İZMİR
KÂTİP ÇELEBİ
ÜNİVERSİTESİ
2010
GRADUATE SCHOOL OF NATURAL
AND APPLIED SCIENCES

Making the Pixels Visible

Submitted to the Graduate School of Natural and Applied Sciences in partial fulfillment of the requirements for the degree of

Master of Science

in Software Engineering

by

Işıl Işık Mutlu

July, 2023

This is to certify that we have read the thesis **Making the Pixels Visible** submitted by **Işıl Işık Mutlu**, and it has been judged to be successful, in scope and in quality, at the defense exam and accepted by our jury as a MASTER'S FINAL PROJECT.

APPROVED BY:

Advisor: **Prof. Dr. Femin Yalçın Küçükbayrak**
İzmir Kâtip Çelebi University

Declaration of Authorship

I, **Işıl Işık Mutlu**, declare that this thesis titled **Making the Pixels Visible** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for the Master's / Doctoral degree at this university.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. This thesis is entirely my own work, with the exception of such quotations.
- I have acknowledged all major sources of assistance.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 28.07.2023

Making the Pixels Visible

Abstract

Websites are one of the most important components of the internet. We use web browsers every day to browse websites using the internet for searching for information, buying goods, reading news, following online lectures, interacting on social media, etc. When a website is opened, it contains many components that are too complex for an ordinary user to be aware of. By using third-party tracking cookies or pixels added for advertising or analytical purposes, ad networks can track the user on the web. As browser extensions that use filter lists became more successful at blocking cookies, followers began to find alternative ways to identify users. One way for followers to identify individuals is to use invisible pixels. In this project, my goal is to perform exploratory data analysis and feature engineering of the dataset, using a previously created invisible pixel dataset, and eventually create a machine learning model that classifies these pixels and convert this machine learning model to suitable form to be used in a browser extension that can be developed.

Keywords: third-party cookies, invisible pixel, browser extension, machine learning, feature engineering, exploratory data analysis, tracker, website, machine learning model

Pikselleri Görünür Yapmak

Öz

Web siteleri internetin en önemli bileşenlerinden biridir. Bilgi aramak, mal satın almak, haber okumak, çevrimiçi dersleri takip etmek, sosyal medyada etkileşim kurmak vb. için interneti kullanarak web sitelerinde gezinmek için her gün web tarayıcılarını kullanırız. Bir web sitesi açıldığında, sıradan bir kullanıcının farkında olamayacağı kadar karmaşık birçok bileşen içerir. Web sitelerine reklam veya analitik amaçlı olarak eklenen üçüncü taraf izleme çerezleri veya pikselleri kullanarak reklam ağları kullanıcının web üzerindeki izlerini sürülebilir. Filtre listelerini kullanan tarayıcı eklentileri, tanımlama bilgilerini engellemede daha başarılı hale geldikçe, takipçiler, kullanıcıları tanımlamanın alternatif yollarını bulmaya başladılar. Takipçilerin bireyleri tanımlamasının bir yolu da görünmez pikseller kullanmaktır. Bu projede amacım, daha önce oluşturulmuş bir görünmez piksel veri seti kullanarak, veri setinin keşifsel veri analizini ve öznitelik mühendisliğini yapmak ve sonuçta bu pikselleri sınıflandıran bir makine öğrenimi modeli oluşturarak, geliştirilebilecek bir internet tarayıcı eklentisinde kullanılmak üzere bu makine öğrenimi modelinin uygun şekle dönüştürmektir.

Anahtar Kelimeler: üçüncü taraf izleme çerezleri, görünmez piksel, tarayıcı eklentisi, makine öğrenimi, öznitelik mühendisliği, keşifsel veri analizi, takipçi , web sitesi, makine öğrenimi modeli

This study is dedicated to my son Batuhan and my daughter Ecem.

Acknowledgment

I would like to thank my Advisor Prof. Femin Yalçın Küçükbayrak who encouraged me to study on the subject of my interest. I would like thank to Doç. Dr. Aytuğ Onan and all faculty members that are helpful and competent in their subjects. Also, special thanks to `all` who contributed to my learning path helping me understand how important my digital privacy is.

Table of Contents

Declaration of Authorship ii

Abstract iii

Öz... iv

Acknowledgment vi

List of Figures ix

List of Tables xi

List of Abbreviations xii

List of Symbols xiii

1 Introduction 1

1.1 Browsing the Internet 4

1.2 Third-Party Trackers: How they track? 6

1.2.1 Cookies 8

1.2.2 Fingerprinting 10

1.2.3 Respawning cookies 10

1.2.4 Invisible pixels (web bugs) 11

2 Exploratory Data Analysis and Feature Engineering 14

2.1 Exploring Pixel Dataset: How is it gathered? 14

2.2 Exploratory data analysis (EDA) 16

2.3 Feature Engineering 20

3 Machine Learning 22

3.1 Invisible Pixel Classifier 22

3.2 Evaluation Metrics 23

3.3 Random Forest Algorithm 28

3.3.1 Random Forest Algorithm Model Evaluation 29

3.4 XGBoost Algorithm 29

3.4.1 XGBoost Algorithm Model Evaluation 30

3.5 MLP-Artificial Neural Network Algorithm 30

3.5.1 MLP-Artificial Neural Network Algorithm Model Evaluation 31

3.6 Logistic Regression Algorithm 32

3.6.1 Logistic Regression Algorithm Model Evaluation 33

4 Model Deployment and Conclusion 34

4.1 Converting and saving the model to deploy in browser extension 34

4.1.1 Convert the scikit-learn model to TensorFlow.js format 34

4.1.2 Load the TensorFlow.js model in browser extension 34

4.2 Conclusion 35

References 36

Curriculum Vitae 39

List of Figures

Figure 1.1 Web document construction from different servers. 4
Figure 1.2 Figure 1.2 An example of an URL..... .5
Figure 1.3 An example of an HTTP header. 5
Figure 1.4 HTTP header request 6
Figure 1.5 Classification of Tracking..... 7
Figure 1.6 First-party cookies and Third-party cookies are set in page 9
Figure 1.7 Cookie respawning with browser fingerprinting tracking technique. 11
Figure 1.8 HTML source code of a webpage which contains a web bug. 12
Figure 1.9 One-pixel sized image..... 13

Figure 2.1 Imported libraries for initial work on the dataset. 17
Figure 2.2 Importing the pixel dataset. 17
Figure 2.3 First 10 rows of the pixel dataset. 17
Figure 2.4 Columns of the pixel dataset 18
Figure 2.5 There is no missing value in the dataset..... 18
Figure 2.6 Checking Info of the dataset. 18
Figure 2.7 Checking target variable `label` of the dataset. 19
Figure 2.8 Checking the class imbalance with bar-plot. 19
Figure 2.9 Count of each label in the dataset 19
Figure 2.10 Image format distribution in the dataset..... 20

Figure 3.1 Confusion matrix..... 24
Figure 3.2 ROC curve..... 27
Figure 3.3 Random Forest Algorithm..... 28
Figure 3.4 Random Forest algorithm classification report of the pixel dataset..... 29
Figure 3.5 XGBoost algorithm classification report of the pixel dataset 30
Figure 3.6 ANN structure 31
Figure 3.7 MLP-Artificial Neural Network algorithm classification report of the pixel dataset..... 31

Figure 3.8 S-shaped logistic function 32
Figure 3.9 Logistic regression algorithm classification report of the pixel dataset.. 33

List of Tables

Table 2.1 Categories of pixels in the pixel dataset 16
Table 2.2 Feature engineering of pixel dataset 21

List of Abbreviations

ANN	Artificial Neural Network
AU-ROC	Area under Receiver operating characteristics curve
CMP	Consent Management Platform
DB	Database
EDA	Exploratory Data Analysis
EU	European Union
FN	False Negative
FP	False Positive
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICO	Icon
IP	Internet Protocol
GPDR	General Data Protection Regulation
PNG	Portable Network Graphics
ML	Machine Learning
RGB	Red, green and blue
SOP	Same-Origin Policy
TCP	Transport Control Protocol
TN	True Negative
TP	True Positive
URL	Uniform Resource Locator
XGBoost	Extreme Gradient Boosting

List of Symbols

H^0	Null hypothesis

Chapter 1

Introduction

Websites are one of the most important components of internet. We everyday use web browsers to surf websites using internet to search for information, to buy goods, to read news, to follow online lessons, and to interact in social media etc. However, websites we trust may not be as innocent as we think. Once we open a website it includes many components that are so complex that an ordinary user cannot be aware of. One component that the users mostly are not aware of are third-party resources.

