Low-Level Design (LLD)

User Response Prediction System Using Machine Learning Techniques

Nabanita Chatterjee

03/08/23

Northcap University

Contents

Abstract

This project focuses on predicting profitable users who are likely to click on target ads through activity targeting in the advertising industry. The objective is to select potential users who are more likely to connect with ads by analyzing their clicking and web browsing data, thereby displaying the most relevant ads to them. The study involves an empirical analysis of various web of things techniques to predict ad click behavior. Click prediction is performed on a binary scale, where 1 represents a click and 0 represents no click.

The dataset used in this project is 'advertizing.csv', obtained as part of a Kaggle competition. Feature selection is conducted to eliminate attributes that do not contribute significantly to improved classifier accuracy. Both manual examination and feature selection capabilities are utilized to analyze the data effectively.

Overall, the project aims to leverage machine learning techniques to identify potential ad-clicking users and optimize ad targeting strategies for advertisers in the online advertising industry.

1. Introduction

In this project, we aim to develop an AI algorithm that predicts whether a specific client will click on an advertisement. Online advertising has become dominant, but targeting the right audience remains a challenge. To address this, we will work with advertising data from a marketing agency, containing 10 variables, including 'Daily Time Spent on Site', 'Age', 'Area Income', 'Daily Internet Usage', 'Ad Topic Line', 'City', 'Male', 'Country', 'Timestamp', and 'Clicked on Ad'. Our focus is on predicting the 'Clicked on Ad' variable, with outcomes 0 (no click) or 1 (click). By analyzing the other nine factors, we aim to accurately predict the 'Clicked on Ad' outcome. Additionally, we will conduct exploratory data analysis to understand how 'Daily Time Spent on Site' and 'Ad Topic Line' influence the user's decision to click on the ad.

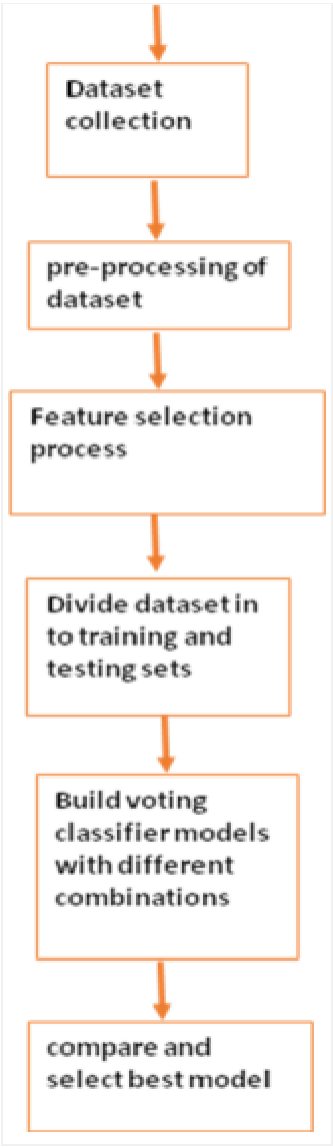
1.1 What is Low-Level design document?

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 programfrom 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 thenrefined 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 observed in the table, the "Ad Topic Line" column contains unique values for each entry, making it unsuitable for prediction due to the lack of data patterns. Similarly, the "City" column contains 969 unique values out of 1000, making it challenging to analyze effectively.

To enhance the data for prediction, we will omit these two categorical columns from further analysis. However, the "Country" column has a unique element, "France," which repeats nine times. We will consider this variable for analysis.

By examining the data, we found that there are 237 different unique countries in the dataset, and no single country dominates the records significantly. The presence of a large number of individual elements can hinder valuable relationship identification for a machine learning model. Therefore, we will exclude this variable as well.

Next, we will focus on the "Timestamp" category, representing the exact time when users clicked on the advertisements. To extract meaningful insights, we will expand this category into four new features: month, day of the month, day of the week, and hour. These new variables will provide additional information that the machine learning model can process to uncover possible dependencies and correlations. After creating these new variables, we will remove the original "Timestamp" variable from the table.

The resulting "Day of the week" variable will contain values from 0 to 6, with each number representing a specific day of the week, ranging from Monday (0) to Sunday (6). This transformation will allow us to gain more meaningful insights from the timestamp data and improve the predictive capabilities of our model.

	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 initial task is to extract relevant features from the dataset that are suitable for the ad-click prediction model. Some features may not be relevant to the topic of interest, and the data may not be in an appropriate format, leading to potential issues like overfitting and underfitting.

In the proposed ad-click prediction model, we focus on human-related features to adapt to the context. Specifically, features like 'Frequent Time Spent on Website,' 'Lifetime,' 'Field Revenue,' 'Frequent Internet Usage,' and 'Gender' are considered in this model. These attributes are extracted from the dataset to efficiently develop the prototype.

On the other hand, features such as 'Advertisement Topic Line,' 'City,' 'Country,' and 'Timestamp' are not human-related and are therefore ignored for consideration. The final selected features are shown in Table 2. All extracted attributes are formatted appropriately to facilitate ease of study and analysis.

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

```
Accuracy of Logistic regression training set: 0.992

Accuracy of Decision Tree training set: 0.988
10 fold CV accuracy: 0.948 +/- 0.016
The cross validated score for Decision Tree Classifier is: 94.75
Text(0.5, 1.05, 'Confusion_matrix')

Accuracy of Random Forest training set: 0.994
10 fold CV accuracy: 0.966 +/- 0.013
The cross validated score for Random Forest Classifier is: 96.62
Text(0.5, 1.05, 'Confusion_matrix')

Accuracy of Support Vector Machine training set: 0.99
10 fold CV accuracy: 0.955 +/- 0.025
The cross validated score for Support Vector Classifier is: 95.5
Text(0.5, 1.05, 'Confusion_matrix')
```

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