PREDICTING TRENDING ELEMENTS ON WEB PAGES USING EYE-TRACKING DATA

A THESIS SUBMITTED TO
THE BOARD OF GRADUATE PROGRAMS
OF
MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS

BY

NAZIHA SHEKH KHALIL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING PROGRAM

AUGUST 2022

Approval of the Board of Graduate Programs

Prof. Dr. Cumali Sabah
Chairperson

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Assoc. Prof. Dr. Enver Ever
Program Coordinator

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Dr. Şükrü Eraslan
Co-Supervisor

Assoc. Prof. Dr. Yeliz Yeşilada
Supervisor

Examining Committee Members

Asst. Prof. Dr. Meryem Erbilek
Computer Engineering, METU NCC

Assoc. Prof. Dr. Yeliz Yeşilada
Computer Engineering, METU NCC

Assoc. Prof. Dr. Melike Şah Direkoğlu
Computer Engineering, NEU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Naziha, Shekh Khalil

Signature :

ABSTRACT

Predicting Trending Elements on Web Pages Using Eye-Tracking Data

Khalil, Naziha Shekh
Master of Science, Computer Engineering Program
Supervisor: Assoc. Prof. Dr. Yeliz Yeşilada
Co-Supervisor: Dr. Şükrü Eraslan

August 2022, 86 pages

Eye-tracking data can be used to understand how users interact with web pages and such understanding can be facilitated for different purposes, for example, it can be used to support better accessibility for disabled users or better usability for all users. However, understanding the sequential behavior of a group of users is challenging from eye-tracking data because individual eye movement sequences, i.e. scanpaths, tend to be complicated and different from each other. Previous work proposes an algorithm called Scanpath Trend Analysis (STA), which brings multiple individual eye movement sequences together and identifies a single representative sequence as a trending path. However, to determine such a path on a web page, an eye-tracking experiment on that page is required. We aim to investigate whether we could identify a trending path on a web page without a new eye-tracking dataset. This thesis shows the experiments towards predicting trending elements without an eye-tracking dataset using different machine learning classifiers based on web page features. We used two pre-collected eye-tracking datasets from previous research to validate the experiments. The results demonstrate that the k-nearest neighbors (KNN) classifier can achieve, from the first dataset, an average F1 score of 0.91 for the browsing task, and an average F1 score of 0.88 for the searching task. With the second dataset, similar results are acquired for the browsing task, however, the synthesis task was not as successful as the browsing task results. Overall, the results show the possibility of predicting the trending elements without using eye-tracking data.

Keywords: Eye-tracking, Trending path, Scanpath trend analysis, Area of interest, Machine learning

ÖZ

KULLANARAK WEB SAYFALARINDA TREND ELEMANLARINI TAHMİN ETMESİ GÖZ İZLEME VERİLERİ

Khalil, Naziha Shekh
Yüksek Lisans, Bilgisayar Mühendisliği Programı
Tez Yöneticisi: Doç. Dr. Yeliz Yeşilada
Ortak Tez Yöneticisi: Dr. Şükrü Eraslan

Ağustos 2022, 86 sayfa

Göz izleme verileri, kullanıcıların web sayfalarıyla nasıl etkileşime girdiğini anlamak için kullanılabilir. Özelllikle, engelli kullanıcılar için daha iyi erişilebilirlik veya tüm kullanıcılar için daha rahat bir kullanımı desteklemek gibi farklı amaçlar doğrulutusunda kullanılabilirler. Ancak her bireyin göz izleme dizileri birbirinden farklı veya karmaşık bir yapıda olma eğilimindedir ve göz izleme verilerini kullanarak bir grup kullanıcının hareketlerini anlamak oldukça zor olabilir. Önceki çalışmalar, birden fazla bireysel göz hareketi dizisini bir araya getiren ve temsili bir trend yol tanımlayan Scanpath Trend Analysis (STA) adlı bir algoritma önermektedir. Ancak bir web sayfasında böyle bir yol belirleyebilmek için o sayfa üzerinde göz izleme verisinin olmasını gerektirmektedir. Biz yeni bir göz izleme veri seti olmadan bir web sayfasında bir trend yol belirleyip belirleyemeyeceğimizi araştırmayı amaçlıyoruz. Bu tezde, web sayfası özelliklerine dayalı farklı makine öğrenimi sınıflandırıcıları kullanarak göz izleme veri seti olmadan trend olan öğeleri tahmin etmeye yönelik deneylerimizi sunuyoruz. Deneyleri doğrulamak için önceki araştırmalardan toplanmış iki göz izleme veri seti kullandık. İlk veri seti k-nearest neighbors (KNN) sınıflandırıcısı için elde edilen sonuçlar tarama görevi için ortalama F1 puanı 0.91 ve arama görevi için ortalama F1 puanı 0.88 elde edebileceğini göstermektedir. İkinci veri seti ile tarama görevi için benzer sonuçlar elde edilmiştir. Ancak sentez görevi tarama görevi sonuçları kadar başarılı olmamıştır. Sonuçlar genel olarak incelendiğinde göz izleme verilerini kullanmadan trend yol öğelerini tahmin etme olasılığının olduğunu göstermektedir.

Anahtar Kelimeler: Göz izleme, Trend yol, Scanpath trend analysis, Makine öğrenme

I dedicate this thesis to my parent, for all their love and support

ACKNOWLEDGMENTS

I wish to express my sincerest thanks and gratitude towards my advisors Professor Dr. Yeliz Yeşilada and Dr. Şükrü Eraslan for their unlimited support, and continuous encouragement, and guidance through my years of study to achieve this stage of education. I also would like to extend my thanks to my dear parents Rida and Nihal for their support and standing behind my back during my study. Finally, special thanks to my brothers Karim and Khalil for their help.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

STA	Scanpath Trend Analysis
AOI	Areas of Interest
VIPS	Vision Based Segmentation
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
SMOTE	Synthetic Minority Oversampling Technique
PET	Predicting Trending Element
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
BR	Bottom Right
BL	Bottom Left
BC	Bottom Center
TR	Top Right
TL	Top Left
TC	Top Center
CL	Center Left
CR	Center Right
CC	Center Center

CHAPTER 1

INTRODUCTION

Using websites and online applications nowadays plays a crucial role in our lives, especially during the recent pandemic, most of our daily tasks are conducted remotely via the web. For understanding user interactions on the web, eye-tracking research has been widely used. These studies allow us to locate and understand where the users are looking in a web page and also what is drawing the user's attention. Understating such information helps the developers in designing web pages. For instance, eye-tracking can be used to understand what attracts users' attention on a web page, and then the web page content can be reorganized accordingly so that it becomes more usable and accessible to disabled users [45]. In addition, these eye tracking studies are also used to analyze the user's viewing patterns. Some studies, in particular, are conducted to understand the reading patterns of users. This information help in understanding how the user will browse a web page and how the user will search for a specific item in a web page. The study by Nielsen [29] observes users' reading patterns by analyzing heatmaps from eye-tracking studies. These heatmaps display the areas that attract users' attention through a range of warm and cold colors depending on the attraction. According to Nielsen [29], users mainly follow the F-shaped pattern where the reading begins from the top-left corner of the page and move horizontally. This case is observed for the left-to-right languages and the opposite case happens when the language is right-to-left. However, Pernice [33] indicates that the F-shaped pattern is not the only pattern followed by users; other forms such as: Layer-cake pattern, Spotted pattern, Marking pattern, Bypassing pattern, and Commitment pattern are also followed. The Layer-cake pattern is where users pay attention to page's headings and subheadings. The Spotted pattern is where users focus on specific words or chunks of words on a page [33]. The Marking pattern is a common reading pattern for mobile devices than on desktop computers, where the user focuses at one place as the mouse scrolls or the fingers swipe [33]. The Bypassing pattern is a reading pattern where users intentionally skip the first words of multiple lines of text when they begin with the same word in a list [33]. The Commitment pattern happens when users are fixating on almost everything they observe in the web page [33].

There have been also studies in understanding the eye movement sequences (i.e. scanpaths) that show the order of the elements visited (i.e. fixated) on the page. However, analyzing the scanpaths of multiple users is challenging since individual scanpaths tend to be complicated and different from each other. Various algorithms are proposed to analyze multiple scanpaths [8]. Some of these algorithms have been used to identify patterns within given scanpaths [43, 19, 5]. The tool developed in [43], which is the eyePatterns tool, finds exact patterns that include a minimum of three elements in individual scanpaths. This approach does not have any tolerance to extra elements within patterns. The SPAM algorithm mentioned in [19, 5] is similar to eyePatterns tool but it is tolerant to extra elements, however it is affected by the positions of elements in individual scanpaths.

Further, other scanpaths approaches have focused on the identification of a single representative path followed by all users [36, 21, 20, 16]. However, these approaches are likely to produce short paths which tend to be limited for further processing of web pages. For example, Goldberg and Helfman [16] propose to use the Dotplots algorithm to hierarchically cluster multiple scanpaths into one scanpath, which causes to lose some shared elements while combining the paths. Besides, Holsanova et al. [21] propose to use a position-weighted model which considers the first visits of users to rank the elements. The drawback of this approach is that some elements can have multiple visits while others can have no visits, and this makes it not applicable for identifying a single path for multiple users.

To overcome the weaknesses of the previous approaches, Scanpath Trend Analysis (STA) is developed to identify a single representative path for multiple scanpaths. The STA algorithm provides the most representative path by having the highest similarity to the individual scanpaths compared to other approaches [10]. Before applying the STA algorithm, a web page is segmented into its visual elements using a web-page segmentation algorithm. The STA algorithm uses an eye-tracking dataset to create individual scanpaths in terms of these visual elements. For example, if the user visits the elements A, B and C respectively, then his/her scanpath is generated as ABC by keeping how long each of them is visited. After that, the trending elements are identified by analyzing their occurrence frequencies and durations, and then used to construct a trending path based on their overall positions in the individual scanpaths. This algorithm was validated with different web-page segmentation approaches [11] and with a different number of individual paths [12, 9]. The full description of the STA algorithm can be found in [10]. This algorithm has been used for various purposes, such as autism detection [14], and identifying common programming code reading patterns [37]. However, the STA algorithm requires a pre-collected eye-tracking dataset.

The work proposed in this study aims to identify whether it is possible to learn and predict the trending path using machine learning without collecting eye-tracking data. It is well known that eye-tracking studies are costly as they need to be conducted systematically with many users that can take a lot of time [9, 12]. In this thesis, we present our work on predicting trending elements of web pages without eye-tracking data, which will be used later to predict a trending path. For experimentation, we use a two pre-collected dataset in this thesis. The first dataset was collected from 40 participants over six different web pages by giving the participants two different tasks which are: a browsing task and a searching task. The second dataset was collected from 19 participants over eight different web pages by giving the participants two different tasks which are: a browsing task and a synthesis task. The trending elements were already identified in the previous research for the first dataset in [10] and for the second dataset in [13]. We conducted different experiments on the dataset with different machine learning classification algorithms. The results showed that k-nearest neighbors (KNN) classifier is the most successful one. We also conducted experiments on the KNN models, such as hypertuning for the parameters and sampling techniques to try to improve the prediction. To evaluate the success of our model, we compared the results with a baseline Random Prediction Algorithm which predicts a random outcome [31].

Identifying trending elements without using an eye-tracking dataset has not been done before; therefore, this work is novel. Automatically identifying trending elements can have many applications. For instance, it can be used to improve the accessibility of web pages by detecting trending elements without eye-tracking data, and these elements can be used to process web pages to make them more accessible for users with disabilities [45, 18].

1.1 Aims and Objectives

The work proposed in this thesis aims to identify whether it is possible to learn and predict the trending elements of a trending path generated by Scanpath Trend Analysis (STA) using machine learning techniques without having to collect eye-tracking data. This thesis has these main objectives:

- Investigate and understand the characteristics of visual elements on web pages that contribute to being in the trending path. Further, identify the different features used in previous research to determine what factors of an element contribute to it being salient. In addition, identify and collect various features from web pages by understanding the trending elements used in our dataset. This will enable us to automatically predict the trending element of the trending path of a web page.

- Investigate and understand the different algorithms and approaches from previous research that could be used to automatically identify the trending elements in a web page. In addition, investigate the different machine learning algorithms to determine the best predictive model for predicting the trending elements.

- Develop an approach that can automatically retrieve features from web pages, prepare the data for training algorithms and perform predictions for the trending element using machine learning algorithms.

1.2 Contributions

The contribution of this study is to identify the trending elements of the trending path generated by STA by using features from web pages without relying on eye-tracking data. Previous works on attention prediction have been limited to relying on using eye-tracking datasets.

We also identified the different features of a web page that could have an influence an element to be trending. We should also remark that none of the of existing works attempt to predict the trending elements.

This study presents a detailed approach for extracting features from different web pages and constructing a predictive model to predict the trending elements, which can be later on used to predict the trending path generated by STA.

While most previous works have mainly focused on using one machine learning model, in this thesis, we have considered and tested different classification models to determine the best classifier for our two different datasets. Our results demonstrated that KNN model is the best predictive model for our datasets.

Identifying the trending elements without conducting an eye-tracker study can immensely help future works. This work can be used to transcode web pages by removing the non-trending elements or by removing elements which can be distracting to people with cognitive disability [38]. This will also make it easier to access web pages with screen readers [45], and therefore will make them more accessible to users with disabilities. In addition, trending elements can give insights to web developers on where to place the advertisement on a web page by determining the areas of a web page that are more attracting to users [30].

1.3 Thesis Outline

Chapter 2 Literature Review: Gives a detailed review of the previous conduction work of identifying the areas of interest (AOI) and sequence prediction. Also, discusses the different features used and models and highlights the gaps in the literature.

Chapter 3 Methodology: Explains the datasets used in this study and the process used to extract features from web page. In addition, gives the detailed process of constructing the predictive model and determining the best model for our datasets.

Chapter 4 Results and Discussion: Highlights the performance of different machine learning classifiers and determine the best performing model. In addition, it shows a detailed comparison of the results of the best model to the baseline. Also, it reports the results of experimenting with different sampling techniques. Finally, it gives a detailed discussion about the results.

Chapter 5 Conclusion: Gives a summary of the work that has been conducted to identify the trending elements of a web page using machine learning models. Also, it explains the limitations as well as further directions that can be investigated in future works.

CHAPTER 2

LITERATURE REVIEW

This chapter gives a summary of the previous work which has been conducted regarding attention prediction and sequence prediction. There has been substantial previous work on attention prediction which is mainly the research area investigating which visual elements in the page attracts users' attention. Some of the prediction work aim to identify the visual elements, which are also known as Areas of Interest (AOI), whereas some of them focus on identifying in which order these elements attract users' attention. The literature review is organized into three: predicting AOI, predicting sequence and predicting sequence and AOI. Table 2.1 gives a broad overview of the existing work.

