User Response Prediction System using Machine Learning Techniques

Vivek Limbad, Manali Kadam, Rueben Patil, Siddhanth Ghag

Pursuing Masters in Data Science & Business Analytics.

Introduction

- The online advertising industry has become a multi-billion industry, and predicting ad CTR (click-through rate) is now central. Nowadays, different types of advertisers and search engines rely on modeling to predict ad CTR accurately.
- We will be predicting the ad click-through rate using the machine learning approach. Before that, let us first understand a few essential concepts and a general practice followed by search engines to decide which ads to display.
- **CTR**: It is the metric used to measure the percentage of impressions that resulted in a click.
- **Search ads**: Advertisements that get displayed when a user searches for a particular keyword.
- Paid search advertising is a popular form of Pay per click **(PPC)** advertising in which brands or advertisers pay (bid amount) to have their ads displayed when users search for specific keywords.
- Relevance of Predicting CTR through a real-life example:

$$\text{CTR} = \frac{\text{Number of click-throughs}}{\text{Number of impressions}} \times 100(\%)$$

- Typically, the primary source of income for search engines like Google is through advertisement. Many companies pay these search engines to display their ads when a user searches for a particular keyword. Our focus is on search ads and CTR, i.e. the amount is paid only when a user clicks on the link and redirects to the brand's website.
- Different advertisers approach these search engines with their ads and the bidding amount to display their ads. The main objective of these search engines is to maximize their revenue. So the question is, how does a search engine decide which ads to display when a user searches for a particular keyword?
- Till now, we have seen what ad click prediction is and why is it important. Let us now explore how to calculate ad click prediction by performing machine learning modeling on a dataset. We will build a Logistic Regression model that would help us predict whether a user will click on an ad or not based on the features of that user. And hence calculate the probability of a user clicking on an ad.
- Using these probabilities, search engines could decide which ads to display by multiplying the possibilities with the bid amount and sorting it out.

Problem Statement

- In this project, we will work with the advertising data of a marketing agency to develop a machine learning algorithm that predicts if a particular user will click on an advertisement.
- The data consists of **10** variables:
- **'Daily Time Spent on Site', 'Age', 'Area Income', 'Daily Internet Usage', 'Ad Topic Line', 'City', 'Male', 'Country', Timestamp' and 'Clicked on Ad'.**
- The primary variable we are interested in is **' Clicked on Ad'.**
- This variable can have two possible outcomes: 0 and 1, where 0 refers to a user who didn't click the advertisement, while one refers to the scenario where a user clicks the ad.
- We will see if we can use the other **9** variables to accurately predict the value **'Clicked on Ad'** variable.
- We will also perform some exploratory data analysis to see how 'Daily Time Spent on Site' in combination with 'Ad Topic Line' affects the user's decision to click on the ad.

Objective

- **The goals of this project are to deeply explore data to do with advertising, perform quantitive analysis and achieve predictions from the data using machine learning techniques.**
- **The table below describes the features of the data.**
- **Feature Description**

1. Daily Time Spent on a Site - Time spent by the user on a site in minutes.

2. Age - Customer's age in terms of years.

3. Area Income - Average income of geographical area of consumer.

4. Daily Internet Usage - Average minutes in a day consumer is on the internet.

5. Ad Topic Line - Headline of the advertisement.

6. City - City of the consumer.

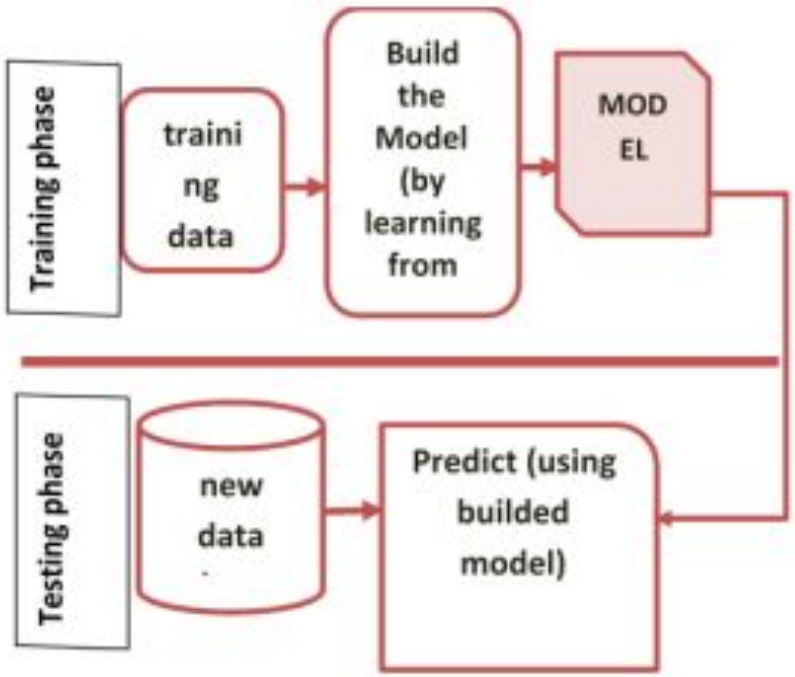
7. Male - Whether or not a consumer was male.