A third-party resource, is a resource that is served from another server, but is also authored by someone other than the site owner. Website owners or developers embed third-party resources which are useful for functionality or appearance of a website. Third-party resources can be web fonts, embedded IFrames of videos, adverts, or JavaScript libraries. Some examples of third-party resources used are web fonts served from Google Fonts, embedded analytic scripts from Google Analytics, embedded `Like` buttons or `Sign in with` buttons from social networks, embedded maps or video, or third-party services that handle shopping purchases [1]

For advertising and analytics purposes websites may use third-party tracking pixels and cookies. If the cookie or pixel is loaded by a HTTP request from the script which is loaded on the script owner's server, it is a third-party, otherwise, if it is loaded by a HTTP request from the script from the website owner's site it is considered first-party. When there is a third-party resource on a website, regardless of what it is, some information is passed to that third-party which place digital traces about the user. This is how both hosting website and advertisers monetize by collecting and selling users data.

Advertising is one of the important parts of the web and it is very different from the traditional offline advertising. Online advertising and offline advertising can be distinguished by two important criteria- 'Targetability' and 'Measurability'.

Targetability means finding customers that are more likely to be influenced by ad whereas measurability is how much the ad was successful. Internet eases the collection of large amounts of data providing measurability and establishes an easy to serve platform for customers providing targetability [2].

Although operational efficiency of companies is enhanced by the internet, tracking across the web raise concerns about privacy. Though companies have always collected information regarding customers, it wasn't that large-scope, large-amount and precise before. Data gathered by companies after digitalization have been more personal and concerning related to privacy.

According to Electronic Frontier Foundation (EEF):

> *"Trackers collect data about our clicks, impressions, taps, and movement into sprawling behavioral profiles, which can reveal political affiliation, religious belief, sexual identity and activity, race and ethnicity, education level, income bracket, purchasing habits, and physical and mental health, user's behavior to predict what they like, how they think, and what they are likely to buy, and it drives much of the third-party tracking industry.* [3]"

Historically, before digitalization, privacy concerns are more about public figures. Now, with recent advances in information technologies personal data is easier to collect and store. Interestingly before digitalization there were no wish-lists, abandoned shopping carts, list of past purchases and recommender systems which offers recommendations to product of interests to another customer who also interested in the same item. [2].

Concerns related to privacy made governments and people take steps. While governments made regulations, people make tools like adblockers to alleviate the concerns. As such, in European Union privacy laws ePrivacy Directive and later General Data Protection Regulation (GPDR) has come into effect. In Turkey, Personal Data Protection Authority makes decisions about processing of personal data by the

data controller. While there is no direct regulation related to invisible pixels, there is a guide for using cookies:

> *"While there is no need to obtain explicit consent for the necessary cookies required for a website to function properly, the use of cookies for advertising, marketing and performance purposes is subject to the explicit consent of the data subjects.* [4]"

According to GPDR the host of a website or a third-party needs a legal basis for data collection. The most common legal basis is the consent from the user. Websites mostly use consent banners or consent notices to gather consent and have the legal basis for the collection of personal data. Consent Management Platform (CMP) handles the legal terms and conditions for privacy policies, and implements the consent banner displaying third-parties involved and the purpose of the tracking technologies used. When it collects consent from the user, it redistributes consent to advertisers, trackers and third-parties. Data collection purposes has divided into usage categories. According to GPDR, if the collection and processing of user data has various purposes, consent should be given for all categories.

Tools that block advertisers and trackers are used for alleviating privacy risks. Filter lists, a set of rules to recognize adverts or trackers, such as EasyList, Easy Privacy and Disconnect are being used in blocking specific web requests by browser extensions like AdBlock Plus and uBlockOrigin [5, 6]. However, they are not able to block all trackers. Fouad et al. demonstrated that two popular methods of detecting trackers, based on EasyList&EasyPrivacy and on Disconnect lists respectively miss 25.22% and 30.34% of the trackers they detected [6]. Fouad [6] elaborates that trackers can avoid detection by filter lists by using different subdomains, registering a new domain and they can include tracking behavior into functional website content that will never be blocked.

1.1 Browsing the Internet

We use browsers to surf web pages on the internet. The protocol we use to make requests from other web servers from our machine is called HTTP. *Hypertext Transfer Protocol (HTTP)* is an application-layer protocol for transmitting hypermedia documents, such as HTML but also images and videos or to post content to servers, like with HTML form results. HTTP can also be used to get parts of documents to update Web pages when demanded. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually by the Web browser. When a user makes an attempt to visit a website, the browser not only request the web page from the server but also requests the images, multimedia content, fonts, JavaScript Libraries, and style sheets and many other resources. The images and code may be hosted on other servers which needs to make further HTTP requests. A complete document is constructed from the different sub-documents fetched.

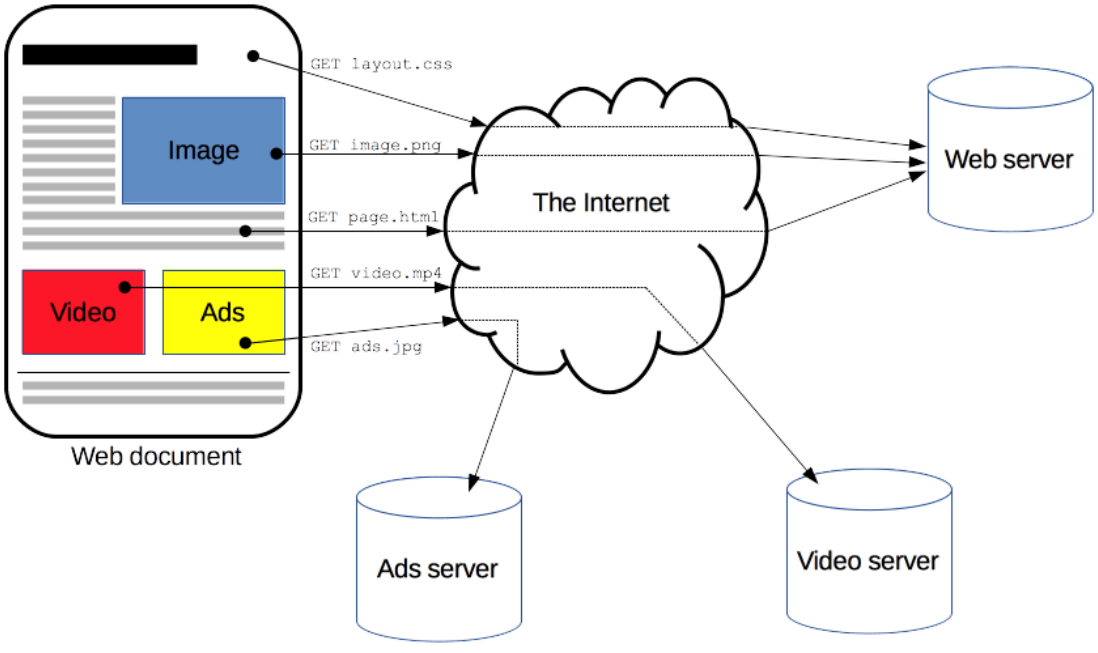


Figure 1.1 Web document construction from different servers

The URL you see in the address bar of your browser is the address for the first request to the server. Parameters can include tracking identifiers.

Figure 1.2 An example of an URL

When a client wants to communicate with a server, first the client opens a TCP connection. The TCP connection is used to send a request, and receive an answer. Second, it sends an HTTP message. There are two types of HTTP messages, requests and responses. Once you make a request, HTTP protocol send a message to the server called HTTP request. HTTP request contains method, path, version of the protocol and HTTP request headers.

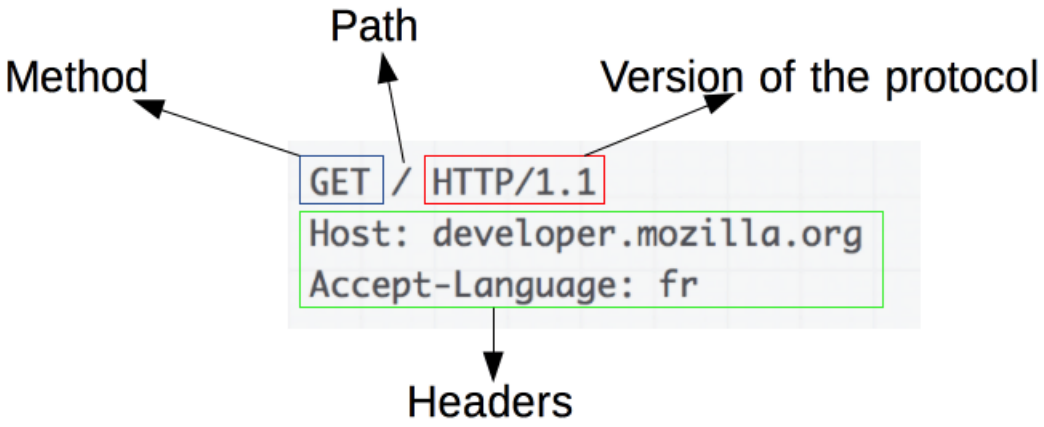


Figure 1.3 An example of an HTTP header

HTTP headers let the client and the server pass additional information with an HTTP request or response. An HTTP header consists of its case-insensitive name followed by a colon (:), then by its value. HTTP request headers contain more information about the resource to be fetched, or about the client requesting the resource [8].