Table 2.1 A broad overview of attention prediction work

Ref.	AOI	Sequence	Features	Target	Model
[3]	✓		Gaze and visual object locations	Translate gaze coordinates into viewed objects	Viewed Object Detection Algorithm
[35]	✓		Image	Predict attention within web page	Attentional priority model with conventional map (AP)
[23]	✓	✓	Intensity contrast, chromatic contrast, the size of image	Predict the locations of the most attended pictorial information	Custom model
[26]	✓		HTML and mouse features	Discover the elements that are likely to attract user attention.	Mixture of interactions and Content Salience model (MICS)
[6]	✓		HTML and positional features	Investigate which features affect predicting AOI in web page	Machine learning
[15]		✓	Searching (motion, size, images, color, text style, position) Scanning (area, proximity & reading order)	Predict visual attention sequence	Visual hierarchy tool
[36]	✓	✓	Motion, size, color, contrast, brightness, media type	Predict heat maps of visual attention	Web Page Analyzer tool (WPA)
[40]		✓	Image features	Predict attention on web page images	Machine learning (Multilabel Classification)
[42]		✓	Image features	Predict the assigned effective visual attention FI on images	Machine learning
[44]		✓	Image features	Predict a sequence of successive fixations for each web page image	Webpage-based saccadic model (deep learning model)
[41]	✓	✓	Text and image features	Predict the ordinal visual attention on an element	Machine learning

2.1 Predicting AOI

There have been different approaches for predicting Areas of Interest (AOI) in web pages. Some approaches relied on developing their own models to identify the AOI, such as in [3, 35, 23, 26, 36]. Other approach tried to use machine learning as in [6, 41].

Different kinds of features are used to predict AOIs [3, 35, 23, 26, 36, 41]. The work presented by Alam and Jianu [3] relies on having eye-tracking data, and they tested three approaches for object detection: AOI-Based Viewed Object Detection, Probabilistic Viewed Object Detection Approach, and Predictive Viewed Object Detection Approach. In the AOI-Based Viewed Object Detection approach, object shapes are treated as dynamic AOIs, and the most recent fixation in the AOI is defined as the most recently viewed object. The Probabilistic Approach interprets gaze data to determine the probability that objects have been viewed instead of certainty by computing object gaze scores. The Predictive Approach incorporates gaze data, visual object location, and prediction score, which is computed by determining the probability that an object will be viewed if another object was seen just before it (e.g., its neighbor) to identify the object of interest. Further, in Alam and Jianu [3], Alam and Jianu [3] gave their participants two tasks, structured and unstructured, and they were given a specific time to complete the task. However, in order for the algorithms mentioned in Alam and Jianu [3] to be used to predict AOI, gaze data needs to be collected from an eye tracker, which can be costly and time consuming since a separate eye-tracking session needs to be allocated for each participant[9, 12]. Still [35] proposes another approach to predict AOIs by using the screenshots of web pages and enhanced the Saliency model in [22] by adding conventional maps, which include the history of contact interactions learned from common design patterns which are top-left bias, bottom-up bias, and center bias. By adding the conventional maps, the predictive model performs better than the original Saliency model. The predictive model is tested with a different set of images (traditional nature scenes, image web pages, equal-parts image and text web pages, and mainly text web pages). The results showed that the model predicts better on equal-parts image and text web pages, but this makes it a limitation for the study, since not all web pages structures come in the form of equal-parts of image and text. Lagun and Agichtein [26] presents a model called Mixture of Interactions and Content Salience (MICS). Their work looks at content that describes how elements are displayed on a web page, and it looks at interaction features that represent cursor movement features. In this work, the content features that they have collected are features related to the HTML tag. In addition, they also collected size and text related features. Further, they also kept track of the time it took the page to load. For the interaction features, they kept track of cursor movement features such as cursor position, speed, binary features if the cursor is hovering on the element or user is clicking on the element. In addition, Lagun and Agichtein [26], segmented web pages using two approaches: rule-based segmentation and classifier based segmentation. Rule-based segmentation was used for frequent page types where they manually engineered segmentation. Whereas classifier based segmentation was used for less frequent pages, where the classifier will determine whether to segment a page or not. The MICS model was compared to two baseline models: Linear Regression and Non-Linear Regression with Kernels. These baseline models are commonly used for estimating the gaze position from cursor interaction features. The MICS model performed better where it can find elements that are likely to attract users' attention. However, their rule-based segmentation approach that automatically identifies the visual elements on a page tends to be limited to a pre-defined HTML template or layout.

The work of Buscher et al. [6] is also related to predicting AOIs. Their goal was to understand how people viewing a web page can give insight into the important AOIs. They try to investigate which features affect predicting AOIs by relating the eye-tracking features to the page's underlying source code. In their work, they construct a predictive model which takes HTML and rendering related features to predict AOIs. Furthermore, Sutcliffe and Namoun [36] proposed a model of structured directed visual attention and developed a tool called Web Page Analyser (WPA) to predict heatmaps. This tool considers the web page structure where web pages have been grouped into the following categories: graphically and animation intensive web pages, only text web pages, mixed (image and text) with small images, and mixed (image and text) with large images. In addition, the model considers different features such as motion, size, color contrast, brightness, and media type. This tool was used to predict heatmaps for the web page for both searching and browsing modes. The model was able to better predict the heatmap for the searching task since the searching task was narrowed down when compared to the browsing task.

The existing AOI prediction work has been limited to either using eye tracking data to identify the elements or being restricted to a specific website structure, as discussed above. Other approaches have been able to provide a better prediction for a specific task, which also makes it limited. Our approach tries to predict the trending elements generated by STA without using eye tracking data or being restricted to a specific web page structure.

2.2 Predicting Sequence

There has also been some work which focus on sequence prediction [23, 15, 36, 42, 39, 44, 41]. One model is created to predict the order of viewing based on visual hierarchy rules [15]. The model is designed based on specific rules that rank the different features for the searching and the scanning phase. The searching phase represents discovering the important elements whereas the scanning phase observes the area which contains the important elements and extracts the information. However, it is not tested for validity whether the predicted scanpath matches the one recorded from an eye tracker. Grier [17] applies different tests on the visual hierarchy model suggested in [15] to observe the validity of the sequence prediction and what are the important features which grab the user attention. Vidyapu et al. [40] studied how to predict the fixation index (the number which indicates the position of the element within a scanpath) on a web image by using a binary support vector machine. Vidyapu et al. [40], collected different features for images, such as color-related features, positional features, and contrast-related features. Further Vidyapu et al. [40], applied features selection and computed information gain to determine which features are important. In addition, they constructed an SVM model with different kernels: Polynomial, Linear, Hyperbolic, Laplacian, Bessel, ANOVA RBF, and Sigmoid. Vidyapu et al. [40] compared their model results with the work done by [6]. Also, they compared their model with two common baseline algorithms: the Zero rule (ZeroR) model and the Random prediction (RP) model. From these comparisons, their results showed that the SVM model with Gaussian RBF kernel achieved the best performance for predicting fixation index. Vidyapu et al. [40] is advanced in [42] to include weighting strategies schema to make the fixation index prediction based on image importance by introducing four different weight schemas: Uniform weighting, Linear weighting, Proportional weighting, and Inverse proportional weighting. Moreover, there has been another approach in [44] that uses deep learning techniques for prediction fixations. Xia and Quan [44] developed a deep learning model to predict the saliency map of webpage's images for the areas that are more likely to be fixated. In addition, they tried to predict the sequence of fixations on the web images. In their model, to enhance the sequence prediction, they combined the following mechanisms: spatial bias (assumes that the top-left corner is the most informative area of the image) and inhibition of return (orientation mechanism that helps detect an object with greater speed and accuracy after attending to it briefly), see [44] for further details.

There have been several attempts for sequence prediction; some relied either on constructing custom models or using machine learning and deep learning approaches to make sequence predictions. Some of the existing work relied on using specific rules of ranking, based on searching and scanning tasks. However, limiting the rules for sequence prediction based on only two tasks is not enough to generalize the sequence prediction on a web page. Also, some of the existing work focuses on extracting image features to predict the sequence, but websites do not only contain images; and they usually have different structures. Also, none of the existing work tried to predict the sequence generated by STA.

2.3 Predicting AOI and Sequence

There has been some work that tries to predict both AOIs and sequences [23, 41]. Jana and Bhattacharya [23] propose a model that predicts users' attention and sequence of viewing the image objects on a web page. They use the relationship between the features of intensity contrast, chromatic contrast, and the size of the image object to compute the attention factor. The attention factor is then used in their model to make the sequence prediction. However, this model focuses mainly on image objects. Vidyapu et al. [41] provides another approach that is based on a machine learning model where the goal is to predict whether the element is salient or not, and they also try to predict the fixation index. They consider both image and text features in their approach. Before constructing their predictive model, they observed which features have a high information gain and they use that to predict the position of the fixation in the scanpath.

2.4 Summary

In brief, as can be seen from Table 2.1, there are different approaches for attention prediction work that investigate how to identify AOIs and their order, which shows in which sequence or order they are looked at. However, some of these approaches have been limited to using data for an eye-tracker as a feature to make the prediction. Other work has focused on using and analyzing images. Although images in websites could contain important information, this can be limiting since not all websites are image dominated. Also, in the existing machine learning approaches, work has been limited to testing one classifier with different configuration settings. In this thesis, we focus on identifying the trending element which is a part of the trending path generated by STA. We have tested different machine learning classification models to identify these trending elements for different tasks and without limiting our feature set to just images or to a specific website structure.

CHAPTER 3

METHODOLOGY

This chapter outlines the proposed research methodology for predicting trending elements of web pages. It starts by presenting the datasets used in this study and how they were collected. After that, we present the overall method of our proposed approach. Then, we will start explaining the data preparation process of how the features were collected. Further, we will explain the model development by observing feature selection and testing different classification models.

3.1 Dataset

This thesis used two pre-collected eye-tracking datasets from previous research to validate the experiments. Dataset-1 was collected between two universities: Middle East Technical University Northern Cyprus Campus and the University of Manchester in the UK 2013 [10]. Dataset-2 was collected at the the University of Wolverhampton 2018 [13].

3.1.1 Dataset-1

The first dataset was collected from 40 study participants consisting of 20 females and 20 males. Six different web pages (Apple, BBC, Godaddy, Babylon, AVG, and Yahoo) were used. Based on visual complexity ranking visual measured by ViCRAM [27] these web pages have been ranked as follows: low complexity (Apple, Babylon), medium complexity (AVG, Yahoo), high complexity (GoDaddy, BBC). Two different tasks, browsing and searching, were given to the study participants. For the browsing task, the participants were asked to view the web pages without any particular objective for 30 seconds, whereas for the searching tasks, they were asked to find specific information and/or items on the web pages while their eye movements were being recorded in a period of max 120 seconds. For example in the searching task, in the Apple page (See Figure 3.1) participants were asked these two questions: (a) Can you locate the link that allows watching the TV ads relating to iPad mini? (b) Can you locate a link labeled iPad on the main menu?

In order for the participants to answer the above two questions they were required to fixate on specific elements of the Apple web page. These were the elements visited by the participants "I, C, F, H, E, B, G" (see Figure 3.1) in order to complete the searching task.

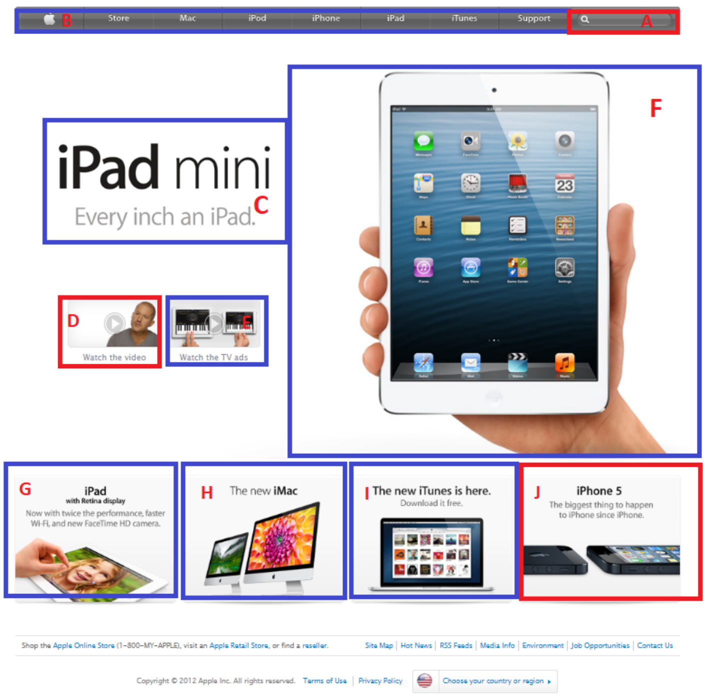


Figure 3.1 Apple web page [10]

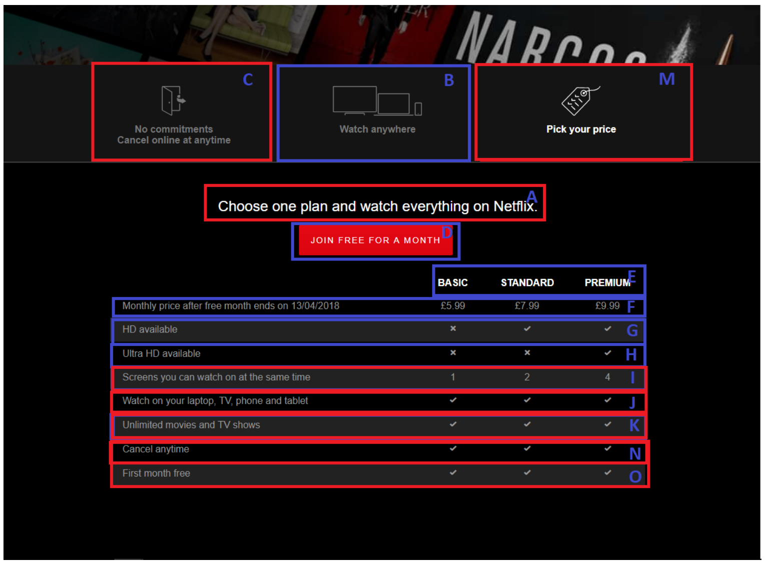
The eye movements of participants were recorded by using the Tobii T60 eye tracker with a degree of accuracy of 0.5 degrees, the screenshots of six different web pages were shown to the participants on a 17-inch monitor with a screen resolution of 1280×1024. The full description of the dataset can be found in [10].

3.1.2 Dataset-2

The second dataset was collected from 19 study participants consisting of 13 females and 6 males. Eight different web pages (Adobe, Amazon, BBC, Netflix, Outlook, WhatsApp, WordPress, and YouTube) were used. Based on visual complexity ranking visual measured by ViCRAM [27] these web pages have been ranked as follows: low complexity (WhatsApp, WordPress, Netflix, Outlook) and high complexity (Amazon, YouTube, Adobe, BBC). Two different tasks namely browsing and synthesis were given to the study participants. For the browsing task, the participants were asked to view the web pages without any particular objective for 30 seconds. In contrast, for the synthesis task, they were asked to answer some specific questions. The study participants needed to combine multiple facts from different elements to provide a new piece of information in a maximum of 120 seconds to answer the questions. For example, participants were given Netflix web page Figure 3.2 and were asked to answer these two questions: (a) Which is the cheapest plan that allows you to watch movies on your laptop, TV and tablet? (b) How much more would you have to pay compared to the basic plan if you wanted to have Ultra HD?

In order for the participants to complete the synthesis task and answer the above questions, these elements "B, D, E, F, G, H" needed to be visited by the participates for the Netflix web page in Figure 3.2.

The eye movements of participants were recorded using the Gazepoint GP3 eye tracker with a sampling rate of 60 Hz and a degree of accuracy of 0.5-1 degree. The screenshots of eight different web pages were then shown to the study participants on a 17-inch monitor with a screen resolution of 1024x768. Full details about the dataset can be found in [13].



Choose one plan and watch everything on Netflix.