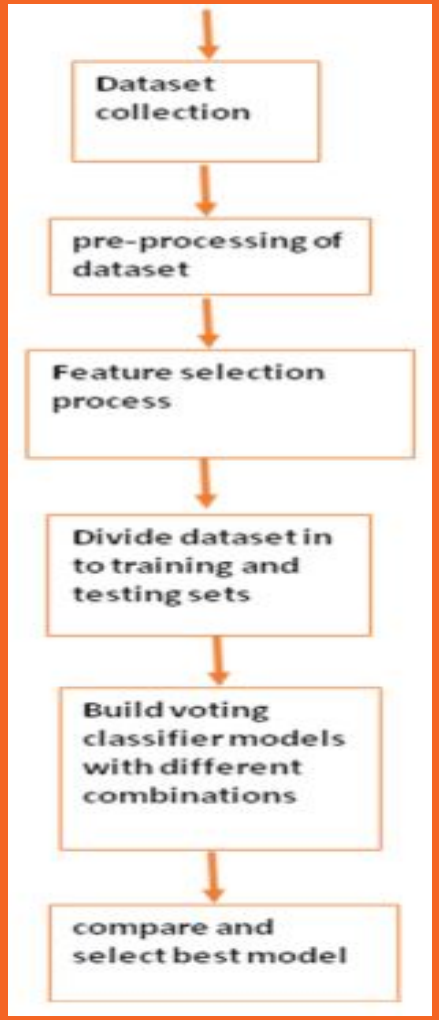
8. Country - Country of the consumer.

9. Timestamp - Time at which user clicked on an Ad or the closed window.

10. Clicked on Ad - 0 or 1 is indicated clicking on an Ad.

Architecture





Overview :

The whole project is divided into 7 steps :

1. Importing dependencies and loading Data set
2. Data Preprocessing
3. Exploratory Analysis
4. Statistical Analysis
5. Train Test Split
6. Training the Model
7. Testing the model accuracy

The data set was provided by the hosted competition, you can find data sets here

—

Step 1 : Importing Dependencies and Loading Dataset

In order to do the predictive analysis we need to import some python libraries which will help in data visualization, dealing with data set and will also provide pre-implemented Machine Learning models.

Step 2 : Data Preprocessing

This is the Most important step of all Machine Learning and Data Science projects. It is about **80%** of the overall work. For this project I have done data cleaning manually by identifying the relation between multiple columns, although there are some tools and standard procedures available but I found it more suitable as per the accuracy.

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

```
object_variables = ['Ad Topic Line', 'City', 'Country']
data[object_variables].describe(include=['O'])
```

	Ad Topic Line	City	Country
count	1000	1000	1000
unique	1000	969	237
top	Reactive bi-directional workforce	Lisamouth	France
freq	1	3	9

Step 2 : Data Preprocessing

As we can see from the table that all the values in column **"Ad Topic Line"** is unique, while the **"City"** column contains **969** unique values out of **1000**. There are too many unique 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. The third categorical variable, i.e "**Country**", has a unique element (France) that repeats 9 times. Additionally, we can determine countries with the highest number of visitors:

The table shows the 20 most represented countries in our DataFrame.

col_0 Country	count
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

Step 2 : Data Preprocessing

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 categories: month, day of the month, day of the week, and hour. In this way, we will get new variables that an ML model will be able to 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

Step 3 :

Exploratory Data Analysis

We have performed following analysis.