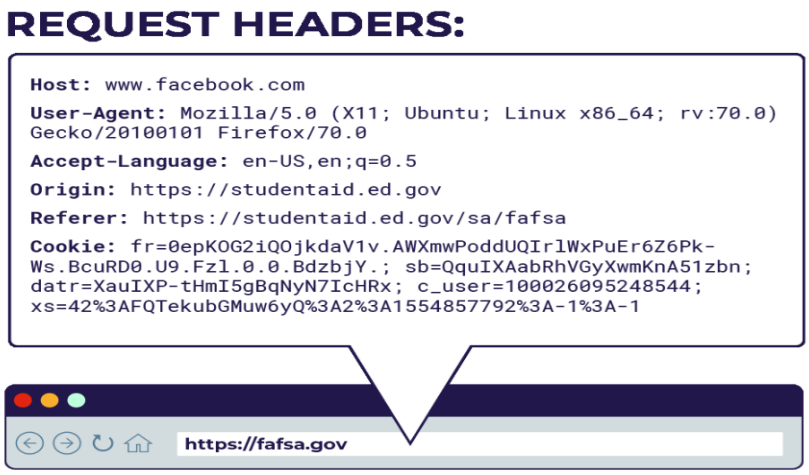


Figure 1.4 HTTP header request

The information retrieved by the server by request headers is called passive features. If information such as browser, time zone is retrieved by the remote server using a script, these features are called active features [7].

Above you can see a request to fafsa.gov domain. However, this is not the only request from the website most of the time. There may be also be requests to third-party trackers.

1.2 Third-Party Trackers: How they track?

When you click on a web link not all the parts of the page may come from the domain you visited but also loads content from other servers other than you visited. The data that is collected through the domain you visited is the first-party domain. For example, when you login to your Facebook profile, you will be giving your sensitive data to the first party. All web elements such as style sheets, JavaScript, images etc. can allow referencing external URL may be used for third-party tracking.

Although not all of them are trackers, the domains that are embedded in web pages you visit are “third-party domains”. Tracking is deployed by different mechanisms that impacts the user’s privacy differently. Some tracking is made on the same domain, others are able to create a browsing history of the user across different domains. Figure 1.5 gives an overview of the classification of tracking based on tracking behavior.

Tracking techniques can also be divided into two categories: stateful and stateless. Stateful techniques relies on storing unique identifiers such as cookies. In this technique trackers store a unique identifier in the cookies and later use it to track a user across websites. Stateless tracking, however doesn't depend on storage because the user can delete it. In this technique a combination of identifiers, browser fingerprinting, such as browser and machine features are used to identify a user [7]. Tracking techniques include uniquely identifying cookies, local storage "supercookies," canvas fingerprinting and invisible pixels [3,7].

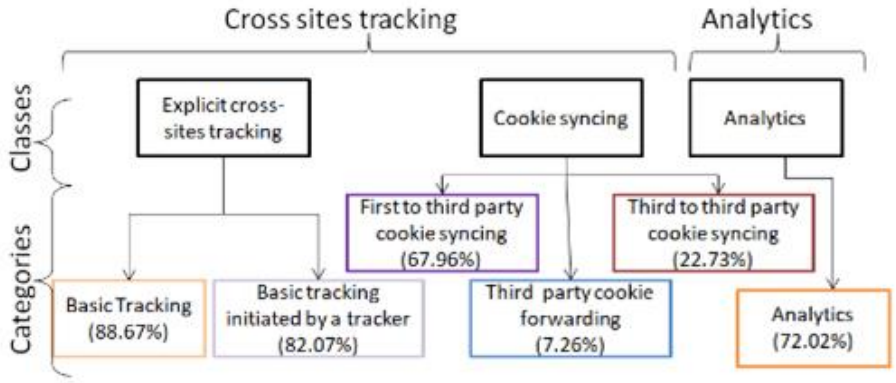


Figure 1.5 Classification of Tracking

Basic tracking is made only on the websites where the identifier cookie is set by a third-party domain and the same tracker has its content on other webpages. In basic tracking initiated by a tracker technique, trackers get to know the website visited by the users, without being included in it because the basic tracker in the website can redirect a second third-party or include it. Cookie syncing creates a more complete profile of a user by merging profiles collected by third-parties on different websites.

There are two distinct categories of cookies syncing: third to third party cookie syncing and third-party cookie forwarding. Third party to third party cookie syncing is like a set of trackers who make basic tracking exchange data that they captured about a user. This is the most dangerous tracking behavior since the tracker can log users visits where its partners included.

In third-party cookie forwarding, the second third-party tracker doesn't have its own identifier but uses the first trackers identifier to recreate the browsing history of the

user. This category is less harmful than cookie syncing because, it just passively gets the user's browsing history not contributing to the profile creation. Using first to third party cookie syncing, first party websites sync cookies with third parties. Websites use third-party analytics services to measure the number of visits, the number of visited pages in the website etc. In analytic behavior third-party cannot track the user across websites because it doesn't set its own cookie in the user's browser therefore, it isn't able to create browsing history of the user. It can only identify the user on the same domain [6].

1.2.1 Cookies

The most known web tracking technique is based on cookies. Cookies are a small piece of text that is stored in the users browser. It is associated with a particular domain but only some cookies contain unique identifiers and hence are capable of tracking the users. Each time the browser makes a request to the website that the browser stored the cookie for, it also sends the cookie that was set before. Therefore, the website knows which user or device the request is coming from. Not all cookies are trackers but mostly third-party cookies are designed for tracking users. Tracking cookies which last at least a month after being placed are also called identifier cookies [6].

The cookie stored in the browser is identified by a tuple (host, key, value). [6]. When a user visits a web site URL with the help of browser, a *web server returns a* content, like text or an image, or with a simple acknowledgement HTTPS response that it received your request. It can also attach a *cookie*, which is a unique identifier for tracking purposes. If it is a first-party cookie like logging into a Facebook, Facebook will receive your public profile and email address information. The host and owner of this cookie is Facebook. Facebook is now capable of tracking the user within the same website.

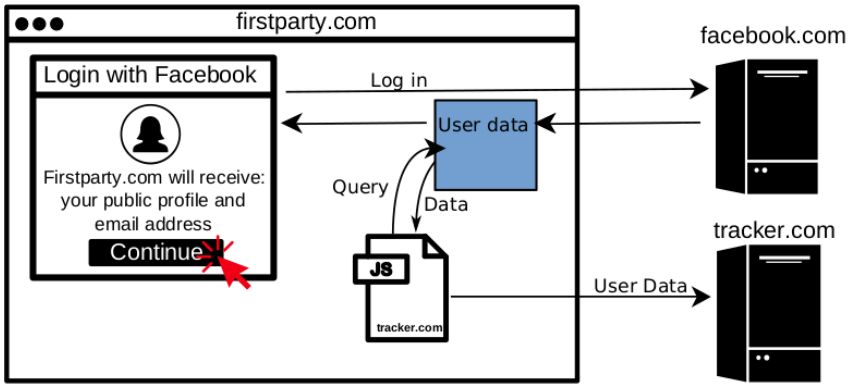


Figure 1.6 First-party cookies and Third-party cookies are set in page

However, if there is a third-party tracker on the website the user visits, the browser can also make requests to a tracker server as shown in figure 1.6. The responses can be from numerous tracker servers. This time, the cookie owner is the third-party server and cookie host is Facebook. The cookie can also be set programmatically via a script as shown in above figure 1.6. Third-party cookies are able to track users cross-websites [6].

The trackers can also share user data to build rich users profiles. In order to share user data, advertising networks perform a process known as cookie syncing. Since a cookie set by a script (owner) in one domain (tracker.com) in which the script is in context of another domain (site.com) which is the host of the cookie, host is the origin of the execution context (site.com) of the script and has access to the cookie due to Same Origin Policy (SOP) [6,7]. Same-Origin Policy is intended to ensure each site in the browser is isolated from all others, and that multiple pages from the same site are not isolated [9]. If a cookie owner sets a cookie in the context of several websites, it can track the same user on several websites. For example, if a user goes to site A. com; obtains page and page contains <iframe src= “B.com”>. Then, browser goes to B.com; obtains page. HTTP response contains cookie from B.com is called a 3rd party cookie. Further, if server goes to site D.com which contains <iframe src= “B.com”> then B.com obtains cookie set when visited A.com. Now, B.com knows user visited A.com and D.com. This is how tracking on several website occur using cookies [7].