JOIN FREE FOR A MONTH

	BASIC	STANDARD	PREMIUM
Monthly price after free month ends on 13/04/2018	£5.99	£7.99	£9.99
HD available	×	✓	✓
Ultra HD available	×	×	✓
Screens you can watch on at the same time	1	2	4
Watch on your laptop, TV, phone and tablet	✓	✓	✓
Unlimited movies and TV shows	✓	✓	✓
Cancel anytime	✓	✓	✓
First month free	✓	✓	✓

Figure 3.2 Netflix web page [13]

3.2 Overall Method

The thesis proposes a machine learning approach for predicting trending visual elements on web pages as summarized in Figure 3.3 which mainly consists of two parts: (1) data preparation ,and (2) modelling.

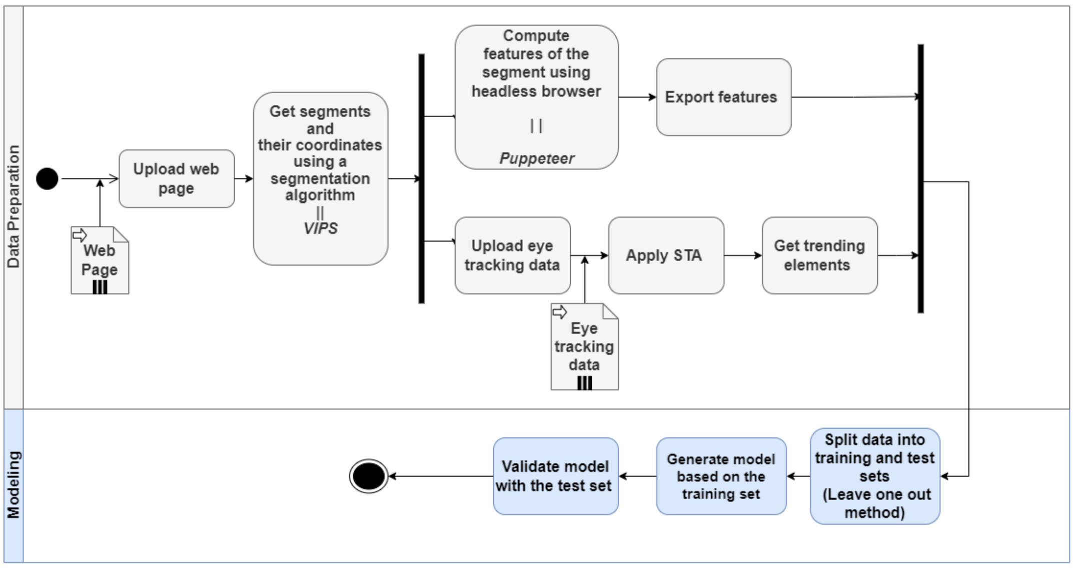

Data Preparation
Modeling
Web Page
Upload web page
Get segments and their coordinates using a segmentation algorithm || VIPS
Compute features of the segment using headless browser || Puppeteer
Export features
Upload eye tracking data
Eye tracking data
Apply STA
Get trending elements
Split data into training and test sets (Leave one out method)
Generate model based on the training set
Validate model with the test set

Figure 3.3 Flow of Predicting Trending Element Approach

Two primary steps are involved in the data preparation: (1) feature extraction and (2) generation of the trending elements by STA (Scanpath Trending Analysis), whereby the web page is uploaded first then segmented. In this thesis, segmentation refers to the process of taking a web page and identifying the visual elements (i.e. areas of interest (AOI) or segments) in that page automatically. In this work, VIPS (Vision-Based Segmentation), is used, as it uses both the underlying source code and the visual rendering of a web page to segment a given web page [2]. The feature extraction stage is responsible for collecting and extracting the features of the segments. The features are collected with a code written in Node JS that uses a headless browser library called Puppeteer [1]. After collecting all the features, the data is exported into a CSV file, and the STA algorithm is used to generate the trending elements using a pre-collected eye-tracking dataset. In this thesis, the pre-collected eye-tracking dataset is only used to obtain the labels of the elements to train and validate our model. The elements generated with VIPS and the trending ones identified with STA algorithm are then sent to the modelling part of the process. This creates a model called Predicting Trending Element (PTE). At this stage, the dataset gets split into train and test sets by using the leave-one-out approach. The test set will contain one web page, and the train set will contain $n-1$ web pages (all pages except the one in the test set). For example, if the dataset contains six different web pages, five will be used for training, and the sixth web page will be used for testing. Afterwards a machine learning model is constructed based on the training set and validated based on the test set.

Once the model is created, the prediction stage starts by uploading a web page and then applying the web page segmentation algorithm (VIPS) to get the segments. The required features are then collected and fed into the PTE model to predict the trending elements as illustrated in Figure 3.4.

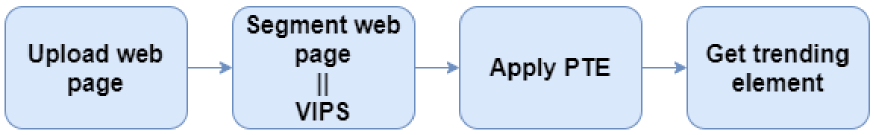


Figure 3.4 Prediction Work Flow

3.2.1 Data preparation

In order for us to determine the required features for predicting the trending elements, we started by analyzing different features that have been previously collected from the literature. The work presented by Buscher et al. [6] have focused on HTML related features and rendering related features. In this work we collected some of the features mention in [6]. In addition, we collected other features based on the observation of how the trending elements differ from non-trending ones.

Features

For each visual element or segment identified on the web page, a set of features are created to represent them. Figure 3.5 shows a web page segmented automatically with the VIPS algorithm where the rectangles represent the web page segments, and each is assigned a unique ID – letters A...X represent the unique IDs. For each segment, we had the URL of the web page as a feature. For further feature creation, we used both the underlying HTML source code and the rendering information from the headless browser. Table 3.1 shows the summary of the group features used in the datasets.

Table 3.1 Summary of the collected features

Group	**Features**
HTML Tags	<div>, <img>, <h1>,.. etc and their count [6] (see Table A.1 for all HTML tags features)
Position	Coordinates: Top, Bottom, Right, Left, Top Center, Bottom Center. Grid Location: Top Left, Top Center, Top Right, Center Left, Center Center, Center Right, Bottom Left, Bottom Center, Bottom Right.
Size	Size [6], Aspect ratio [6], Relative size
Text	Font size, Font weight, Word count
Color	Euclidean distance of color, Standard deviation of color, Background color (exist or not) within a segment and within the web page

For each HTML tag that occurred, for example, for all HTML tags such as <div, <img>, <h1>, <h2>.. etc features are created. In total, we had 32 different HTML el-

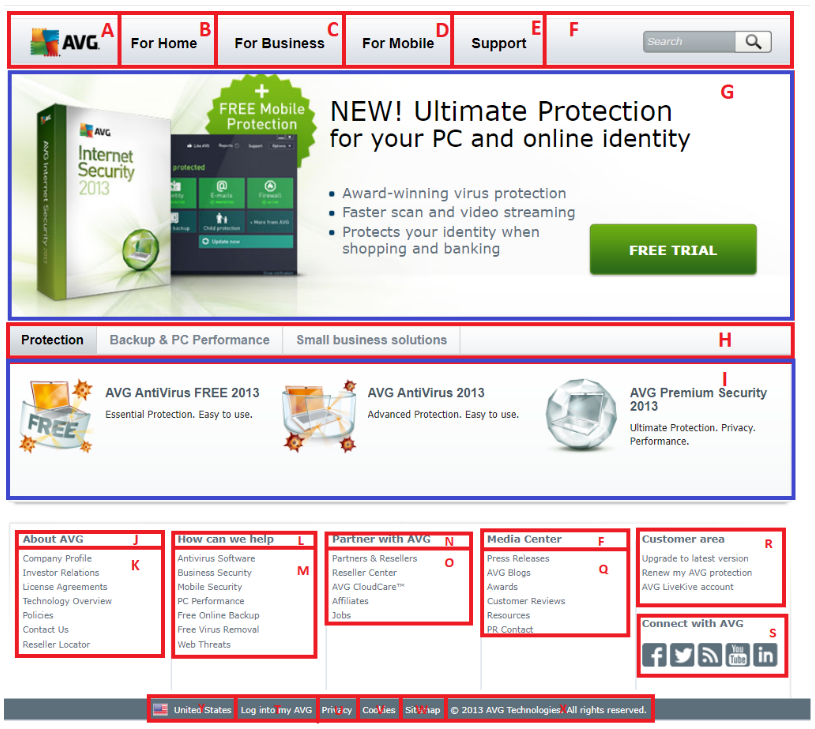


Figure 3.5 AVG web page segmentation [10]

ements that were created as features counting each element in a segment and included that as a feature. In addition to that, we computed positional features where the segment is located on a web page (top left, top center, top right, etc.) by dividing the web page into a 3×3 grid. Figure 3.6 illustrates the grid and how the other rendered features are created. For each segment we also computed Size (width $\times$ height), Aspect ratio $(\min(\text{width,height}) / \max(\text{width}, \text{height}))$ and Relative size of the segment with respect to the whole web page.

This thesis further investigated the trending elements of the different web pages to observe the common properties of trending elements that can be added as features. After observing the trending elements in the different web pages and how they differ from the non-trending elements some text related features were also added. For example, Babylon web page in Figure 3.7 the 'H' segment is one of the trending elements; it can be seen that the segment has a large text size and a heavy font-weight. Another example is the Outlook web page in Figure 3.8, where Segment 'A' and 'B' have a large text size. It can also be seen from some of the web pages that the segments

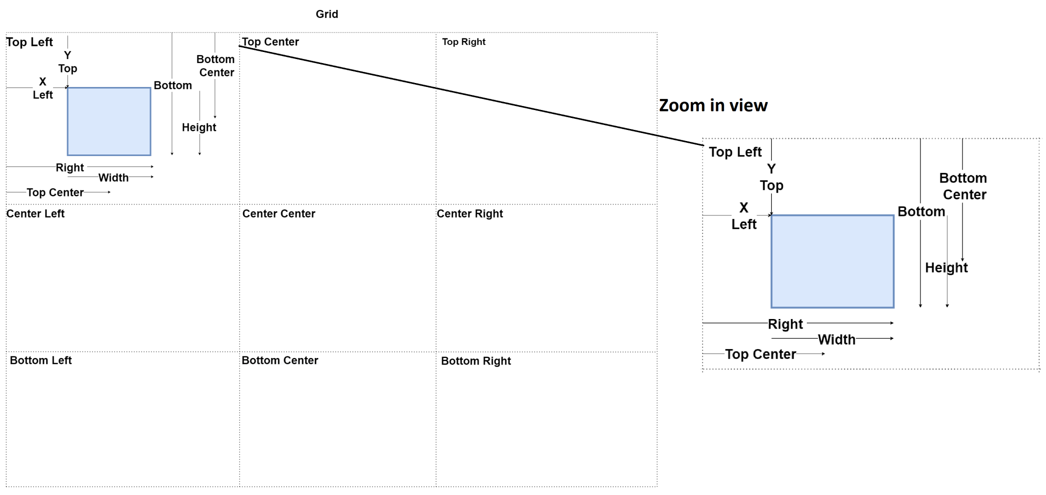


Figure 3.6 Positional features

which contains a lot of words tend to be trending. For instance, on the AVG web page in Figure 3.5 the segment ‘G’ has 26 words and ‘I’ has 25 words. After these observations, text-related features have been added to the dataset, which are binary features for different font size ranges from (XX-small<10px) size to (XX-large>=32px) and weight ranges from (Thin<200) weight to (Ultra-black>=900) (see Table 3.2) for the full sizes and ranges used in the dataset.

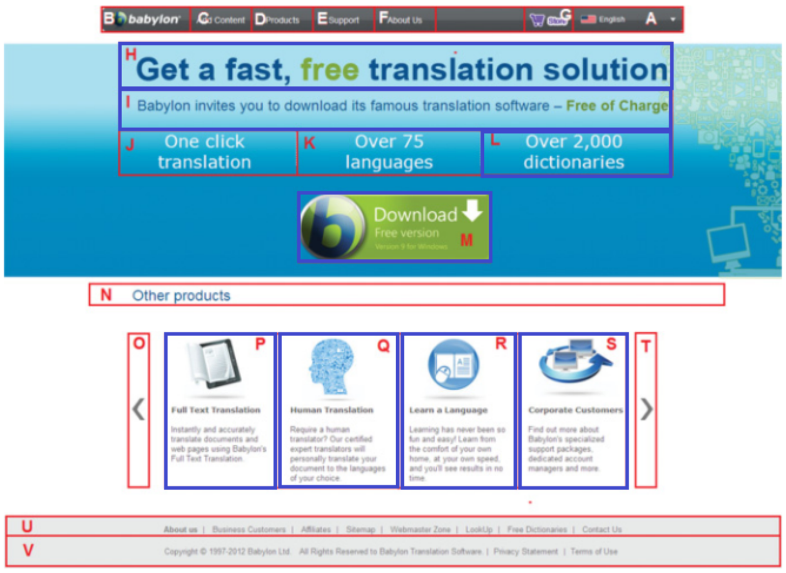
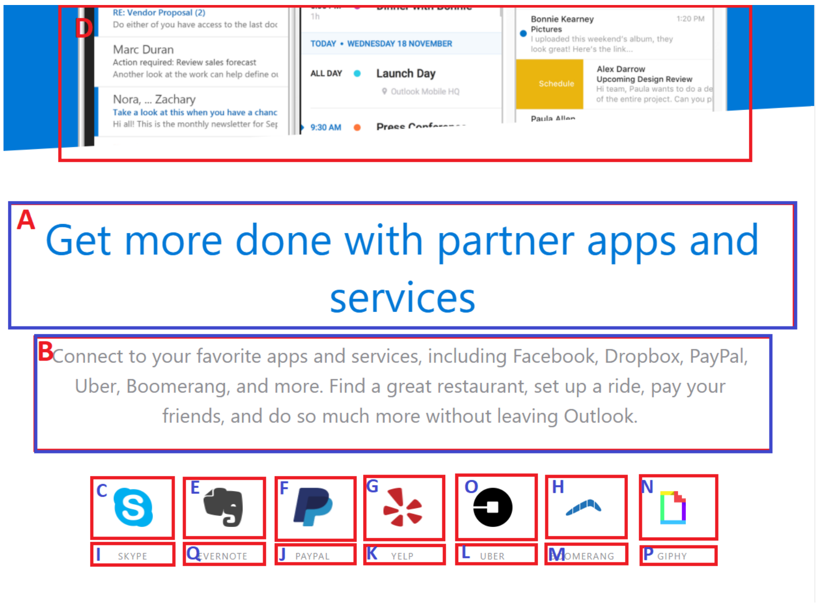

B babylon
C Content
D Products
E Support
F About Us
G
English
A
H Get a fast, free translation solution
I Babylon invites you to download its famous translation software – Free of Charge
J One click translation
K Over 75 languages
L Over 2,000 dictionaries
Download Free version
M
N Other products
O
P Full Text Translation
Q Human Translation
R Learn a Language
S Corporate Customers
T
U About us | Business Customers | Affiliates | Sitemap | Webmaster Zone | LookUp | Free Dictionaries | Contact Us
V Copyright © 1997-2012 Babylon Ltd All Rights Reserved to Babylon Translation Software | Privacy Statement | Terms of Use

Figure 3.7 Babylon web page [10]