1. Distribution of daily time with ads
2. Distribution of daily internet with ads
3. Top city with daily time
4. Top city with area income
5. Top city with avg internet
6. Investigating the Country Variable
7. Top city with avg internet
8. Top country with daily time
9. Top city with area income

Extracted Features Visualizations

1. Investing Country Variable
2. Distribution of top 12 country's ad clicks based on Sex
3. Hourly distribution of ad clicks
4. Distribution by each hour and by gender.
5. Daily distribution of ad clicks
6. Monthly distribution of ad clicks
7. Top Ad clicked on specific date
8. Daily internet usage and daily time spent on site based on age
9. All ad topics (word cloud)
10. Distribution and Relationship Between Variables
11. Visualizing target variable Clicked on Ad
12. Click on Ad features based on Sex
13. Distribution of who clicked on Ads based on area income of sex.
14. Correlation Between Variables (Heatmap)

Step 4 : Statistical Analysis

We have performed following analysis.

1. Examine the data
2. Data type and length of the variables
3. Check for Missing Values
4. Numerical and Categorical Variables Identification
5. Summarizing Numerical Variables
6. Summarizing Categorical Variables
7. Categorizing Quantitative and Qualitative Variables
8. Outliers
9. Identifying Potential Outliers using IQR
10. T-Test & F-Test Between Groups of People that Clicked on Ads
11. Variance
12. Mean
13. Testing for Normality
14. Mann-Whitney U Test

Step 5 : Train and Test Data Sets

Once the dataset is processed, we need to divide it into two parts: training and test set. We will import and use the train_test_split function for that. All variables except 'Clicked on Ad' will be the input values X for the ML models. The variable 'Clicked on Ad' will be stored in y, and will represent the prediction variable.

X_train and Y_train are used to train the Machine Learning model while x_test is used as input for making predictions which will be then validated with the y_test values.

```
# Importing train_test_split from sklearn.model_selection family
from sklearn.model_selection import train_test_split

# Assigning Numerical columns to X & y only as model can only take numbers
X = df[['Daily Time Spent on Site', 'Age', 'Area Income', 'Daily Internet Usage', 'Male']]
y = df['Clicked on Ad']

# Splitting the data into train & test sets
# test_size is % of data that we want to allocate & random_state ensures a specific set of
#this train test split is going to occur randomly
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# We dont have to use stratify method in train_tst_split to handle class distribution as it
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

Step 6 : Training the Machine Learning Model

Since there are two categories in output data which are :

Either Customer will click on the ad (i.e. 1) or

Customer won’t click on the ad (i.e. 0)

It simply suggests that it is a classification problem. Also visualization of data also gave an intuition that there are decision boundaries which can be used as the basis of selecting the Machine Learning model.

Step 7 : Checking Model Accuracy

Final step is to check the accuracy of the Machine Learning model which we have created for ad click prediction :

Experimental Results :

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

Conclusion

While the random forest could have been tuned further, it had good precision. It did not take too much time to fit the model, which would allow for fast tuning of parameters. The linear kernel SVC took a very long time to provide the data. It is shorter at predicting than the random forest and k nearest neighbors classifiers. This time taken to fit the data is mitigated, as only one parameter must be tuned. The k Nearest Neighbors performed the worst in AUC and prediction time. This was not a good model for this data. In the end, the linear SVC should be used as it had a slightly higher AUC and faster prediction time when compared to the random forest.

The end accuracy of this project is 96%. This is not anywhere near as good as the random forest or support vector classifier from before!

Conclusion

Daily Internet Usage is an essential feature.

The lower Daily Internet Usage and Daily Time Spent on Site, the more likely to click the ad.

Sex and Age are the least relevant feature. Area Income affects a little. Thus,

targeting ads to the people who use the internet little and rarely spend time on a

website is efficient to make more likely to click the ad.

Q & A

Why Ad click is important ?

A company wants to know the CTR (Click Through Rate) in order to identify whether spending their money on digital advertising is worth or not.

A higher CTR represents more interest in that specific campaign, whereas a lower CTR can show that your ad may not be as relevant. High CTRs are important because they show that more people are clicking through to your website. Along with this high CTRs also help to get better ad position for less money on online platforms like Google, Bing etc.