With a cookie each user is given a unique ID and this identifier is passed through the URL parameters. However, the cookie can be set in two different ways: Cookie set by

HTTP(s) header *Set-Cookie* parameter and set by a script that can be observed by using *document. cookie*. Fouad et al. found that there are different ways of sharing identifier information among trackers. It can be shared as: a) a URL parameter value, b) part of the parameter value, c) part of the cookie, d) Google Analytics sharing which rely on delimiters, e) encoded Base64 sharing, f) Encrypted sharing [7].

Advertisers and other companies can then store the rest of the information about the user on their own systems. In order to accurately target an audience, advertisers need to incorporate user data from various domains and sources, which happens as part of data-buying agreements and partnerships between different advertising companies. This process is known as ***cookie syncing*** [10].

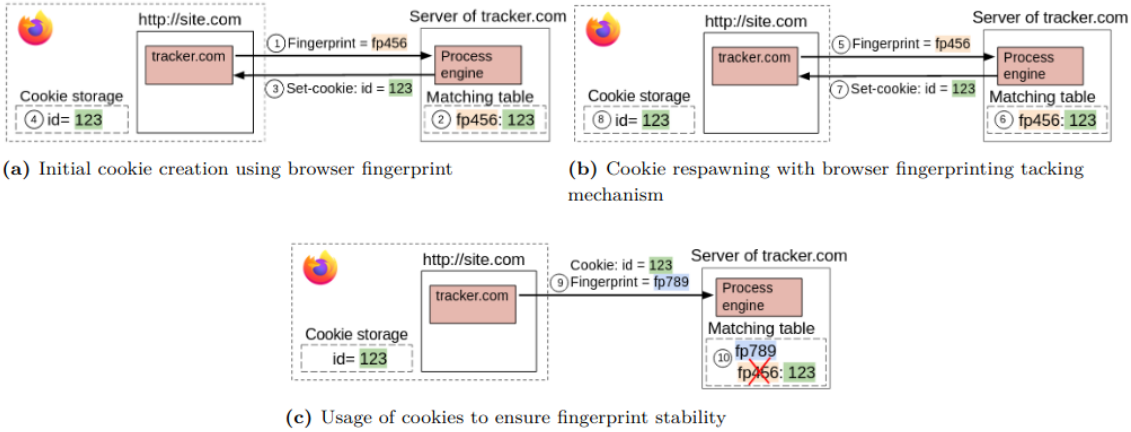
1.2.2 Fingerprinting

Information sent along with the request made every time the browser visits a site, along with attributes that can be discovered by running a fingerprinting script on the page. Attributes include the resolution of the machine's screen, the specific version of software installed, and time zone. Any information that your browser exposes to the websites you visit can be used to help assemble a browser fingerprint. These features are also called active features [7]. Researchers have found *canvas fingerprinting* techniques to be particularly effective for browser identification. The *HTML Canvas* is a feature of HTML5 that allows websites to render complex graphics inside of a web page [3].

1.2.3 Respawning cookies

Because cookie-based tracking can be blocked easily by blocker extensions respawning cookies are emerged. After cleaning the browser storage, cookies can be recreated using HTML5 local storage, Flash cookies, eTags, Indexed DB that contains the same information in the cookie. According to Fouad et al. cookies can also be respawned using active and passive features together. Hence, the same exact cookie can be created even the user deletes it. In this technique a browser fingerprint is created and stored in browser's cookie. It can be recreated because the fingerprint can be created again with the same features. Even more if the fingerprint changes due to changing features the new fingerprint can be matched to the old fingerprint of the same

user as shown in figure 1.7 (c). This makes third-party tracking persistent for the deletion of the cookie or a fingerprint change and allows for creation if larger users' profiles. Only if they are changed at the same time, the tracker cannot identify the user. Features for fingerprinting are both browser related and machine related such as Accept language, Geolocation, User agent, WebGL, Canvas, IP address, and Time Zone [7].



(a) Initial cookie creation using browser fingerprint

(b) Cookie respawning with browser fingerprinting tacking mechanism

(c) Usage of cookies to ensure fingerprint stability

Fig. 1.7 Cookie respawning with browser fingerprinting tracking technique

Web browsers like Google Chrome are claiming that they are phasing out third-party cookies to a more privacy safe browsing experience for the user however via persistent tracking with respawned cookies, domains can still track the users using cookie respawning with browser fingerprinting. [7, 11]

1.2.4 Invisible pixels (web bugs)

Another way to track users by third-party is to use invisible pixels which are also named web bugs, web beacons, tracking pixel, invisible images, and a pixel tags. It is shown in Figure 1.9 which can't be seen unless the web page is zoomed. They are mostly one-pixel-per-one-pixel images embedded to a certain web page in an <img> tag, inside and IFrame or JavaScript and hosted from another server. The difference between a tracking image and a non-tracking image is that the tracking image send identifying and tracking information (third-party cookies) of the user to the server it is requested from when it is fetched by the browser. Although with every image it is possible to track users, invisible pixels have advantage of little loading delay and

invisibility to the user. They generally allow the third-party to send information using the requests sent to retrieve the images. Because they are invisible and users are unaware of them, they are a threat to the privacy of users. It is important to mention not every image of size 1x1 pixel is a tracking pixel, sometimes they are used for positioning elements in the webpage. [5, 6, 12].

Although the idea of keeping track of the number of hits on websites or total number of active users, how many times a post is read or forwarded are not seem to be intrusive, web bugs or invisible pixels, one-pixel sized images loading from a different than the rest of the page, keep track of personal identifying information about web page users and email users. The information that an invisible pixel can collect are: IP address of the web page user, the URL of the page the bug present, time when the web site or email is visited, user-agent of the client who viewed the website, running programs, and previously set cookie value. If those identifiers are tied to a phone number or street address for example it will be enough to identify a person. When used in emails the sender can follow if it is read, the IP address of the recipient and how often it is read and forwarded. Companies use this technique to show personalized advertisements directed to a particular person. The invisible pixel can be found investigating the html source code of the web page. Below image tag contains a php file instead of an image file which make a request to another server thereby the third-party receiving a tracking information [12].

```
File  Edit  View  Help
<html>
<body background=loc2.jpg>
<Title>UnSecure_Spy</Title>
<style> .head1(margin:10px 250px;box-shadow:10px 10px 5px #888888;background:#FFFeef;border:8px solid #ffcc00;padding
<img src=webbug.php>
<h1 class=head1><b>Home > UnSecure-Spy</b></h1>
<style>
div.floating-menu
(transition:width 6s ;box-shadow:10px 10px 5px #888888;opacity:0.5;right:38%;bottom:10px;position:fixed;background:#D
div.floating-menu a, div.floating-menu h3
(display:block;margin:1 0.5em;)
div:hover
{
transform:scale(1.2,1.2);
-ms-transform:scale(1.2,1.2); /* IE 9 */
-moz-transform:scale(1.2,1.2); /* Firefox */
-webkit-transform:scale(1.2,1.2); /* Safari and Chrome */
-o-transform:scale(1.2,1.2); /* Opera */
}
ul
{
list-style-type:none;
```

Figure 1.8 HTML source code of a webpage which contains a web bug

Figure 1.9 One-pixel sized image

Ruohonen and Leppanen [17] have shown that they can classify the third party 1x1 images from other images by analyzing the source code of landing HTML page and extract images from the <img> tag. They also confirm that adblockers are unable to fully block these classical image-based tracking pixels. Fouad et al. [6] demonstrated that two popular methods to detect tracking based on EasyList&EasyPrivacy and on Disconnect lists respectively miss 25.22% and 30.34% of the trackers. Fouad et al. shown that pixels are widely deployed, being present on 94% of domains and constituting 35.66% of all third-party images, responsible for 23.34% tracking requests.

A real-world example of a random netloc in the URL of a tracking pixel: https://w2txo5aaxmi27hddahoncgxp2bwomwddvbnz5r7s6be29835369921bb-am1.e.aa.online-metrix.net/fp/clear.png?org id=w2txo5aa&session i-d=prsktdt1rhhv35ggtezs2jhl&nonce=6be29835369921bb&di=yes

The first eight characters of the random netloc is equal to org id in the query and the sixteen characters before am1 is equal to nonce in the query. The nonce or the numbers in between org id could likely be used to identify a user. Manually changing numbers in the browser still retrieves the same pixel [6].

Chapter 2

Exploratory Data Analysis and Feature Engineering

2.1 Exploring Pixel Dataset: How is it gathered?

A training dataset containing consent label data and observed pixels in json format from https://zenodo.org/record/6320338#.ZDVYs45ByC2 which is developed by Ganz, Rita for their Bachelor Thesis is used in this study. The feature extraction scripts can also be found in https://github.com/ganzri/Tracking-Pixels. Ganz [5] collected a dataset which crawl a list of websites and collected the labels from consent notices provided by CMPs, tracking pixels and the identified feature of the pixels and extracted the pixels.