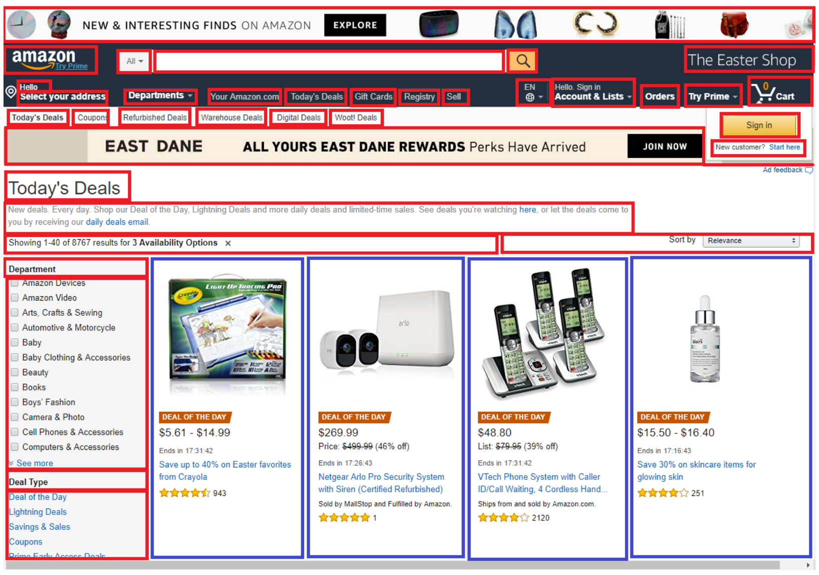
The color-related features which were included in the dataset are: Euclidean distance


D
RE: Vendor Proposal (2)
Do either of you have access to the last doc
Marc Duran
Action required: Review sales forecast
Another look at the work can help define ou
Nora, ... Zachary
Take a look at this when you have a chanc
Hi all! This is the monthly newsletter for Sep
TODAY • WEDNESDAY 18 NOVEMBER
ALL DAY
Launch Day
Outlook Mobile HQ
9:30 AM
Bonnie Kearney
1:20 PM
Pictures
I uploaded this weekend's album, they
look great! Here's the link...
Schedule
Alex Darrow
Upcoming Design Review
Hi team, Paula wants to do a de
of the entire project. Can you p
A
B
C
E
F
G
O
H
N
I
Q
J
K
L
M
P
SKYPE
EVERNOTE
PAYPAL
YELP
UBER
BOOMERANG
GIPHY

Get more done with partner apps and services

Connect to your favorite apps and services, including Facebook, Dropbox, PayPal, Uber, Boomerang, and more. Find a great restaurant, set up a ride, pay your friends, and do so much more without leaving Outlook.

Figure 3.8 Outlook web page [13]



Today's Deals

New deals. Every day. Shop our Deal of the Day, Lightning Deals and more daily deals and limited-time sales. See deals you're watching here, or let the deals come to you by receiving our daily deals email.

Showing 1-40 of 8767 results for 3 Availability Options ×

$5.61 - $14.99
Ends in 17:31:42
Save up to 40% on Easter favorites from Crayola
943

$269.99
Price: ~~$499.99~~ (46% off)
Ends in 17:26:43
Netgear Arlo Pro Security System with Siren (Certified Refurbished)
Sold by MallStop and Fulfilled by Amazon.
1

$48.80
List: ~~$79.95~~ (39% off)
Ends in 17:31:42
VTech Phone System with Caller ID/Call Waiting, 4 Cordless Hand...
Ships from and sold by Amazon.com.
2120

$15.50 - $16.40
Ends in 17:16:43
Save 30% on skincare items for glowing skin
251

Figure 3.9 Amazon web page [13]

Table 3.2 List of the different font sizes and weights

Font Size	**Font weights**
XX-small<10px	100=<Thin<200
X-small<13px	200=<Light<300
Small<16px	300=<Book<400
Medium<18px	400=<Normal<500
Large<24px	500=<Medium<600
X-large<32px	600=<Semi-bold<700
XX-large>=32px	700=<Bold<800
	800=<Heavy<900
	Ultra-black>=900

of color, standard deviation of color, and binary feature for the background color (see Table 3.1). It was noticed that the segments with different text color ranges were likely to be trending. Consider for example, the screenshot of the trending elements highlighted in blue in Figure 3.9 of the Amazon web page within the segment, we observe that there are four different color variations. Another example can be seen from the AVG web page in Figure 3.5 the segment 'G' which has three color variations, and we also notice that the segment 'I' has two different colors. Similar trends were observed among other web pages. Based on these analyses, Euclidean distance was computed for all the text colors in all segment of web pages. To compute the Euclidean distance, we used the RGB values of all the text colors within the segment. For example, if a segment has three different color variation, a 3×3 matrix is constructed by computing the Euclidean distance between the colors, and then highest value is selected. Let us take the AVG web page segment 'G' as an example. This segment contains three RGB values $rgb(0, 0, 0)$, $rgb(92, 111, 123)$, and $rgb(255, 255, 255)$. The matrix is constructed as follows:

$$\begin{bmatrix} 0 & 153.99 & 441.67 \\ 153.99 & 0 & 254.42 \\ 441.67 & 254.41 & 0 \end{bmatrix}$$

Based on the matrix, the value 441.67 is selected as the Euclidean distance feature of color. The Standard deviation was calculated for the different text colors by using Euclidean distance color matrix. We created a binary feature for the background color by determining whether the segment contains a background color with respect to the web page or with respect to the segment. Namely, by looking at the screenshot of the trending elements highlighted in blue for the Amazon web page in Figure 3.9, the AVG web page in Figure 3.5, and the Babylon web page in Figure 3.7, to name a few, it was noticed that the trending elements were more likely to have a different background color.

Target variable

The dataset-1 contained 112 elements in total for browsing and searching tasks. The browsing task has 27 trending elements as it can be seen from Table 3.3 which represents 24% of the dataset. Based on Table 3.3 for the browsing task it can be seen that the AVG web page has the lowest percentage of trending elements, whereas the Apple web page has the highest number of trending elements. The searching task has 31 trending elements as seen in Table 3.4 which represents 27.7% of the elements. Based on Table 3.3 and 3.4 it can be noticed that the number of trending elements is less than the number of non-trending which means that the dataset is imbalanced.

Table 3.3 Summary of Dataset-1 browsing task

Page	**Trending**	**Not Trending**	**Number of Segments**	**% Trending Elements**
Apple	7	11	18	38.89%
AVG	2	23	25	8%
Babylon	8	14	22	36.36%
BBC	4	17	21	19%
GoDaddy	3	13	16	18.75%
Yahoo	3	7	10	30%
Total	27	85	112	24.1%

The dataset-2 contained 206 elements, and 38 of these elements are trending for both browsing and synthesis tasks (see Table 3.5 and Table 3.6). Based on the values in Table 3.5 we can notice that the Netflix and YouTube web pages have the highest number of trending elements compared to the other web pages for browsing tasks. Whereas for synthesis, we see from Table 3.6 that the Outlook and WhatsApp web pages have the highest number of trending elements. For both tasks the BBC web

Table 3.4 Summary of Dataset-1 searching task

Page	**Trending**	**Not Trending**	**Number of Segments**	**% Trending Elements**
Apple	7	11	18	38.89%
AVG	2	23	25	8%
Babylon	8	14	22	36.36%
BBC	6	15	21	28.57%
GoDaddy	5	11	16	31.25%
Yahoo	3	7	10	30%
Total	31	81	112	27.7%

page has the least number of trending elements (see Table 3.5 and Table 3.6).

Table 3.5 Summary of Dataset-2 browsing task

Page	**Trending**	**Not Trending**	**Number of Segments**	**% Trending Elements**
Adobe	7	3	10	70%
Amazon	4	37	41	9%
BBC	2	37	39	5.13%
Netflix	8	6	14	57.14%
Outlook	2	15	17	11.76%
Whatsapp	4	8	12	33.33%
WordPress	3	14	17	17.64%
Youtube	8	48	56	14.28%
Total	38	168	206	18.44%

Table 3.6 Summary of Dataset-2 synthesis task

Page	**Trending**	**Not Trending**	**Number of Segments**	**% Trending Elements**
Adobe	3	7	10	30%
Amazon	3	38	41	7%
BBC	2	37	39	5
Netflix	6	8	14	42.85%
Outlook	9	8	17	52.94%
Whatsapp	7	5	12	58.33%
WordPress	3	14	17	17.65%
Youtube	5	51	56	8.93%
Total	38	168	206	18.44%

We formulated the problem of identifying whether a visual element is trending or not as a binary classification problem. Specifically, the trending elements were encoded as one, and non-trending elements were encoded as zero. A data format encoding was also used for web pages URL. The URL of web pages as a feature is converted to numeric values for processing. For example, www.apple.com was labeled with the number 0, and www.avg.com was labeled as 1..., etc.

3.2.2 Modeling

Since we had a binary classification problem where we tried to identify which element was trending, we tested both datasets with various classification algorithms, including Support Vector Machine (SVM), K-nearest neighbors algorithm (KNN), Logistic Regression, Random Forest, Decision Tree, and Naive Bayes. We also inspected the effects of using different sampling techniques, including oversampling and Synthetic Minority Oversampling Technique (SMOTE). We have used the leave-one-out method to make our page-specific prediction.

3.2.2.1 Features Selection

The initial experiment investigated which features most affect the trending elements, and this was done by computing information gain and correlation matrix.

Information gain computes the reduction in entropy from the transformation of a dataset [25]. Table 3.7 show the results of the top five features with the highest information gain for each web page for dataset-1. It can be seen from the results that each web page has at least a positional feature and a size-related feature. Table 3.7 shows that the Div count feature is ranked first for four out six web pages in the browsing task. When observing the information gain results of Table 3.7 for the searching task, we notice that the coordinates and the size of the elements are the most common features between web pages. Overall, there is no specific trend for feature selection; they vary depending on the web page.

In addition, Table 3.8 show the results of the top five features with the highest information gain for each web page for dataset-2. When observing the results of Table 3.8 for the browsing task, the Top Coordinate feature has appeared as the first feature for four out of eight web pages, and these web pages are Adobe, Netflix, Outlook, WhatsApp. For the synthesis task from Table 3.8 Top Coordinate feature appeared as the first feature in two web pages, which are Adobe and Netflix, while the Word Count feature appeared as the first feature in WordPress and YouTube web pages. Similarly to the observation from dataset-1, in dataset-2 there is no specific trend for the features selection.

Table 3.7 Information Gain Dataset-1

Task	**Page**	**1**	**2**	**3**	**4**	**5**
Browsing	Apple	TopCenter(0.96)	Right(0.81)	Left(0.78)	Size(0.67)	AspectRatio(0.673)
	AVG	Div Count(0.40)	Top(0.40)	Bottom(0.40)	BottomCenter(0.40)	Size(0.40)
	Babylon	Right(0.95)	Size(0.95)	AspectRatio(0.95)	RelativeSize (0.95)	Left(0.82)
	BBC	Div Count(0.70)	Bottom(0.70)	BottomCenter(0.70)	Size(0.70)	AspectRatio(0.70)
	GoDaddy	Div Count(0.70)	Li count(0.70)	Li(0.70)	Top(0.70)	Bottom(0.70)
	Yahoo	Div Count(0.88)	Li count(0.88)	A count(0.88)	Top(0.88)	Bottom(0.88)
Searching	Apple	TopCenter(0.96)	Right(0.81)	Left(0.78)	Size(0.67)	AspectRatio(0.67)
	AVG	Div Count(0.40)	Top(0.40)	Bottom(0.40)	BottomCenter(0.40)	Size(0.402)
	Babylon	Size(0.95)	AspectRatio(0.95)	RelativeSize (0.95)	Right(0.82)	Left(0.82)
	BBC	Bottom(0.86)	BottomCenter(0.86)	Size(0.86)	AspectRatio(0.86)	RelativeSize (0.86)
	GoDaddy	Bottom(0.90)	TopCenter(0.90)	BottomCenter(0.90)	Size(0.90)	AspectRatio(0.90)
	Yahoo	Div Count(0.88)	Li count(0.88)	A count(0.88)	Top(0.88)	Bottom(0.88)
Abbreviations: BR $\Rightarrow BottomRight, BC \Rightarrow BottomCenter, BL \Rightarrow BottomLeft, TR \Rightarrow TopRight, TC \Rightarrow TopCenter$ CL$\Rightarrow CenterLeft, CR \Rightarrow CenterRight, CC \Rightarrow CenterCenter$						

Table 3.8 Information Gain Dataset-2

Task	**Page**	**1**	**2**	**3**	**4**	**5**
Browsing	Adobe	Top(0.61)	Bottom(0.61)	BottomCenter(0.61)	Size(0.61)	AspectRatio(0.61)
	Amazon	Div Count(0.46)	Span count(0.46)	A count(0.46)	Bottom(0.46)	Right(0.46)
	BBC	Div Count(0.29)	Li count(0.29)	Span count(0.29)	A count(0.29)	h3 count(0.29)
	Netflix	Top(0.84)	Bottom(0.84)	BottomCenter(0.84)	Right(0.44)	Left(0.44)
	Outlook	Top(0.52)	Bottom(0.52)	Right(0.52)	Left(0.52)	TopCenter(0.52)
	WhatsApp	Top(0.92)	Bottom(0.92)	BottomCenter(0.92)	Size(0.92)	AspectRatio(0.92)
	WordPress	Span count(0.67)	Top(0.67)	Right(0.67)	Left(0.67)	TopCenter(0.67)
	YouTube	Bottom(0.51)	WordCount(0.48)	Top(0.47)	BottomCenter(0.47)	Right(0.34)
Synthesis	Adobe	Top(0.61)	Bottom(0.61)	Right(0.61)	Left(0.61)	TopCenter(0.61)
	Amazon	Right(0.38)	Left(0.38)	TopCenter(0.38)	Size(0.28)	AspectRatio(0.28)
	BBC	Div Count(0.29)	Size(0.29)	AspectRatio(0.29)	RelativeSize(0.29)	WordCount(0.29)
	Netflix	Top(0.84)	Bottom(0.84)	BottomCenter(0.84)	Right(0.44)	Left(0.44)
	Outlook	Right(0.53)	Left(0.53)	TopCenter(0.53)	Top(0.40)	Bottom(0.40)
	WhatsApp	Size(0.98)	AspectRatio(0.98)	RelativeSize(0.98)	WordCount(0.98)	Top(0.75)
	WordPress	WordCount(0.67)	EcludianDistance(0.67)	Std(0.67)	Top(0.51)	Right(0.51)
	YouTube	WordCount(0.31)	Top(0.28)	Bottom(0.28)	BottomCenter(0.28)	TopCenter(0.17)
Abbreviations: BR $\Rightarrow BottomRight, BC \Rightarrow BottomCenter, BL \Rightarrow BottomLeft, TR \Rightarrow TopRight, TC \Rightarrow TopCenter$ CL$\Rightarrow CenterLeft, CR \Rightarrow CenterRight, CC \Rightarrow CenterCenter$						

Correlation is a statistical technique that determines how one variable changes in relation to the other variables [4]. In this thesis, observations were made regarding which features are positively correlated with our target variable. We have selected the top five correlated features for each web page. Table 3.9 and Table 3.10 show the top five correlated features for each dataset. Based on [34] it is mentioned that if the correlation coefficient threshold is less than 0.3, it indicates a weak relationship, and if it's between 0.3 and 0.7, then it has a moderate positive linear relationship. Finally, if the the correlation coefficient threshold is between 0.7 and 1.0, then it indicates a strong positive linear relationship with the target variable. From Table 3.9 we can notice that size and text-related features seem to be appearing most for the different web pages. Whereas, for dataset-2, we can see from Table 3.10 position and text-related features are occurring most for the different web pages.

