the cubic centimetre models. The variable ‘Clicked on Ad’ are keep in y, can represent the prediction variable and that we at random selected to portion thirty third of the whole knowledge for the coaching set.

3.5 SUMMARY OF VARIOUS PREDICTION

3.5.1 Experimental Results

Algorithm	Accuracy
LogisticRegression	95.33%
RandomForestClassifier	96%
XGBClassifier	95%
Linear Support Vector Classification	96%
k Nearest Neighbors Classifier	68.85%

Table 3: Accuracy values of ML models

4.1 Data from User

Here we will collect data of user such as Daily Time Spent on Site, Age, Area Income, Daily Internet Usage, Male

4.2 Data preparing

Here given data will be undergone all the pre-processing techniques (3.3) which we done on the early available dataset.

4.3 Model called for the data

The saved model will be called for the prediction on the given data.

4.4 Predicted data

On the given data the loaded model will perform prediction.

5. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user is able to see input fields on logging in	1. Application is accessible 2. User is able to see input fields.	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is able to see input fields. 3. User is able to edit input fields.	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is able to see input fields. 3. User is able to edit input fields. 4.User is able to see submit button.	User should get Submit button to submit the inputs
Verify whether user is presented with prediction results on clicking submit	1. Application is accessible 2. User is able to see input fields. 3. User is able to edit input fields. 4.User is able to see submit button.	User should be presented with Predicted with results on clicking submit