Ganz inspired from web crawler for cookies by Dino Bollinger. However, collecting purpose labels from CMPs may be misleading due to varying levels of human operators labelling and intentional mislabeling. Tough there is a publicly available Database for the cookies named Tokopedia there is no publicly accessible equivalent for pixels and their purpose labels. They choose domains by using Tranco for the website popularity rankings as it combines data from different sources (Alexa, Cisco Umbrella, and Majestic), provides higher stability and resilience against manipulation than any of the three resources on their own and allows replicability as it provides a permanent record of the lists. They used Tranco lists, one for European domains and one for world-wide domains.

Since they need to be able to extract pixel specific purpose declarations from a CMP, it is needed in CMP that:

1) declared purpose labels must be remotely accessible in a computer-readable format,

2) include some sort of identifier as such list purpose for each cookie or other tracking technology publicly and reliably on every website where the plugin is correctly implemented,

3) declares tracking pixels.

However, many CMPs which declared the purpose of each tracker don't specify the kind of tracker. According to their assessment, they choose CMP Cookiebot as it has a high market share and fulfills all three conditions above. They found that pixels are small part of trackers declared.

They used the same purpose categories for the tracking pixels as used for cookies which correspond to the categories defined by UK's International Chamber of Commerce. Cookiebot also use these four categories to classify pixels. The privacy threat increases in the order of categories:

1) *Strictly Necessary*: Provide a core functionality for a website.

2) *Functional*: Not essential but can be used to customize a website.

3) *Analytics*: Analyze website performance and they should only track a user on a single domain. However, they can be used to track a user across multiple devices. Google Analytics pixel is a well-known example for an analytics pixel.

4) *Advertising*: Deliver an advertisement of a product. They can be used to track the user across multiple websites. Facebook pixel is an advertising pixel.

Consent crawler was used to collect purpose labels of pixels for domains hosting Cookiebot returned by using presence crawler. Consent crawler was implemented by Dino Bollinger on top of OpenWPM which is a web privacy measurement framework. They collect all the tracking pixels and images monitoring HTTP traffic and filter for images requested and received. Given a list of URLs to crawl, the website's landing page is connected and data is extracted from consent notice. If it's successful, it collects images from the landing page monitoring HTTP traffic. It then accesses five subpages and also collects the images there. From all the unique 10380 Cookiebot domains by the CMP presence crawl, they found, 8722 of them contained consent declarations. They collected 39387-pixel labels from a total of 449955 consent labels. They

collected 1063638 HTTP requests and 900930 HTTP responses for images including pixels.

First, they link a pixel to a consent notice banner by HTTP request of an image. Second, they link the consent notice and observed pixel from HTTP requests to actual pixels returned in the HTTP response. They included all images without matching consent declaration in the 'necessary' category since to block HTTP requests for certain categories of tracking pixels without breaking a websites core functionality, it has to be able to recognize normal images as 'necessary' [5].

Table 2.1 Categories of pixels in the pixel dataset

Label	Number of Samples	Percentage of Samples
Strictly Necessary	619′569	76.2%
Functional	49	<0.1%
Analytics	63′043	7.8%
Advertising	130′501	16.0%

2.2 Exploratory data analysis (EDA)

Exploratory data analysis is an approach of analyzing data sets and performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of statistical graphics and other data visualization methods [14 ,15]. Once EDA is complete, its outputs can then be used for more sophisticated data analysis and machine learning.

I will be using python as a tool to make an EDA. Python is an interpreted, object-oriented programming language with dynamic semantics. Its high-level, built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together [15]. Another tool we will be using is Jupyter Notebook. It is an interactive computational environment, in which you can combine code execution, rich text, mathematics, plots and rich media [16].

Steps of the EDA process:

Load data and import Pandas, NumPy or another similar libraries

```
#import important data manipulation frameworks
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import zlib
```

Figure 2.1 Imported libraries for initial work on the dataset

```
#import pixel dataset
data=pd.read_json('C:\\Users\\Dell\\Downloads\\Training_Data\\training.json')
```

Figure 2.2 Importing the pixel dataset

We first read the dataset using the Pandas read_csv() function and print the 1st ten rows using head (figure 2.3). As the dataset it transposed, we use transpose to see the actual data.

```
pixel=data.transpose()
pixel.head(10)
```

	visit_id	request_id	url	first_party_domain	label	triggering_origin	headers	img
5127	5	127	http://googleads.g.doubleclick.net /pagead/view...	http://www.b-travel.com/	3	http://www.b-travel.com	[["Host","googleads.g.doubleclick.net"], ["User...	
5130	5	130	http://www.google-analytics.com /collect?v=1&_v...	http://www.b-travel.com/	2	http://www.b-travel.com	[["Host","www.google-analytics.com"], ["User-Ag...	
5248	5	248	https://www.facebook.com /tr/?id=26283382541559...	http://www.b-travel.com/	3	http://www.b-travel.com	[["Host","www.facebook.com"],["User-Agent","Mo...	
5319	5	319	https://www.facebook.com /tr/?id=26283382541559...	http://www.b-travel.com/	3	http://www.b-travel.com	[["Host","www.facebook.com"],["User-Agent","Mo...	
5338	5	338	https://www.google-analytics.com /collect?v=1&_...	http://www.b-travel.com/	2	http://www.b-travel.com	[["Host","www.google-analytics.com"], ["User-Ag...	
11107	11	107	https://cms.quantserve.com /dpixel?a=p-n5wLvRd...	http://www.zeno.org/	3	https://pagead2.googlesyndication.com	[["Host","cms.quantserve.com"],["User-Agent","...	
11147	11	147	https://cms.quantserve.com /dpixel?a=p-n5wLvRd...	http://www.zeno.org/	3	https://pagead2.googlesyndication.com	[["Host","cms.quantserve.com"],["User-Agent","...	
11153	11	153	https://ag.innovid.com /trk?tid=11711&google_gi...	http://www.zeno.org/	3	https://pagead2.googlesyndication.com	[["Host","ag.innovid.com"],["User-Agent","Mozi...	
11158	11	158	https://cms.quantserve.com /dpixel?a=p-n5wLvRd...	http://www.zeno.org/	3	https://pagead2.googlesyndication.com	[["Host","cms.quantserve.com"],["User-Agent","...	
11164	11	164	https://ag.innovid.com /trk?tid=11711&google_gi...	http://www.zeno.org/	3	https://pagead2.googlesyndication.com	[["Host","ag.innovid.com"],["User-Agent","Mozi...	

Figure 2.3 First 10 rows of the pixel dataset

We can see the column names using ‘columns’ attribute:

```
pixel.columns
```

```
Index(['visit_id', 'request_id', 'url', 'first_party_domain', 'label',
       'triggering_origin', 'headers', 'img_format', 'img_size', 'img_mode',
       'img_colour', 'matched', 'moved', 'blocked'],
      dtype='object')
```

Figure 2.4 Columns of the pixel dataset

We should check for any missing data:

```
pixel.isnull().sum()
```

```
visit_id              0
request_id            0
url                   0
first_party_domain    0
label                 0
triggering_origin     0
headers               0
img_format            0
img_size              0
img_mode              0
img_colour            0
matched               0
moved                 0
blocked               0
dtype: int64
```

Figure 2.5 There is no missing value in the dataset

We can also check information about the dataset:

```
pixel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 813162 entries, 5127 to 10380996
Data columns (total 14 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   visit_id            813162 non-null  object
 1   request_id          813162 non-null  object
 2   url                 813162 non-null  object
 3   first_party_domain  813162 non-null  object
 4   label               813162 non-null  object
 5   triggering_origin   813162 non-null  object
 6   headers             813162 non-null  object
 7   img_format          813162 non-null  object
 8   img_size            813162 non-null  object
 9   img_mode            813162 non-null  object
 10  img_colour          813162 non-null  object
 11  matched             813162 non-null  object
 12  moved               813162 non-null  object
 13  blocked             813162 non-null  object
dtypes: object(14)
memory usage: 125.3+ MB
```

Figure 2.6 Checking Info of the dataset

We have 4 labels: Strictly Necessary, Functional, Analytics, and Advertising

```
pixel['label'].unique()   #etiket verisi
```

```
array([3, 2, 0, 1], dtype=object)
```

Figure 2.7 Checking target variable `label` of the dataset

Below to see the class imbalance between each label a bar-plot is drawn:

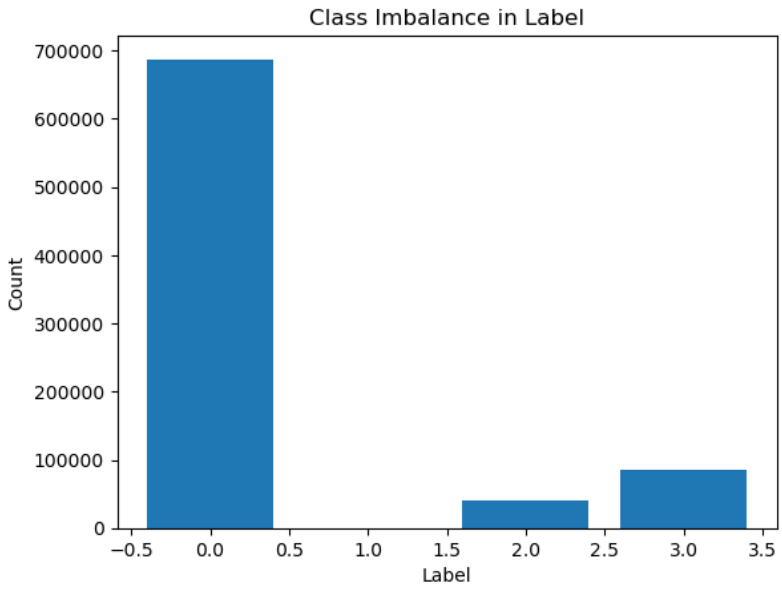


Figure 2.8 Checking the class imbalance with bar-plot

As shown below `label: 1` has only 108 samples which results in class imbalance in this dataset.

```
label_counts=pixel['label'].value_counts()
label_counts
```

```
0    687288
3     85976
2     39790
1       108
Name: label, dtype: int64
```

Figure 2.9 Count of each label in the dataset

Below we can see the image format distribution

```
# Count the occurrences of each img_format
img_format_counts = pixel['img_format'].value_counts()
img_format_counts
```

```
PNG     251940
GIF     242640
JPEG    241161
WEBP     53888
ICO      22699
CUR        694
BMP         90
MPO         47
TIFF         3
Name: img_format, dtype: int64
```

Figure 2.10 Image format distribution in the dataset

2.3 Feature Engineering

The goal of feature engineering is to have uniformly-sized real-valued vector representation for all pixel as this can be used as input to a range of classifiers.

The features that are collected for each image or pixel are the following [5]:

1. The URL of the image resource requested. It may include domain, subdomains, path and different query parameters to the image resource.

2. The headers of the HTTP request. It may contain different header fields.

3. The actual image returned from the HTTP response. The features extracted from that image are:

a) The size as width and height in number.

b) The color in RGBA color values of first pixel.

c) The format (GIF, JPEG, TIFF, MPO, ICO, PNG, WEBP, and WEBP) of the pixel image.

d)The color mode (RGB, CMYK, LA, L, RGBA, 1, P and I) of the image.

4. The URL of the first party.

5. The URL of the triggering origin of the resource whether first or third party.

Below table depicts how the above features are engineered to turn into a real-valued vector representation.

Table 2.2 Feature engineering of pixel dataset

Feature Name	Size	Description
Top Netlocs	500	One-hot vector of the most common netlocs
Top Query	500	Multiple-hot vector of the most common query parameters
Top Path Pieces	500	Multiple-hot vector of the most common 'words' in the path
URL Length	2	Number of character in path and in query part of URL
Has Query	1	Binary indicator, true if the URL contains any query
Query Count	1	Number of query parameters the URL contains
Entropy URL	1	Shannon entropy of the URL
Compressed URL	2	Size and reduction of URL after *zlib* compression
Is blocked	2	Whether it would be blocked by EasyList or EasyPrivacy
Header Fields	4	Binary indicator of presence of a specific header field
Header Length	1	Number of characters in the headers
Entropy Headers	1	Shannon entropy over all headers
Compressed Headers	2	Size and reduction of headers after *zlib* compression
Image Size	2	Width and height of the image in pixels
Is 1x1	1	Binary indicator, true if the image is of size 1x1 pixel
Image Format	9	One-hot vector indicating the format of the image, e.g., GIF
Image Mode	8	One-hot vector indicating the colour mode of the image
Transparency	1	Value of alpha channel of first pixel in RGBA
Image Colour	3	Colour value of the first pixel in RGBA
Top T.O. Netlocs	500	One-hot vector of the most common triggering origin netlocs
Third Party	1	Binary indicator, true if the pixel originates from a third party

Entropy is used because it is a measure of disorder (as well as a measure of purity) and it is computed based on frequentist probability distribution. High level of disorder means low level of purity – if probabilities between special characters are not much different (i.e. 4/7 and 3/7), it has high entropy or low level of purity. Otherwise, low entropy or high purity [18].

Chapter 3

Machine Learning

Machine Learning is a branch of artificial intelligence that develops algorithms by learning the hidden patterns of the datasets used it to make predictions on new similar type data, without being explicitly programmed for each task.

3.1 Invisible Pixel Classifier

The goal of training this classifier is to detect features and hidden patterns in the invisible pixel dataset so that it can be used to predict the purpose of the pixel correctly.

My purpose in this study is to use important features to develop classifier that best predicts the label which will be used later to help developing an invisible pixel browser extension which will show rare pixels in the web site that are not found by filter lists. Some pixels for example are not declared and they can be assigned to a proper category by this model. In addition, wrong assignments can also be found.

Since there are four labels, it is a multiclassification problem. The labels are pre-labeled so supervised machine learning is used.

Supervised learning is classified into two categories of algorithms:

Classification: A classification problem is when the output variable is a category, such as “True” or “False”, “have” or “don’t have”.

Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

The most widely used supervised machine learning algorithms are: (https://www.geeksforgeeks.org/supervised-unsupervised-learning/)

- Support-vector machines.
- Linear regression.
- Logistic regression.
- Naive Bayes.
- Linear discriminant analysis.
- Decision trees.
- K-nearest neighbor algorithm.
- Neural networks (Multilayer perceptron)

In this study four classifiers are compared: Random Forest Algorithm, XGBoost Algorithm, MLP-Artificial Neural Network Algorithm, and Logistic Regression Model.

Ganz [5] trained the dataset with XGBoost model to predict purposes of pixels. They compared classifications by the model, the consent declarations, and the filter lists and found strong agreement to their findings. They also found that there are wrongly blocked normal images and rare tracker missed by filter lists. The model created has three meaningful results. Identification of important features of tracking pixels, assessment and maintenance of filter lists, and detection of potential violations.

3.2 Evaluation Metrics

Once the model is trained using the classifiers, it can be evaluated on the test dataset to determine its accuracy and performance using different metrics. Choosing the right metric is crucial while evaluating machine learning (ML) models. Since a classification is performed, Classification Metrics is used. Classification Metrics evaluate a model's performance and tell how the classification performs, however each of the metrics evaluates performance in a different way.

Accuracy is defined as the number of correct predictions divided by the total number of predictions, multiplied by 100.

Confusion Matrix

Confusion Matrix is a tabular visualization of the ground-truth labels versus model predictions. Each row of the confusion matrix represents the instances in a predicted

class and each column represents the instances in an actual class. Confusion Matrix is not exactly a performance metric but sort of a basis on which other metrics evaluate the results.

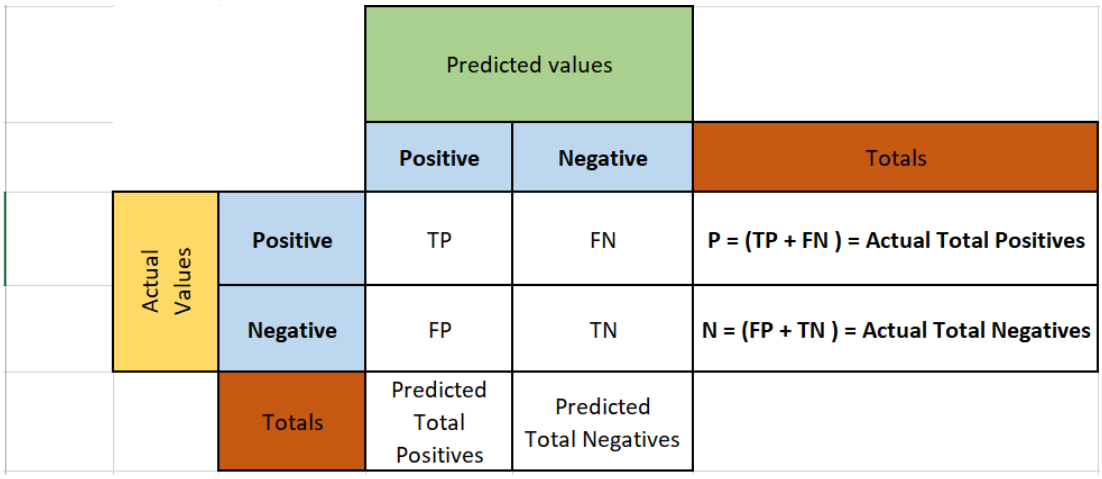


Figure 3.1 Confusion matrix

Each cell in the confusion matrix represents an evaluation factor.