Table 3.9 Correlation scores Dataset-1

Task	Page	1	2	3	4	5
Browsing	Apple	Img_count(0.66)	Image(0.66)	Size(0.50)	RelativeSize (0.50)	AspectRatio(0.50)
	AVG	Size(0.96)	RelativeSize (0.96)	Div Count(0.87)	CC segment(0.80)	WordCount(0.80)
	Babylon	600=<semi-bold<700(0.72)	RelativeSize (0.60)	Size(0.60)	WordCount(0.54)	Span count(0.53)
	BBC	RelativeSize (0.89)	Size(0.89)	Large<24(0.87)	Div Count(0.78)	Img_count(0.75)
	GoDaddy	Li(1.00)	X-large<32(1.00)	WordCount(0.94)	Li count(0.8704)	Size(0.82)
	Yahoo	Small<16(1.00)	EcludianDistance(0.82)	H2(0.80)	Ul count(0.77)	Iframe(0.76)
Searching	Apple	Img_count(0.66)	Image(0.66)	Size(0.5035)	RelativeSize (0.50)	AspectRatio(0.49)
	AVG	Size(0.95)	RelativeSize (0.96)	Div Count(0.87)	CC segment(0.80)	WordCount(0.79)
	Babylon	600=<Semi-bold<700(0.72)	RelativeSize (0.69)	Size(0.69)	WordCount(0.53)	H2 count(0.42)
	BBC	400=<Normal<500(0.8944)	Div Count(0.88)	RelativeSize (0.88)	Size(0.88)	WordCount(0.85)
	GoDaddy	WordCount(0.72)	Li(0.71)	X-large<32(0.71)	Std(0.66)	EcludianDistance(0.66)
	Yahoo	Small<16(1.00)	EcludianDistance(0.82)	H2(0.80)	Ul count(0.77)	Iframe(0.76)

Abbreviations: BR $\Rightarrow BottomRight, BC \Rightarrow BottomCenter, BL \Rightarrow BottomLeft, TR \Rightarrow TopRight, TC \Rightarrow TopCenter$
CL$\Rightarrow CenterLeft, CR \Rightarrow CenterRight, CC \Rightarrow CenterCenter$

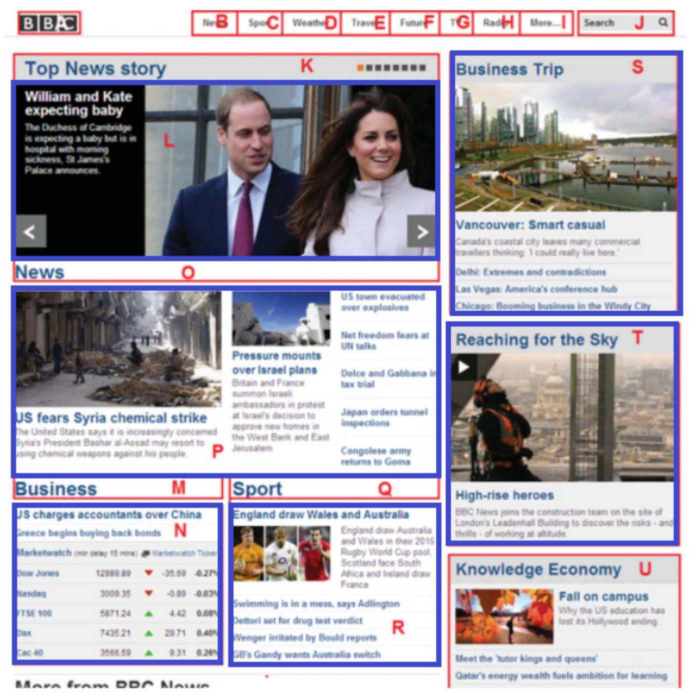


Figure 3.10 BBC web page [10]

In this work, the information gain and correlation matrix were calculated to observe which features had the most effect on predicting the trending elements. The results scores have varied due to the variation of the web pages. For example, the Apple (see Figure 3.1) web page was image dominated so the feature set showed that the image had a high correlation with the trending element. Also, some web pages like the BBC web page (See Figure 3.10) had mixture of images and text but the images

Table 3.10 Correlation scores Dataset-2

Task	Page	1	2	3	4	5
Browsing	Adobe	BR segment(0.53)	WordCount(0.51)	Div Count(0.43)	Div(0.43)	H3 count(0.43)
	Amazon	Medium<18(1.00)	Segmentbackground(1.00)	EcludianDistance(0.95)	Div Count(0.95)	Std(0.90)
	BBC	X-large<32(1.00)	I count(0.97)	Span count(0.96)	Div Count(0.96)	Ul count(0.95)
	Netflix	WordCount(0.48)	400=<Normal<500(0.41)	BC segment(0.29)	H2 count(0.24)	H2(0.24)
	Outlook	WordCount(0.83)	RelativeSize (0.74)	Size(0.74)	Span count(0.68)	Span(0.68)
	WhatsApp	Div(0.50)	Medium<18(0.43)	400=<Normal<500(0.41)	BC segment(0.41)	AspectRatio(0.37)
	WordPress	Size(0.93)	RelativeSize (0.93)	Span count(0.87)	Div Count(0.83)	Div(0.83)
	YouTube	Std(0.67)	EcludianDistance(0.67)	Div Count(0.66)	RelativeSize (0.65)	Size(0.65)
Synthesis	Adobe	H1 count(0.51)	H1(0.51)	TC segment(0.51)	XX-large>=32(0.51)	Medium<600(0.51)
	Amazon	Button count(0.56)	Button(0.56)	Medium<18(0.54)	Segmentbackground(0.54)	EcludianDistance(0.51)
	BBC	Img_count(0.76)	RelativeSize (0.76)	Size(0.76)	WordCount(0.74)	BC segment(0.70)
	Netflix	Button count(0.32)	Button(0.32)	Small<16(0.32)	Left(0.2239)	TopCenter(0.14)
	Outlook	RelativeSize (0.462)	Size(0.46)	Img_count(0.42)	Image(0.42)	AspectRatio(0.25)
	WhatsApp	Li count(0.66)	Li(0.66)	A(0.66)	Ul count(0.66)	Ul(0.66)
	WordPress	WordCount(0.91)	Div Count(0.83)	Div(0.83)	Img_count(0.83)	Image(0.83)
	YouTube	CR segment(0.61)	I count(0.40)	A count(0.39)	Std(0.38)	EcludianDistance(0.38)

Abbreviations: BR $\Rightarrow BottomRight, BC \Rightarrow BottomCenter, BL \Rightarrow BottomLeft, TR \Rightarrow TopRight, TC \Rightarrow TopCenter$
CL$\Rightarrow CenterLeft, CR \Rightarrow CenterRight, CC \Rightarrow CenterCenter$

were covering most of the web page. Other web pages were text dominated so the word count was appearing as a high correlated feature.

This made the task of applying feature selection to be difficult, and thus page categorization might need to be applied, where web pages can be divided into mostly text, mostly images, and mixture of image and text as in [35]. Furthermore, based on previous work, it was found that size and element position on a web page is very informative in attention prediction [6, 15, 17, 36, 41]. By observing the results Tables 3.9 and 3.10, we observe that size related features and positional features seem to be appearing most for the different pages. This means that these features are important for identifying the trending elements.

Overall, it was noticed that each web page has a different feature set. So, given a web page, we would not be able to decide which features should be included. Therefore, in this thesis, it was decided not to continue with feature selection and all features were used.

3.2.2.2 Model Development and Evaluation

At this stage the classification models were constructed and trained by using the leave-one-out approach. In the case of dataset-1 and dataset-2, the Recall, Precision, Accuracy and F1 score metrics of each web page for every model were computed. In order to describe these metrics, we first need to define the following terms:

- True Positive (TP): the number of times that a trending element is classified correctly.
- True Negative (TN): the number of times that a non-trending element is classified correctly.
- False Positive (FP): the number of times that a trending element is misclassified.
- False Negative (FN): the number of times that a non-trending element is misclassified

Given TP, TN, FP and FN, we computed the following metrics: Recall, Precision, Accuracy, and F1 score.

- Recall: the percentage of correctly predicted positive observations to the total number of observations.

$$Recall = \frac{TP}{TP + FN}$$

- Precision: the percentage of correctly predicted positive observations to the total number of predicted positive observations.

$$Precision = \frac{TP}{TP + FP}$$

- Accuracy: the performance of a model across all classes. Calculated as the ratio between the number of correct predictions to the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- F1 score: the balance between Precision and Recall.

$$F1 \;\; score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Table 3.11 shows the most popular classifications models that were used in this work:

In order to determine which classification model is better for us, we computed the metrics of all the web pages in dataset-1 and dataset-2 by using leave-one-out approach, then we computed the averages of each metric for each classification model. For example, in dataset-1 we have six different web pages we computed the metrics using the leave-one-out approach for each web page, and then we computed the average. This was carried out for all the classification model.

Table 3.11 Classification Models (see Table A.2 for configurations)

Model	**Descrption**	**Python Library**	**Parameters**
SVM Linear	separates data into two classes using one straight line to predict the label of the sample.	sklearn [32]	Default
Logistic Regression	it is used to predict a binary outcome based on a set of independent variables.	sklearn [32]	Default
Decision Tree	it is a tree-like model where features are split through a cost function.	sklearn [32]	Default
Random Forest	combines the output of multiple decision trees to reach a single result tree to obtain a better predictive model.	sklearn [32]	Default
KNN	is an estimation method used to classify data points based on what group the closest data points are in determines how likely a data point is to be in.	sklearn [32]	Different K values, and distance metric
Naive Bayes	it is an algorithms that apply Bayes' theorem based on the assumption that the class variable is independent of the pair of features.	sklearn [32]	Default

For the SVM Linear, Logistic Regression, Random Forest, and Decision Tree, with each iteration, the prediction is different due to randomization, and thus the need to compute the average of 100 iterations.

kNN has many different variations due to its parameter settings thus the use of the Elbow method, to determine the value of the number k in the kNN model. This method finds the optimal k value by computing the error rate.
The study uses the leave-one-out approach, and therefore the k value for each web page of the two datasets had to be computed. In order to obtain the best possible outcome, we used the most common distance metric method which are: Euclidian, Manhattan, Chebyshev, and Minkowski distance metrics.

After conducting different experiments and determining which model is the best for dataset-1 and dataset-2. The baseline F1-score was computed using the Random prediction algorithm to compare the results. The Random prediction algorithm does not use complex techniques to make the predictions; it uses a random one that represents the worst case of prediction. Which makes it a benchmark for measuring how other algorithms are better [31]. In addition, further analysis was done by applying sampling techniques.

Since the dataset was imbalanced, which is expected due to the nature of the data, the number of trending elements tends to be lower than the number of non-trending elements. Due to this, oversampling techniques have been tested on both datasets. Specifically, oversampling creates training examples from the minority class (which is treading element in our case) by randomly duplicating examples [28]. Synthetic Minority Oversampling Technique (SMOTE) is where a synthetic instance is randomly created in feature space based on the nearest neighbor of the minority class instance [7].

In this study after we obtained the best model, we apply sampling techniques (oversampling or SMOTE) to the datasets. To preform sampling, we first split the dataset using the leave-one-out approach and then we apply a sampling method (oversample or SMOTE) to the training set only. Then we use this training set to build our model and obtain the prediction on the testing set.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter focuses on presenting the results obtained by applying the methodology explained in Chapter 3. In this chapter, we compare the classification models, which are SVM, KNN, Logistic Regression, Random Forest, Decision Tree, and Naive Bayes, where we compare the results of the different classification models and determine which model is the best for our datasets. Then present a baseline comparison, where we compare our best model to the random prediction model. After that, we present a sampling comparison, where we apply oversampling techniques to handle our imbalanced data and compare the results with the original data. Finally, we discuss our results regarding the performance of the model.

4.1 Comparison of Classification Models

When observing the results of dataset-1, for the browsing task, in Table 4.1, it was noticed that the KNN model showed the highest average F1 score of 0.91 compared with the other models. The F1 score ranges for the other models are in $[0.53, 0.77]$. The Bernoulli NB had the lowest average F1 score of 0.53, and Complement NB and Multinomial NB had a good average of F1 score 0.77. In addition, when observing the results in Table 4.2 for the searching task, the KNN model also outperformed the other models; it had an average F1 score of 0.88, while the other models' scores ranged in $[0.52, 0.83]$. Similar results were observed with dataset-2. For the browsing task, as it can be seen in Table 4.3, the KNN model scored a high average F1 score with 0.82. The other models had a prediction score in the rang of $[0.44, 0.82]$, with the Bernoulli NB model reporting the lowest average F1 score of 0.08. Furthermore, by observing the results of the synthesis task, all the modes did not perform very well, but KNN model scored the highest with Multinomial NB and Complement NB, with an average F1 score of 0.44. After all these observations, it was decided to focus on using the KNN model in our prediction experiments. For further details regarding the results of each model (see appendix B).

Table 4.1 Results of different classifiers for browsing task in dataset-1

Classification Model	**Recall**	**Precision**	**Accuracy**	**F1**
SVM Linear	0.6665	0.6337	0.8283	0.5937
Gaussian NB	0.6905	0.5695	0.7763	0.5391
Multinomial NB	0.8452	0.7693	0.8462	0.7710
Complement NB	0.8452	0.7693	0.8462	0.7710
Bernoulli NB	0.6667	0.4611	0.8028	0.5361
Random forest	0.6505	0.6261	0.8124	0.5663
Decision Tree	0.6661	0.7569	0.8354	0.6563
Logistic Regression	0.7351	0.7786	0.8360	0.6870
KNN	**0.9315**	**0.8935**	**0.9328**	**0.9083**

Table 4.2 Results of different classifiers for searching task in dataset-1

Classification Model	**Recall**	**Precision**	**Accuracy**	**F1**
SVM Linear	0.5846	0.6609	0.8084	0.5589
Gaussian NB	0.6238	0.6171	0.7713	0.5400
Multinomial NB	0.8441	0.8361	0.8732	0.8139
Complement NB	0.8441	0.8361	0.8732	0.8139
Bernoulli NB	0.6417	0.5921	0.7784	0.5241
Random forest	0.7476	0.7633	0.8354	0.6870
Decision Tree	0.8228	0.7157	0.8066	0.7052
Logistic Regression	0.8173	0.8546	0.8702	0.7968
KNN	**0.8857**	**0.9160**	**0.9285**	**0.8841**

Table 4.3 Results of different classifiers for browsing task in dataset-2

Classification Model	**Recall**	**Precision**	**Accuracy**	**F1**
SVM Linear	0.6505	0.5303	0.7602	0.5262
Gaussian NB	0.6094	0.4375	0.7428	0.4833
Multinomial NB	0.8438	0.5643	0.7499	0.6282
Complement NB	0.8438	0.5643	0.7499	0.6282
Bernoulli NB	0.5357	0.3708	0.7415	0.4432
Random forest	0.6369	0.5790	0.7747	0.5943
Decision Tree	0.6369	0.5790	0.7747	0.5943
Logistic Regression	0.6741	0.5014	0.7141	0.4709
KNN	**0.8661**	**0.8292**	**0.8518**	**0.8243**

Table 4.4 Results of different classifiers for synthesis task in dataset-2

Classification Model	**Recall**	**Precision**	**Accuracy**	**F1**
SVM Linear	0.5227	0.3775	0.6811	0.3752
Gaussian NB	0.4643	0.4167	0.7649	0.3975
Multinomial NB	0.6006	0.4732	0.6579	0.4493
Complement NB	0.6006	0.4732	0.6579	0.4493
Bernoulli NB	0.4333	0.2169	0.6572	0.2821
Random forest	0.1059	0.0749	0.6764	0.0848
Decision Tree	0.3311	0.2717	0.6292	0.2303
Logistic Regression	0.5060	0.3747	0.7205	0.3780
KNN	**0.3457**	**0.6354**	**0.7907**	**0.4409**

4.2 Baseline Comparison

After determining the wining model in the previous section 4.1, we further analyze the KNN model to test the efficiency, and we compared the results using the baseline comparison.

4.2.1 Baseline Comparison Dataset-1

Table 4.5 and Table 4.6 show the results of Recall, Precision, Accuracy, and F1 score using the KNN model for original dataset-1. In addition, Table 4.5 and Table 4.6 also show the baseline F1 scores using Random prediction algorithm for each browsing and searching task of dataset-1. As can be seen from the results of the browsing task in Table 4.5, the average F1 score for the browsing task is 0.91, while the baseline is 0.33. In addition, it can be noticed that in Table 4.6, the average F1 score of the searching task is 0.88, whereas the baseline is 0.34.

When observing the F1 score for browsing task in Table 4.5, we can also see that the model can identify trending elements with a high F1 score for most of the web pages, the highest being the AVG, BBC, and GoDaddy web pages with a F1 score of 1. The lowest F1 score 0.77 was recorded for the Apple web page in the browsing task. From Table 4.6 the results of the highest F1 score for the searching task was recorded in the AVG web page with F1 score of 1 and the lowest was observed in the GoDaddy web page with a F1 score of 0.75. In addition, when observing the results of the baseline for both the browsing and searching task (see Table 4.5 and Table 4.6), we can see that the KKN model outperformed the the random prediction algorithm for all the web pages.

Table 4.5 Results of original browsing task in dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**	**Baseline F1**
Apple	0.7143	0.8333	0.8333	0.7692	0.5882
AVG	1.0000	1.0000	1.0000	1.0000	0.1429
Babylon	0.8750	0.7778	0.8636	0.8235	0.4211
BBC	1.0000	1.0000	1.0000	1.0000	0.2667
GoDaddy	1.0000	1.0000	1.0000	1.0000	0.3333
Yahoo	1.0000	0.7500	0.9000	0.8571	0.2500
Average	0.9316	0.8935	0.9328	0.9083	0.3337

Table 4.6 Results of original searching task in dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**	**Baseline F1**
Apple	0.7143	1.0000	0.8889	0.8333	0.5882
AVG	1.0000	1.0000	1.0000	1.0000	0.1429
Babylon	1.0000	0.8889	0.9545	0.9412	0.5263
BBC	1.0000	0.8571	0.9524	0.9231	0.3529
GoDaddy	0.6000	1.0000	0.8750	0.7500	0.4286
Yahoo	1.0000	0.7500	0.9000	0.8571	0.2500
Average	0.8857	0.9160	0.9285	0.8841	0.3815

4.2.2 Baseline Comparison Dataset-2

Table 4.7 and Table 4.8 show the results of Recall, Precision, Accuracy, and F1 score using the KNN model for the original dataset-2. Further, These tables also show the baseline F1 scores for each browsing and synthesis task for dataset-2. From Table 4.7, the results show that the average F1 score is 0.82 while the baseline is 0.30, we can observe that there is a considerable difference. By following page specific results, the scores demonstrates that Amazon, BBC, and WordPress web pages yield a result of 1.0 F1 score measure, which means the KNN model successfully predicted all the elements correctly for the browsing task as shown in Table 4.7. The results in Table 4.8 show that KNN model cannot make a proper prediction for some web pages for the synthesis task, and that could be due to the complexity of the synthesis task. By looking at the average F1 score of the KNN model for the synthesis task, it demonstrates a higher F1 score than the baseline, where the KNN scored 0.44 while the baseline 0.29 (see Table 4.8). However, by observing the results of a specific web page, the baseline of the Adobe web page is higher than the KNN F1 score. The KNN F1 score for the Adobe web page is 0, and the baseline F1 score is 0.25.

Table 4.7 Results of original browsing task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**	**Baseline F1**
Adobe	0.4285	1.0000‘	0.6000	0.6000	0.5000
Amazon	1.0000	1.0000	1.0000	1.0000	0.1739
BBC	1.0000	1.0000	1.0000	1.0000	0.0952
Netflix	0.7500	0.6666	0.6428	0.7058	0.3750
Outlook	1.0000	0.6666	0.9411	0.8000	0.1667
Whatsapp	0.7500	0.5000	0.6666	0.6000	0.6000
WordPress	1.0000	1.0000	1.0000	1.0000	0.3077
YouTube	1.0000	0.8000	0.9642	0.8888	0.2162
Average	0.8661	0.8292	0.8518	0.8243	0.3043

Table 4.8 Results of original synthesis task in dataset-2

Page	Recall	Precision	Accuracy	F1	Baseline F1
Adobe	0	0	0.7000	0	0.2500
Amazon	0	0	0.9268	0	0.0909
BBC	0.5000	1.0000	0.9744	0.6667	0.1905
Netflix	0.5000	0.7500	0.7143	0.6000	0.4286
Outlook	0.5556	0.8333	0.7059	0.6667	0.5263
Whatsapp	0.1429	0.5000	0.4167	0.2222	0.4615
WordPress	0.6667	1.0000	0.9412	0.8000	0.1538
YouTube	0.4000	1.0000	0.9464	0.5714	0.2343
Average	0.3457	0.6354	0.7907	0.4409	0.2920

4.3 Sampling Comparison

After observing the results of the original dataset-1 and dataset-2, more analysis was conducted on the dataset to try and improve the prediction results. Since the dataset is imbalanced, sampling techniques were tested: oversample and SMOTE.

4.3.1 Sampling Comparison Dataset-1

Table 4.9 and Table 4.10 show the F1 score for original, oversampled, and SMOTE dataset for both browsing and searching tasks using the KNN model. Based on the F1 scores from tables, there is no specific trend on which approach is better. The average F1 score for the original data performed better than the sampled data. The original dataset had an average F1 score of 0.91 while the oversampled and the SMOTE had an average F1 score of 0.86 and 0.88, respectively. However, sometimes the oversampled data had a slight change on particular web pages. For example, in the Apple web page, the oversampled and SOMTE approach a had slightly higher F1 score than the original data in both browsing and searching task. In the browsing task, the oversampled and SMOTE F1 score of the Apple web page is 0.86, and in the searching task, oversampled and SMOTE F1 score for the Apple web page is 0.86. For further details regarding the results of KNN model with the sampling techniques (see appendix B).

Table 4.9 Summary of F1 scores for browsing task in dataset-1

Page	Original F1	Oversample F1	SMOTE F1	Baseline F1
Apple	0.7692	0.8571	0.8571	0.5882
AVG	1.0000	1.0000	1. 0000	0.1429
Babylon	0.8235	0.8235	0.8235	0.4211
BBC	1.0000	0.7273	0.7273	0.2667
GoDaddy	1.0000	1.0000	1. 0000	0.3333
Yahoo	0.8571	0.7500	0.8571	0.2500
Average	0.9083	0.8597	0.8775	0.3337

Table 4.10 Summary of F1 scores for searching task in dataset-1

Page	Original F1	Oversample F1	SMOTE F1	Baseline F1
Apple	0.8333	0.8571	0.8571	0.5882
AVG	1.0000	1.0000	1.0000	0.1429
Babylon	0.9412	0.8889	0.8889	0.5263
BBC	0.9231	0.9231	0.9231	0.3529
GoDaddy	0.7500	0.7500	0.7500	0.4286
Yahoo	0.8571	0.7500	0.7500	0.2500
Average	0.8841	0.8615	0.8615	0.3815

4.3.2 Sampling Comparison Dataset-2

Table 4.11 and Table 4.12 show the F1 score for original, oversampled, and SMOTE dataset using the KNN model, for both browsing and synthesis tasks. In Table 4.11 the oversampled and SMOTE improved the F1 score for the Adobe web page. The Adobe web page had an F1 score of 0.60 for the original dataset, while oversampling and SMOTE had the same F1 score of 0.86. There was also a slight increase in the F1 score for the Netflix web page when using oversampling techniques. For the Netflix web page, the F1 score increased from 0.71 to 0.73 and to 0.78 by using oversampled and SMOTE techniques, respectively. For WordPress and Amazon web pages, the F1 score dropped with oversampling. The Amazon F1 score for original data was 1, and then with oversampling and SMOTE, the F1 score became 0.73 and 0.67, respectively.

The original data F1 score was the highest by observing the average scores. Also, all the models provided a better F1 score compared with baseline. When observing the results of Table 4.12 the oversampling techniques showed a higher F1 score compared to the original F1 and baseline. For WordPress and Youtube, the original F1 score is

Table 4.11 Summary of F1 scores for browsing task in dataset-2

Page	Original F1	Oversample F1	SMOTE F1	Baseline F1
Adobe	0.6000	0.8571	0.8571	0.5000
Amazon	1.0000	0.7273	0.6667	0.1739
BBC	1.0000	1.0000	0.8000	0.0952
Netflix	0.7058	0.7368	0.7778	0.375
Outlook	0.8000	0.8000	0.8000	0.1667
Whatsapp	0.6000	0.8000	0.6667	0.6000
WordPress	1.0000	0.6000	0.5455	0.3077
YouTube	0.8888	0.6957	0.8421	0.2162
Average	0.8243	0.7771	0.7445	0.3043

Table 4.12 Summary of F1 scores for synthesis task in dataset-2

Page	Original F1	Oversample F1	SMOTE F1	Baseline F1
Adobe	0	0.5000	0.5000	0.25
Amazon	0	0.5000	0.4444	0.0909
BBC	0.6667	0.6667	0.8000	0.1905
Netflix	0.6000	0.8000	0.8000	0.4286
Outlook	0.6667	0.8750	0.8750	0.5263
Whatsapp	0.2222	0.8235	0.8235	0.4615
WordPress	0.8000	0.6000	0.6000	0.1538
YouTube	0.5714	0.3529	0.4000	0.2343
Average	0.4409	0.6398	0.6554	0.2920

better. On average, SMOTE has the most significant F1 score, 0.6554. For page-specific comparison, oversample and SMOTE, to some extent, show similar results.

4.4 Discussion

After conducting several tests on dataset-1 and dataset-2 using different classification algorithms such as SVM, KNN, Logistic Regression, Random Forest, Decision Tree, and Naive Bayes, it was observed among the different classification algorithms that the KNN classification model, performed better than the other models for both dataset-1 and dataset-2 (see Tables in Section 4.1). This work focused on the F1 scores since it finds an equal balance between Precision and Recall, as the datasets are imbalanced in terms of trending and non-trending elements. The KNN model predicted an average F1 score of 0.91 for dataset-1 for the browsing task, as shown in Table 4.1, and it predicted an average F1 score of 0.88 for the dataset-1 for the searching task, as shown in Table 4.2. While observing the other classification models, the average F1 scores for the browsing task are in the range $[0.53, 0.77]$, as seen in Table 4.1), and the average F1 scores for the searching task are in the range $[0.54, 0.83]$, as seen in Table 4.2). When observing the results of dataset-2 which consists of the browsing task and synthesis task, the KNN model demonstrated the highest average F1 score for the browsing task is 0.88, as seen in Table 4.3, and an average F1 score for the synthesis task is 0.44, as seen in Table 4.4. In contrast, the other models' values of the F1 score for the browsing task are in the range $[0.44, 0.62]$, as seen in Table 4.3, and the values of the F1 score for the synthesis task are in the range $[0.23, 0.44]$, as seen in Table 4.4. These results show that the KNN model was able to predict the trending elements on a web page with the highest F1 scores.

Different visual features could affect the prediction of the trending elements. In particular distinguishable visual features could be more likely to be trending. In browsing task participants were asked to view the web page freely for 30 seconds. Based on the results of the browsing task in Table 4.5, web pages with fewer trending elements were predicted correctly with an F1 score of 1; for example, the AVG, BBC, and GoDaddy web pages had lower percentage of trending elements compared to the non-trending (see Table 3.3). This could be because the trending elements were more distinguished in terms of visual features than the non-trending elements. For example, the trending elements of the AVG web page (See Figure 3.5) were "G" and "I". These were the largest elements which had different font colors and text sizes and they contained images. Similar case was seen in the BBC and GoDaddy web pages.

When observing the results of the browsing task in Table 4.5 for the other web pages, it can be seen that the KNN model could predict the Apple web page with an F1 score of 0.77 which is relatively low. This could be because some of the web page elements are similar in terms of feature set. Based on Figure 3.1, the trending elements of the Apple web page are "B, C, E, F, G, H, I" . However, the element "D" is similar to the element "E" and the element "J" is similar to "G, H, I", both in terms of feature set. Another example can be seen from the Babylon web page, see Figure 3.7 in Table 4.5, where the KNN model gave an F1 score of 0.82 for the browsing task. Some of the trending elements are similar to the non-trending ones. For example, the element "L" is trending, and the elements "K" and "J" are not trending even though they have similar feature sets. This could have made it hard for the model to identify the trending and non-trending elements. Similarly, dataset-2 results were consistent with the previous observations for the browsing task. Based on the results in Table 3.5 for dataset-2, the KNN model managed to predict an average F1 score of 0.82 for the original dataset for the browsing task. As seen from Table 3.5, the KNN model managed to predict the trending elements with an F1 score of 1 for Amazon, WordPress and BBC web pages. Since these web pages had few trending elements, the KNN managed to predict them correctly since these trending elements were more likely to be different from the non-trending elements. However, for web pages such as Adobe, Netflix, and Whatsapp, the F1 score was low compared to the other web pages. This is because the trending elements had similar feature set as the non-trending elements, which made it difficult for the model to distinguish. For example, in Netflix web page, (see Figure 3.2), the elements "F, G, H, I, J, K, N" are similar, where some of them such as "F, G, H, I, K" are trending, while others such as "J, N, O" are not trending.

Based on the KNN model, we can observe from the results that it possible to predict the elements of the searching task based on the collected features. However, the complexity of the website might have an effect.

The results of the searching task in Table 4.6 show that, in dataset-1, the KNN model can predict the elements with an average F1 score of 0.88 for the searching task. In the searching task participants were asked to answer some questions, this resulted in that certain visual elements had to be visited to complete the task. For example, in Apple web page (see Figure 3.1) participants were asked these two questions: (a) Can you locate the link that allows watching the TV ads relating to iPad mini? (b) Can you locate a link labeled iPad on the main menu?

The trending elements of this task are "B, C, E, F, G, H, I". These items needed to be visited to complete the task. The KNN model was able to predict the elements with F1 score of 0.83. This could be because the Apple web page has a low visual complexity based on Visual Complexity Rankings and Accessibility Metrics and the

KNN model was able to identify most of the trending elements from collected feature set. In the GoDaddy web page participants were asked to answer these two questions: (a) Can you find a telephone number for technical support and read it? (b) Can you locate a text box where you can search for a new domain?

These are the elements that needed to be visited to answer the questions "O, F, N, M, L". In terms of feature set, the element "F" and the element "L" were not unique or distinguishable, but these elements were required to be visited to answer the questions. Since "F" was the phone number and "L" was the heading title, this gave the participants information on finding the search box, it had to be included in the trending elements set. The F1 score is 0.75 for the GoDaddy web page which is relatively low compared to the other web pages. This could be because trending elements are more related with the question that the participant need to answer rather than distinguishable visual features. Another reason could be that the GoDaddy web page considered to have high complexity based on visual complexity rankings and accessibility metrics [27]. Due to the structure of the web page, locating elements to answer the questions can be difficult, which might have made it harder for the model predict the elements.