- *True Positive (TP)* signifies how many positive class samples your model predicted correctly.
- *True Negative (TN)* signifies how many negative class samples your model predicted correctly.
- *False Positive (FP)* signifies how many negative class samples your model predicted incorrectly. This factor represents **Type-I error**. This error positioning in the confusion matrix depends on the choice of the null hypothesis.
- *False Negative (FN)* signifies how many positive class samples your model predicted incorrectly. This factor represents **Type-II** error.

Precision and Recall

Precision is the ratio of true positives and total positives predicted:

0<P<1

$$P = \frac{TP}{TP + FP} \qquad (3.1)$$

The precision metric focuses on Type-I errors (FP). A Type-I error occurs when we reject a true null Hypothesis(H^0). So, in this case, Type-I error is incorrectly labeling functional invisible pixel as analytics invisible pixel.

A precision score towards 1 will show that the model didn't miss any true positives, and is able to classify well between correct and incorrect labeling of invisible pixels. What it cannot measure is the existence of Type-II error, which is false negatives – cases when a functional invisible pixel is identified as analytics invisible pixel.

A low precision score (<0.5) means your classifier has a high number of false positives which can be an outcome of imbalanced class or untuned model hyperparameters. In an imbalanced class problem, the data has to be prepared beforehand with over/under-sampling or focal loss in order to curb FP/FN. I know that in my invisible pixel dataset there is an imbalanced class problem.

A *Recall* is essentially the ratio of true positives to all the positives in ground truth.
0<R<1

$$R = \frac{TP}{TP + FN} \qquad (3.2)$$

The recall metric focuses on type-II errors (FN). A type-II error occurs when we accept a false null hypothesis(H^0). So, in this case, type-II error is incorrectly labeling functional invisible pixel as analytics invisible pixel.

Recall towards 1 will signify that the model didn't miss any true positives, and is able to classify well between correctly and incorrectly labeling of invisible pixels.

What it cannot measure is the existence of type-I error which is false positives.

A low recall score (<0.5) means your classifier has a high number of false negatives which can be an outcome of imbalanced class or untuned model hyperparameters. In an imbalanced class problem, the data has to be prepared beforehand with over/under-sampling or focal loss in order to curb FP/FN.

F1-score metric uses a combination of precision and recall. In fact, the F1 score is the harmonic mean of the two. The formula of the two essentially is:

$$F1\ Score = 2\ x \frac{recall\ x\ precicion}{recall + precision} \qquad (3.3)$$

High F1 score symbolizes a high precision as well as high recall. It presents a good balance between precision and recall and gives good results on imbalanced classification problems.

A low F1 score only tells about performance at a threshold. Low recall means we didn't try to do well on very much of the entire test set. Low precision means that, among the cases we identified as positive cases, we didn't get many of them right.

But low F1 doesn't say which cases. High F1 means we likely have high precision and recall on a large portion of the decision (which is informative). With low F1, it's unclear what the problem is (low precision or low recall?), and whether the model suffers from type-I or type-II error.

AU-ROC (Area under Receiver operating characteristics curve)

It makes use of true positive rates (TPR) and false positive rates (FPR).

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN} \qquad (3.4)$$

Intuitively TPR/recall corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. In other words, the higher the TPR, the fewer positive data points we will miss.

Intuitively FPR/fallout corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. In other words, the higher the FPR, the more negative data points will be misclassified.

To combine the FPR and the TPR into a single metric, we first compute the two former metrics with many different thresholds for the logistic regression, then plot them on a single graph. The resulting curve is called the ROC curve, and the metric considered is the area under this curve, which we call AU-ROC.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis. (https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide)

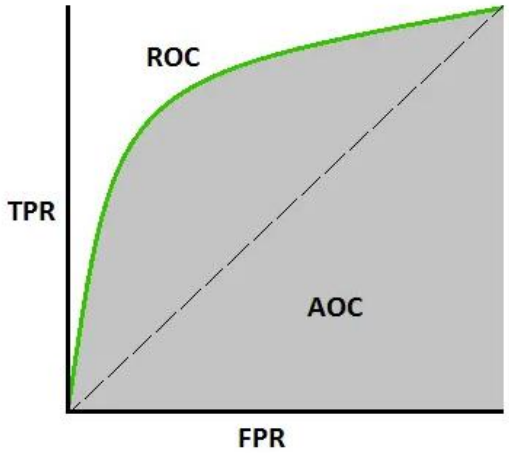


Figure 3.2 ROC curve

An excellent model has AUC near to the 1 which means it has a good measure of separability. A poor model has an AUC near 0 which means it has the worst measure of separability. In fact, it means it is reciprocating the result. It is predicting 0s as 1s

and 1s as 0s. And when AUC is 0.5, it means the model has no class separation capacity whatsoever. (https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5)

3.3 Random Forest Algorithm

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. For classification tasks, the output of the random forest is the class selected by most trees. (https://www.javatpoint.com/machine-learning-random-forest-algorithm)

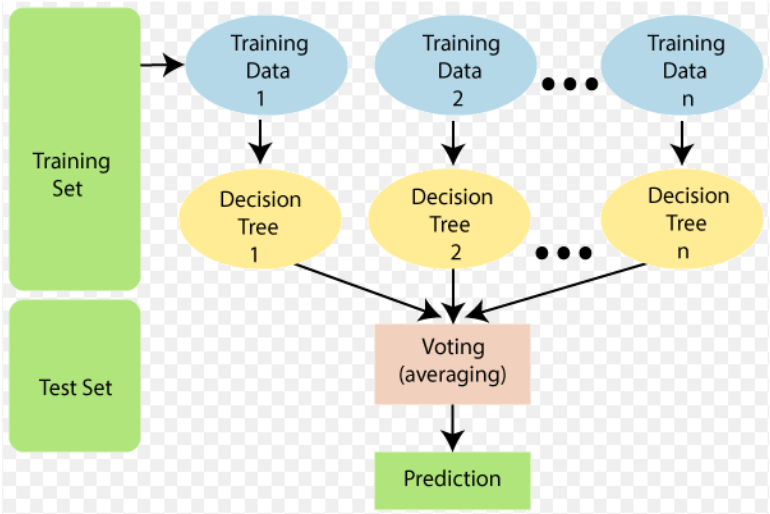


Figure 3.3 Random Forest Algorithm

3.3.1 Random Forest Algorithm Model Evaluation

```
Accuracy: 0.9744270842940854
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.99    137474
           1       1.00      0.84      0.91        19
           2       0.91      0.91      0.91      7936
           3       0.89      0.96      0.92     17204

    accuracy                           0.97    162633
   macro avg       0.95      0.92      0.93    162633
weighted avg       0.98      0.97      0.97    162633
```

Figure 3.4 Random Forest algorithm classification report of the pixel dataset

3.4 XGBoost Algorithm

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting” used for classification problems. One of the key features of XGBoost is its efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing. Additionally, XGBoost has built-in support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time. (https://www.geeksforgeeks.org/xgboost/)

3.4.1 XGBoost Algorithm Model Evaluation

Accuracy: 0.9499671038473126
Classification Report:

	precision	recall	f1-score	support
0	0.98	0.96	0.97	137474
1	1.00	0.79	0.88	19
2	0.76	0.81	0.78	7936
3	0.82	0.94	0.87	17204
accuracy			0.95	162633
macro avg	0.89	0.87	0.88	162633
weighted avg	0.95	0.95	0.95	162633

Figure 3.5 XGBoost algorithm classification report of the pixel dataset.