The KNN was not successfully predicting the elements for the synthesis task. The model is trained based on HTML related features, some positional features and some text features and it does not consider semantically the information the element contains which is crucial for completing the synthesis task. Also, high complexity website structure can make it hard for the model to predict the elements for such a task.

When observing the results of the synthesis task of dataset-2, the KNN model did not perform very well, the F1 score was 0.44. This could be because of the complexity of the task. In this task participants needed to answer some questions by combining multiple facts from different elements to provide a new piece of information in a maximum of 120 seconds, to answer the questions. For example, in the Netflix web page shown in Figure 3.2, these were the questions the participant had to answer: (a) Which is the cheapest plan that allows you to watch movies on your laptop, TV and tablet? (b) How much more would you have to pay compared to the basic plan if you wanted to have Ultra HD?

In order to answer these questions these elements needed to be visited "B, D, E, F, G, H" of the Netflix web page (see Figure 3.2). These elements contain the information needed to answer the questions. The KNN model was able to predict the web page elements with an F1 score of 0.60 for the Netflix web page, which is relatively low. However, for some web pages such as the Amazon and Adobe, the model was not able to identify the elements and the F1 score is zero for both web pages. Besides the complexity of the synthesis task, another factor that could have contributed to the low F1 score is that the web pages had high visual complexity based on visual complexity measured by ViCRAM [27].

Overall, the KNN model provided high F1 scores for both dataset-1 and dataset-2 when compared with the baseline. For dataset-1, the browsing task average F1 score is 0.91 while the baseline F1 score is 0.33. In addition, a similar result can be seen from the searching task where the average F1 score is 0.88 while the baseline F1 score is 0.38. Also, the results for browsing and searching tasks for page specific prediction had higher F1 scores in all the cases compared to the baseline.

In dataset-2, both the browsing task and synthesis task had a higher F1 score compared with the baseline. However, for some web pages in the synthesis task, the KNN model predictions were not able to predict the elements properly. This can be because the task is relying more on the semantic information the element holds rather than the visual features.

From the Tables 3.3 to 3.6, it can be seen that both datasets were imbalanced. For example, in dataset-1, only 8% of the AVG web page data was trending (see Table 3.3). Hence, due to the imbalanced dataset, sampling techniques were applied (oversampling and SMOTE) to the datasets and results were observed.

When observing the results of dataset-1 of the oversampled data, only Apple's web page showed a slight increase in the F1 score for when using random oversampling and SMOTE, while for the other web pages, the original data performed better for both tasks (browsing and searching).

In the browsing task of dataset-2, the F1 score of three web pages increased when using oversampling techniques as seen in Table 3.5 for the browsing task. The F1 score of Adobe web page increased from 0.60 to 0.86 for both random oversampling and SMOTE. SMOTE gave a better F1 score in the Netflix web page, while in Whatsapp web page, the random oversampling gave a better F1 score. This can be because the Netflix web page has higher number of trending elements then non-trending.

In the synthesis task, the oversampling techniques also improved the prediction results. As it can be seen from the Table 3.6 for the synthesis task, the model managed to improve the prediction score for all web pages except WordPress and YouTube web pages. Based on the average F1 score, SMOTE provided the best score of 0.66. The random oversampling only performed better on Amazon's web page with a slight difference when compared to SMOTE as seen in Table 3.6 for the synthesis task. The F1 score for random oversampling is 0.5 and the F1 score for SMOTE is 0.44.

CHAPTER 5

CONCLUSION

In this work, we first introduced the problem of identifying and predicting trending elements generated by Scanpath Trend Analysis (STA) without conducting an eye-tracking study. In the next step, we presented a detailed literature review on previously conducted works on attention prediction that investigates how to identify AOIs and predict the sequence of viewing order. Also, we explained how previous work has been limited to specific set of features or relied on using eye-tracking datasets. In addition, none of the existing work tried to predict the trending elements of the trending path generated by the STA algorithm.

Scanpath Trend Analysis (STA) algorithm is developed to cluster multiple eye-tracking scanpaths of different users into a single scanpath. It aims to find this single path in terms of trending visual elements. In this work, we manged to automatically identify the trending elements without the need to conduct an eye-tracking study. Our experiments presented in this thesis showed that, with the help of machine learning algorithms, we can develop a model that can be used to predict the trending elements. In addition, experiments with different machine learning algorithms show that the model with KNN classifier performs the best for our datasets used in this study.

In this study we validated our approach using two pre-collected datasets that included different types of tasks: browsing, searching, and synthesis. Using the KNN model, we obtained the prediction scores on individual web pages for the browsing and the searching tasks in dataset-1 and the browsing and the synthesis tasks for dataset-2. These results have been compared with the baseline scores obtained from using the Random prediction algorithm. Additionally, we also investigated the effect of sampling methods on the prediction results of the KNN model. The experiments show that the results of the predictions score of the KNN model improved with the sampled data for the synthesis task. However, for the other tasks, original data prediction scores were better.

Our study is not without limitations. Feature selection could not be applied since every web page had a different feature set. For example, as we observed in the correlation results in Table 3.9 for dataset-1, we see that the Apple web page had the following features: Image tag count, Image tag, Size, Relative size, and Aspect ratio, as the top five highly correlated ones, while the BBC web page had the following feature: Semi-bold font weight, Relative size, Size, Word count, and Span tag count, as the top five highly correlated features. Hence, the top five highly correlated features are different among the different web pages. Similar results were also observed when computing the information gain (see Table 3.7 and Table 3.8), and thus it was difficult to apply feature selection. Some of the web pages contained images that had text. Hence, they could not be analyzed to determine the font size and weight. For example, in the Apple web page in Figure 3.1 the element "C" is an image element which contains text, so we could not identify the font size and weight of the text. In order to analyze these type of elements, it would require some image processing techniques, where it will help us obtain the text to processes it and extract the required text features. Even though we used two pre-collected datasets that were collected in different institutions, with different participants, pages, and tasks, the data size is small, and therefore more data is needed. Thus, further studies can be conducted by increasing the number of web pages and/or participants. Moreover, although in this study more tasks have been considered compared with the previous research, we still have the number of tasks as a limitation, and complex tasks of different cognitive complexity could be explored [24]. Further experiments could be conducted with more different types of web pages. For example, dynamic web pages where the information displayed keeps on changing depending on the visitor. Page categorization could also be used to group web pages. For example, web pages can be grouped as mostly images, mostly text, and mixture of images and text. Despite these limitations, the results presented in this thesis show the possibility of automatically identifying trending elements without using eye-tracking data.

The work proposed here have many applications. Automatic identification of trending elements can be used to transcode web pages since transcoding with STA still requires pre-collected eye-tracking dataset. It is not feasible to collect an eye-tracking dataset for each web page, and also these web pages can be frequently changed. Identifying these trending elements will help reorganize web pages such that they are easier to access with screen readers [45], and therefore more accessible to users with disabilities.

Trending elements can also be used to give better and more efficient advertisement delivery on web pages[30]. For example, identifying trending elements can give insights on where to place advertisement to attract web users. Finally, they can also be used to better assess the usability of web pages [29] by determining the reading pattern of users. For example, identifying the reading pattern of users is important for web developers, so they know where to place the important content.

The work we have done so far concerns predicting the trending elements. In our future work, we are planning on investigating different algorithms and features to predict the path generated by STA. In this study we have already looked at previous work conducted about sequence prediction to identify the different features, and algorithms that have been used. We are planning to try to add more features which could help us in predicting the order of trending elements. In addition, we want to determine what are other possible machine learning models that we could use to predict the order of trending elements.

We are planning on collecting larger datasets so we can use it to conduct more experiments. For example, we are planning to identify the required number of participants to have sufficient data for our experiments. In addition, we want to use more web pages so we can have different varieties of web pages. For example, having a sufficient number of web pages which are: mostly images, mostly text, and a mixture of text and images could help in applying feature selection by grouping the web pages into different groups. This will also help us validate our current work, and the data can be used later on for more tests. Since in this current work we only considered the most common classification algorithm, we could try to experiment with developing a deep learning model and applying image processing techniques to extract image related features. Here having a larger dataset is also important since building a deep learning model would require a large dataset. In addition, we could also try to conduct more experiments and apply parameter tuning for the other classification algorithms instead of using the default ones. Also, we may consider developing our own custom prediction model for predicting the trending elements and the path. The custom model could consider the visual hierarchy rules mentioned in [15] and the reading patterns of users [29]. Furthermore, we want to investigate more task related predictions and try to come up with an approach to improve the prediction score for the synthesis task. We could consider applying semantic analysis to the text to extract meaning by obtaining, for example, the grammatical structure, the relationship between words, etc. We could also incorporate the conducted tasks as a feature in the feature set, which could help improve the predictions.

This current work has mainly focused on using datasets from neurotypical people, but this work could also be extended to dataset for people with autism, which have been already collected from a previous study [13]. This could help us understand the different visual web features that people with autism find attracting/distracting.

REFERENCES

[1] 2022. Puppeteer - Chrome developers. https://developers.google.com/web/tools/puppeteer

[2] M. Elgin Akpınar and Yeliz Yesilada. 2013. Vision Based Page Segmentation Algorithm: Extended and Perceived Success. In *Current Trends in Web Engineering*, Quan Z. Sheng and Jesper Kjeldskov (Eds.). Springer International Publishing, Cham, 238–252.

[3] S. S. Alam and R. Jianu. 2017. Analyzing Eye-Tracking Information in Visualization and Data Space: From Where on the Screen to What on the Screen. *IEEE Transactions on Visualization and Computer Graphics* 23, 5 (2017), 1492–1505. https://doi.org/10.1109/TVCG.2016.2535340

[4] Agustin Garcia Asuero, Ana Sayago, and AG González. 2006. The correlation coefficient: An overview. *Critical reviews in analytical chemistry* 36, 1 (2006), 41–59.

[5] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. 2002. Sequential PAttern Mining Using a Bitmap Representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Edmonton, Alberta, Canada) *(KDD '02)*. ACM, New York, NY, USA, 429–435. https://doi.org/10.1145/775047.775109

[6] Georg Buscher, Edward Cutrell, and Meredith Ringel Morris. 2009. What Do You See When You're Surfing? Using Eye Tracking to Predict Salient Regions of Web Pages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/1518701.1518705

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[8] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. 2015. Eye tracking scanpath analysis techniques on web pages: A survey, evaluation and comparison. *Journal of Eye Movement Research* 9, 1 (Dec. 2015). https://doi.org/10.16910/jemr.9.1.2

[9] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. 2016. Eye Tracking Scanpath Analysis on Web Pages: How Many Users?. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (Charleston, South Carolina) *(ETRA '16)*. ACM, New York, NY, USA, 103–110. https://doi.org/10.1145/2857491.2857519

[10] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. 2016. Scanpath Trend Analysis on Web Pages: Clustering Eye Tracking Scanpaths. *ACM Trans. Web* 10, 4, Article 20 (Nov. 2016), 35 pages. https://doi.org/10.1145/2970818

[11] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. 2016. Trends in Eye Tracking Scanpaths: Segmentation Effect?. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media* (Halifax, Nova Scotia, Canada) *(HT '16)*. Association for Computing Machinery, New York, NY, USA, 15–25. https://doi.org/10.1145/2914586.2914591

[12] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. 2017. Less users more confidence: How AOIs don't affect scanpath trend analysis. *Journal of Eye Movement Research* 10, 4, Article 6 (Nov. 2017), 8 pages. https://doi.org/10.16910/jemr.10.4.6

[13] Sukru Eraslan, Yeliz Yesilada, Victoria Yaneva, and Le An Ha. 2021. "Keep It Simple!": An Eye-Tracking Study for Exploring Complexity and Distinguishability of Web Pages for People with Autism. *Univers. Access Inf. Soc.* 20, 1 (mar 2021), 69–84. https://doi.org/10.1007/s10209-020-00708-9

[14] Sukru Eraslan, Yeliz Yesilada, Victoria Yaneva, and Simon Harper. 2020. Autism Detection Based on Eye Movement Sequences on the Web: A Scanpath Trend Analysis Approach. In *Proceedings of the 17th International Web for All Conference* (Taipei, Taiwan) *(W4A '20)*. Association for Computing Machinery, New York, NY, USA, Article 11, 10 pages. https://doi.org/10.1145/3371300.3383340

[15] Pete Faraday. 2000. Visually Critiquing Web Pages. In *Multimedia '99*, Nuno Correia, Teresa Chambel, and Glorianna Davenport (Eds.). Springer Vienna, Vienna, 155–166.

[16] Joseph H. Goldberg and Jonathan I. Helfman. 2010. Scanpath Clustering and Aggregation. In *Proceedings of the 2010 Symposium on Eye-Tracking Research &#38; Applications* (Austin, Texas) *(ETRA '10)*. ACM, New York, NY, USA, 227–234. https://doi.org/10.1145/1743666.1743721

[17] Rebecca Grier. 2004. *Visual Attention and Web Design*. Ph. D. Dissertation.

[18] Simon Harper, Sukru Eraslan, and Yeliz Yesilada. 2019. It's All About the Message: Visual Experience is a Precursor to Accurate Auditory Interaction. In *Proceedings of the 16th International Web for All Conference* (San Francisco, CA, USA) *(W4A '19)*. Association for Computing Machinery, New York, NY, USA, Article 20, 8 pages. https://doi.org/10.1145/3315002.3317554

[19] Prateek Hejmady and N. Hari Narayanan. 2012. Visual Attention Patterns During Program Debugging with an IDE. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. ACM, New York, NY, USA, 197–200. https://doi.org/10.1145/2168556.2168592

[20] Helene Hembrooke, Matt Feusner, and Geri Gay. 2006. Averaging Scan Patterns and What They Can Tell Us. In *Proceedings of the 2006 Symposium on Eye Tracking Research and Applications* (San Diego, California) *(ETRA '06)*. ACM, New York, NY, USA, 41–41. https://doi.org/10.1145/1117309.1117325

[21] Jana Holsanova, Henrik Rahm, and Kenneth Holmqvist. 2006. Entry points and reading paths on newspaper spreads: comparing a semiotic analysis with eye-tracking measurements. *Visual Communication* 5, 1 (2006), 65–93. https://doi.org/10.1177/1470357206061005

[22] L. Itti, C. Koch, and E. Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 11 (1998), 1254–1259. https://doi.org/10.1109/34.730558

[23] Ananya Jana and Samit Bhattacharya. 2015. Design and Validation of an Attention Model of Web Page Users. *Advances in Human-Computer Interaction* 2015 (02 2015), 1–14. https://doi.org/10.1155/2015/373419

[24] Jiepu Jiang, Daqing He, and James Allan. 2014. Searching, Browsing, and Clicking in a Search Session: Changes in User Behavior by Task and over Time. In *Proceedings of the 37th International ACM SIGIR Conference on Research amp; Development in Information Retrieval* (Gold Coast, Queensland, Australia) *(SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 607–616. https://doi.org/10.1145/2600428.2609633

[25] John T Kent. 1983. Information gain and a general measure of correlation. *Biometrika* 70, 1 (1983), 163–173.