3.5 MLP-Artificial Neural Network Algorithm

Artificial Neural Networks (ANNs) are computer systems developed to automatically realize skills such as generating new information, creating and discovering new information through learning, which are characteristics of the human brain, without any help. ANNs can model nonlinearly without the need for any assumptions or prior knowledge of input and output variables Multilayer Perceptron (MLP) is one of the most commonly used ANN models for the solution of nonlinear problems. There is at least one layer between the input and output layers in this feed-forward backpropagation network. The relation weight values between the layers are updated to decrease the calculated error value in the reverse propagation stage when the network's output and error value are measured in the forward propagation stage. (https://dergipark.org.tr/en/download/article-file/1699616)

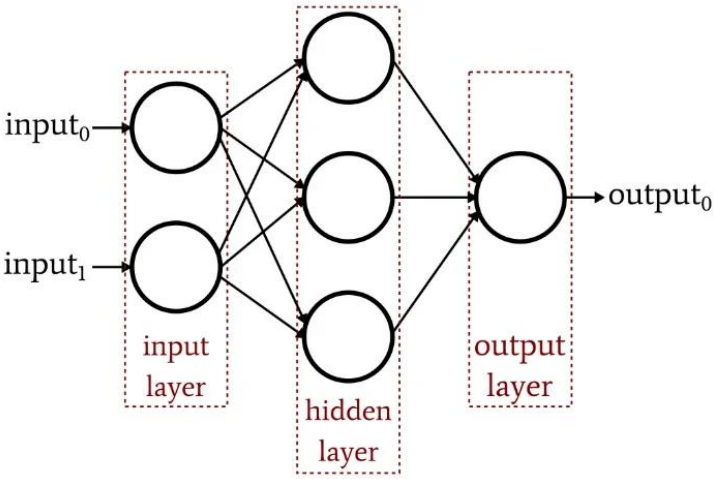


Figure 3.6 ANN structure

3.5.1 MLP-Artificial Neural Network Algorithm Model Evaluation

Accuracy: 0.9536317967448181
Classification Report:

	precision	recall	f1-score	support
0	0.98	0.96	0.97	137474
1	0.50	0.21	0.30	19
2	0.80	0.79	0.79	7936
3	0.82	0.96	0.88	17204
accuracy			0.95	162633
macro avg	0.77	0.73	0.74	162633
weighted avg	0.96	0.95	0.95	162633

Figure 3.7 MLP-Artificial Neural Network algorithm classification report of the pixel dataset

3.6 Logistic Regression Algorithm

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class. It is used for classification algorithms its name is logistic regression. it's referred to as regression because it takes the output of the linear regression function as input and uses a sigmoid function to estimate the probability for the given class. The difference between linear regression and logistic regression is that linear regression output is the continuous value that can be anything while logistic regression predicts the probability that an instance belongs to a given class or not. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). In Multinomial Logistic Regression, the output variable can have more than two possible discrete outputs. In this case, the SoftMax function is used in place of the sigmoid function. (https://www.geeksforgeeks.org/understanding-logistic-regression/)

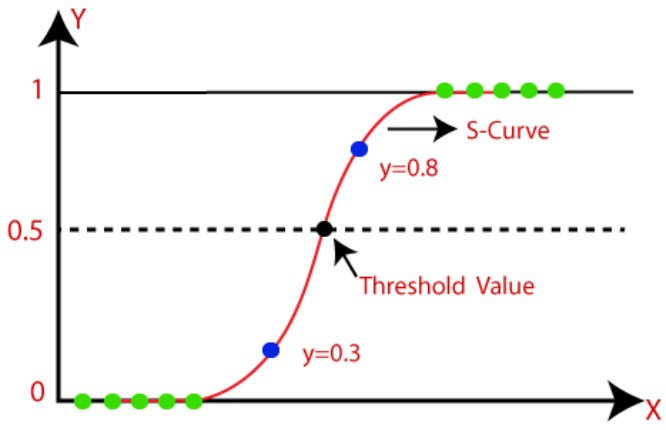


Figure 3.7 S-shaped logistic function

3.6.1 Logistic Regression Algorithm Model Evaluation

```
Accuracy: 0.9481470550257328
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.96      0.97    137474
           1       0.00      0.00      0.00        19
           2       0.77      0.74      0.76      7936
           3       0.81      0.94      0.87     17204

    accuracy                           0.95    162633
   macro avg       0.64      0.66      0.65    162633
weighted avg       0.95      0.95      0.95    162633
```

Figure 3.8 Logistic regression algorithm classification report of the pixel dataset.

According to the classification metrics Random Forest Algorithm has the best performance as it is proven that it works well for multi-class classification tasks and can handle a large number of features and data points effectively. In order to use this model for deployment I will save it.

Chapter 4

Model Deployment and Conclusion

4.1 Converting and saving the model to deploy in browser extension

Browser extensions are composed of a collection of semi-independent elements. Which are powerful components of browsers since they can view and modify every element in the browser while it is an independent component.

Once the model is trained and tuned, it can be deployed in a production environment to make predictions on new data. This step requires integrating the model into an existing browser extension code.

4.1.1 Convert the scikit-learn model to TensorFlow.js format

TensorFlow.js was developed by Google to let JavaScript developers develop machine learning models.

To convert a scikit-learn model to TensorFlow.js, the sklearn-porter library is used to export the model to JavaScript code, and then TensorFlow.js is used to load the model. The exported TensorFlow.js model can now be used to perform predictions in browser extension's JavaScript code.

4.1.2 Load the TensorFlow.js model in browser extension

Include the model.js file in your browser extension's files and load it in content script or background script using a script tag.

```
<script src="model.js"></script>
```

4.2 Conclusion

My goal in this project is to perform exploratory data analysis and feature engineering of the dataset, using a previously created invisible pixel dataset, and eventually create a machine learning model that classifies these pixels and convert this machine learning model to suitable form to be used in a browser extension that can be developed.

I used an invisible pixel dataset to classify pixels according to their purpose categories. There were 813162 samples which are used to train four classifiers: Random Forest Algorithm, XGBoost Algorithm, MLP ANN Algorithm and Logistic Regression Algorithm. The best performing model is found to be Random Forest Algorithm which achieved a high accuracy (97.44%).

Using Random Forest Algorithm, I showed how to deploy the model in a browser extension converting the scikit-learn to tensorflow.js and loading the tensorflow.js model in a browser extension script.

References

[1] Third parties [Internet]. [Erişim tarihi 26.03.2023]. [https://web.dev/learn/privacy/third-parties/].

[2] Privacy and Innovation. Avi Goldfarb & Catherine Tucker. Innovation Policy and the Economy. 2012: 65-90. https://www.nber.org/papers/w17124

[3] The Electronic Frontier Foundation. Behind the One-Way Mirror: A Deep Dive into the Technology of Corporate Surveillance [İnternet]. [erişim tarihi 26.03.2023]. https://www.eff.org/wp/behind-the-one-way-mirror.

[4] Summary of the Board Decision on "the processing of personal data by the data controller operating in e-commerce sector through cookies used by the websites/mobile applications" [İnternet]. [erişim tarihi 26.03.2023]. https://www.kvkk.gov.tr/Icerik/7408/Summary-of-the-Board-Decision-on-the-processing-of-personal-data-by-the-data-controller-operating-in-e-commerce-sector-through-cookies-used-by-the-websites-mobile-applications-

[5] Ganz, Rita. Understanding GDPR compliance of tracking pixel declarations using filter lists (bachelor`s thesis). Zurich: ETH Zurich; 2022. https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/535362/Thesis__Tracking_pixels_RitaGanz.pdf?sequence=1&isAllowed=y

[6] Fouad, Imane & Bielova, Nataliia & Legout, Arnaud & Sarafijanovic-Djukic, Natasa. (2020). Missed by Filter Lists: Detecting Unknown Third-Party Trackers

with Invisible Pixels. Proceedings on Privacy Enhancing Technologies. 2020. 499-518. 10.2478/popets-2020-0038.

[7] Fouad, Imane & Bielova, Nataliia & Legout, Arnaud & Santos, Cristiana. My Cookie is a phoenix: detection, measurement, and lawfulness of cookie

[8] respawning with browser fingerprinting Proceedings on Privacy Enhancing Technologies; 2022 (3):79–98

[9] An overview of HTTP. [Internet]. [Erişim tarihi 26.03.2023]. [https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview/].

[10] Web Security 1: Same-Origin and Cookie Policy. [Internet]. [Erişim tarihi 26.03.2023]. https://inst.eecs.berkeley.edu/~cs261/fa17/scribe/web-security-1.pdf

[11] What is Cookie Syncing and How Does it Work? [Internet]. [Erişim tarihi 26.03.2023]. https://clearcode.cc/blog/cookie-syncing/

[12] Preparing for the end of third-party cookies.? [Internet]. [Erişim tarihi 26.03.2023]. https://www.youtube.com/watch?v=gm8O8-b2B8c

[13] Tracking user activities using web bugs. 2013. [Internet]. [Erişim tarihi 26.03.2023]. https://resources.infosecinstitute.com/topic/tracking-user-activities-using-web-bugs/

[14] Exploratory data analysis. [Internet]. [Erişim tarihi 26.03.2023]. https://en.wikipedia.org/wiki/Exploratory_data_analysis

[15] What is exploratory data analysis? [Internet]. [Erişim tarihi 26.03.2023]. https://www.ibm.com/topics/exploratory-data-analysis

[16] The Jupyter Notebook. [Internet]. [Erişim tarihi 26.03.2023]. https://ipython.org/notebook.html

[17] Ruohonen, Jukka & Leppänen, Ville. (2018). Invisible Pixels Are Dead, Long Live Invisible Pixels! 28-32. 10.1145/3267323.3268950.

[18] Entropy: How Decision Trees Make Decisions. [Internet]. [Erişim tarihi 26.03.2023]. https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

Curriculum Vitae

Name Surname : Işıl Işık Mutlu

Education:

1997–2002 Hacettepe University, Dept. of Nutrition and Dietetics

Work Experience:

2003 – 2023 Türkiye Cumhuriyet Merkez Bankası