[26] Dmitry Lagun and Eugene Agichtein. 2015. Inferring Searcher Attention by Jointly Modeling User Interactions and Content Salience. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 483–492. https://doi.org/10.1145/2766462.2767745

[27] Eleni Michailidou. 2006. ViCRAM: Visual Complexity Rankings and Accessibility Metrics. *SIGACCESS Access. Comput.* 86 (sep 2006), 24–27. https://doi.org/10.1145/1196148.1196154

[28] Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. 2020. Machine learning with oversampling and undersampling techniques: overview

study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)*. IEEE, 243–248.

[29] Jakob Nielsen. 2006. F-Shaped Pattern For Reading Web Content (original eye-tracking research). https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content-discovered/

[30] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2018. The Cost of Digital Advertisement: Comparing User and Advertiser Views. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1479–1489. https://doi.org/10.1145/3178876.3186060

[31] Manos Papagelis and Dimitris Plexousakis. 2005. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. *Engineering Applications of Artificial Intelligence* 18, 7 (2005), 781–789. https://doi.org/10.1016/j.engappai.2005.06.010

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[33] Kara Pernice. 2017. F-Shaped Pattern of Reading on the Web: Misunderstood, But Still Relevant (Even on Mobile). https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/

[34] Bruce Ratner. 2009. The correlation coefficient: Its values range between +1/1, or do they? *Journal of Targeting, Measurement and Analysis for Marketing* 17 (06 2009). https://doi.org/10.1057/jt.2009.5

[35] Jeremiah Still. 2017. Web page attentional priority model. *Cognition, Technology and Work* 19 (06 2017), 1–12. https://doi.org/10.1007/s10111-017-0411-9

[36] Alistair Sutcliffe and Abdallah Namoun. 2012. Predicting user attention in complex web pages. *Behaviour & Information Technology* 31, 7 (2012), 679–695. https://doi.org/10.1080/0144929X.2012.692101

[37] Christine Lourrine Tablatin and Ma. Mercedes Rodrigo. 2018. Identifying Common Code Reading Patterns using Scanpath Trend Analysis with a Tolerance. In *Proceedings of thee 26th International Conference for Computers in Education (ICCE 2018)*. Metro Manila, Philippines, 286–291.

[38] Idil Ece Trabzon, Furkan Yagiz, Elmas Eda Karadavut, Mahmoud Elhewahey, Sukru Eraslan, Yeliz Yesilada, and Simon Harper. 2022. Framework for Experiential Transcoding of Web Pages with Scanpath Trend Analysis. In *Proceedings of the 19th International Web for All Conference* (Lyon, France) *(W4A '22)*. Association for Computing Machinery, New York, NY, USA, Article 17, 5 pages. https://doi.org/10.1145/3493612.3520450

[39] S. Vidyapu, V. S. Vedula, and S. Bhattacharya. 2019. Attention Prediction on Webpage Images using Multilabel Classification. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. 1975–1980. https://doi.org/10.1109/SMC.2019.8913888

[40] Sandeep Vidyapu, Vijaya Saradhi Vedula, and Samit Bhattacharya. 2019. Quantitative Visual Attention Prediction on Webpage Images Using Multiclass SVM. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research amp; Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 90, 9 pages. https://doi.org/10.1145/3317960.3321614

[41] Sandeep Vidyapu, Vijaya Saradhi Vedula, and Samit Bhattacharya. 2020. Investigating and Modeling the Web Elements' Visual Feature Influence on Free-Viewing Attention. 15, 1 (2020). https://doi.org/10.1145/3409474

[42] Sandeep Vidyapu, Vijaya Saradhi Vedula, and Samit Bhattacharya. 2020. Weighted Voting-Based Effective Free-Viewing Attention Prediction On Web Image Elements. *Interacting with Computers*

32, 2 (07 2020), 170–184. https://doi.org/10.1093/iwcomp/iwaa013 arXiv:https://academic.oup.com/iwc/article-pdf/32/2/170/33646090/iwaa013.pdf

[43] Julia M. West, Anne R. Haake, Evelyn P. Rozanski, and Keith S. Karn. 2006. eyePatterns: Software for Identifying Patterns and Similarities Across Fixation Sequences. In *Proceedings of the 2006 Symposium on Eye Tracking Research &Amp; Applications* (San Diego, California) *(ETRA '06)*. ACM, New York, NY, USA, 149–154. https://doi.org/10.1145/1117309.1117360

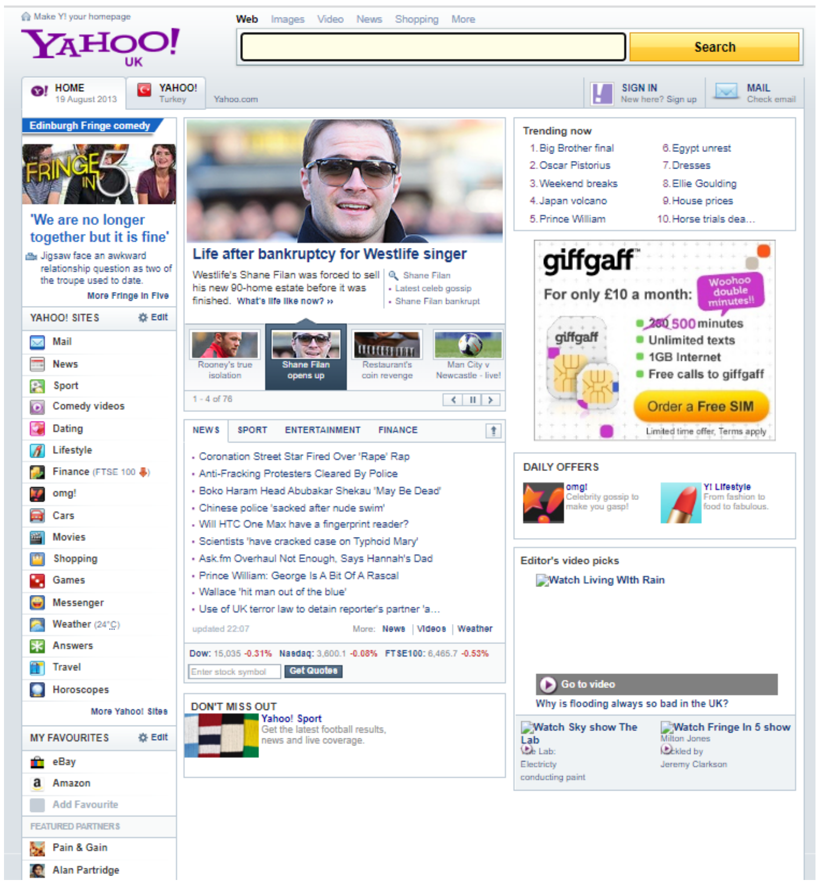
[44] C. Xia and R. Quan. 2020. Predicting Saccadic Eye Movements in Free Viewing of Webpages. *IEEE Access* 8 (2020), 15598–15610. https://doi.org/10.1109/ACCESS.2020.2966628

[45] Yeliz Yesilada, Simon Harper, and Sukru Eraslan. 2013. Experiential Transcoding: An EyeTracking Approach. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility* (Rio de Janeiro, Brazil) *(W4A '13)*. Association for Computing Machinery, New York, NY, USA, Article 30, 4 pages. https://doi.org/10.1145/2461121.2461134

APPENDIX A

MATERIAL

A.1 Dataset-1 web pages screenshots



Make Y! your homepage

Web Images Video News Shopping More

Search

HOME 19 August 2013 | YAHOO! Turkey | Yahoo.com | SIGN IN New here? Sign up | MAIL Check email

Edinburgh Fringe comedy

'We are no longer together but it is fine'

Jigsaw face an awkward relationship question as two of the troupe used to date.

More Fringe in Five

YAHOO! SITES Edit

- Mail
- News
- Sport
- Comedy videos
- Dating
- Lifestyle
- Finance (FTSE 100)
- omg!
- Cars
- Movies
- Shopping
- Games
- Messenger
- Weather (24°C)
- Answers
- Travel
- Horoscopes

More Yahoo! Sites

MY FAVOURITES Edit

- eBay
- Amazon
- Add Favourite

FEATURED PARTNERS

- Pain & Gain
- Alan Partridge

Life after bankruptcy for Westlife singer

Westlife's Shane Filan was forced to sell his new 90-home estate before it was finished. What's life like now? »

Shane Filan
- Latest celeb gossip
- Shane Filan bankrupt

Rooney's true isolation | Shane Filan opens up | Restaurant's coin revenge | Man City v Newcastle - live!

1 - 4 of 76

NEWS SPORT ENTERTAINMENT FINANCE

- Coronation Street Star Fired Over 'Rape' Rap
- Anti-Fracking Protesters Cleared By Police
- Boko Haram Head Abubakar Shekau 'May Be Dead'
- Chinese police 'sacked after nude swim'
- Will HTC One Max have a fingerprint reader?
- Scientists 'have cracked case on Typhoid Mary'
- Ask.fm Overhaul Not Enough, Says Hannah's Dad
- Prince William: George Is A Bit Of A Rascal
- Wallace 'hit man out of the blue'
- Use of UK terror law to detain reporter's partner 'a...

updated 22:07 More: News | Videos | Weather

Dow: 15,035 -0.31% Nasdaq: 3,600.1 -0.08% FTSE100: 6,465.7 -0.53%

Enter stock symbol Get Quotes

DON'T MISS OUT

Yahoo! Sport
Get the latest football results, news and live coverage.

Trending now

1. Big Brother final
2. Oscar Pistorius
3. Weekend breaks
4. Japan volcano
5. Prince William
6. Egypt unrest
7. Dresses
8. Ellie Goulding
9. House prices
10. Horse trials dea...

giffgaff™

For only £10 a month:

Woohoo double minutes!!

- ~~250~~ 500 minutes
- Unlimited texts
- 1GB Internet
- Free calls to giffgaff

Order a Free SIM

Limited time offer, Terms apply

DAILY OFFERS

omg!
Celebrity gossip to make you gasp!

Y! Lifestyle
From fashion to food to fabulous.

Editor's video picks

Watch Living With Rain

Go to video

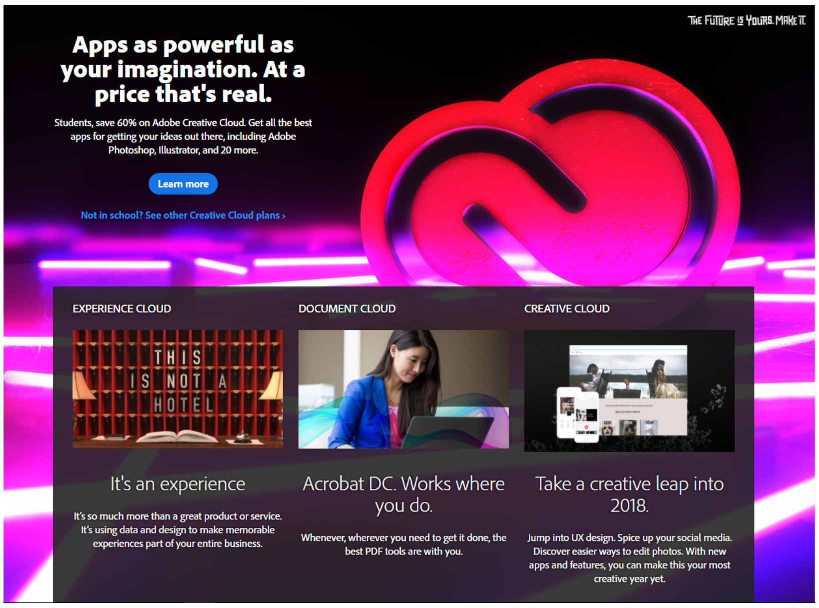
Why is flooding always so bad in the UK?

Watch Sky show The Lab
e Lab: Electricty conducting paint

Watch Fringe In 5 show
Milton Jones
ckled by Jeremy Clarkson

Figure A.1 Yahoo web page [10]

A.2 Dataset-2 web pages screenshots


THE FUTURE IS YOURS. MAKE IT.

Apps as powerful as your imagination. At a price that's real.

Students, save 60% on Adobe Creative Cloud. Get all the best apps for getting your ideas out there, including Adobe Photoshop, Illustrator, and 20 more.

Learn more

Not in school? See other Creative Cloud plans ›

EXPERIENCE CLOUD

THIS IS NOT A HOTEL

It's an experience

It's so much more than a great product or service. It's using data and design to make memorable experiences part of your entire business.

DOCUMENT CLOUD

Acrobat DC. Works where you do.

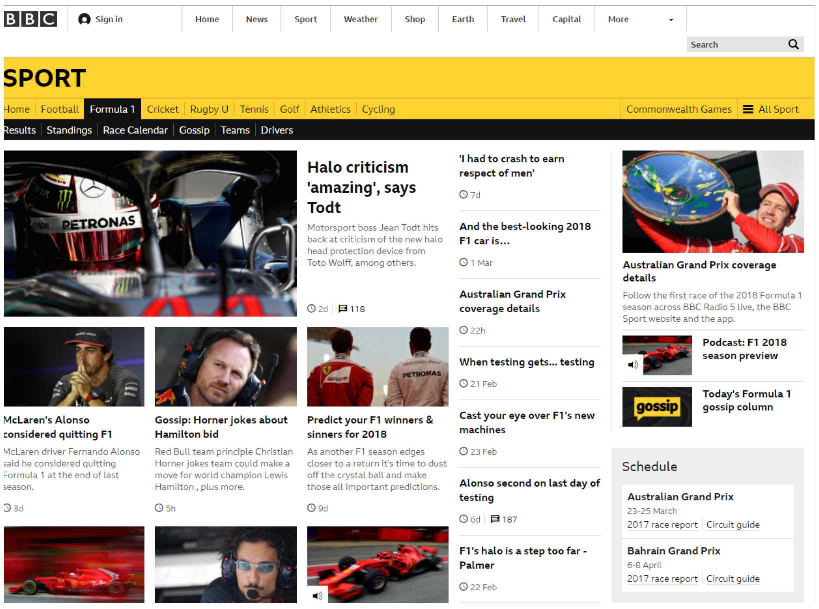
Whenever, wherever you need to get it done, the best PDF tools are with you.

CREATIVE CLOUD

Take a creative leap into 2018.

Jump into UX design. Spice up your social media. Discover easier ways to edit photos. With new apps and features, you can make this your most creative year yet.

Figure A.2 Adobe web page [13]


BBC | Sign in | Home | News | Sport | Weather | Shop | Earth | Travel | Capital | More

Search

SPORT

Home | Football | Formula 1 | Cricket | Rugby U | Tennis | Golf | Athletics | Cycling | Commonwealth Games | All Sport

Results | Standings | Race Calendar | Gossip | Teams | Drivers

Halo criticism 'amazing', says Todt

Motorsport boss Jean Todt hits back at criticism of the new halo head protection device from Toto Wolff, among others.

2d | 118

'I had to crash to earn respect of men'

7d

And the best-looking 2018 F1 car is...

1 Mar

Australian Grand Prix coverage details

22h

When testing gets... testing

21 Feb

Cast your eye over F1's new machines

23 Feb

Alonso second on last day of testing

6d | 187

F1's halo is a step too far - Palmer

22 Feb

McLaren's Alonso considered quitting F1

McLaren driver Fernando Alonso said he considered quitting Formula 1 at the end of last season.

3d

Gossip: Horner jokes about Hamilton bid

Red Bull team principle Christian Horner jokes team could make a move for world champion Lewis Hamilton , plus more.

5h

Predict your F1 winners & sinners for 2018

As another F1 season edges closer to a return it's time to dust off the crystal ball and make those all important predictions.

9d

Australian Grand Prix coverage details

Follow the first race of the 2018 Formula 1 season across BBC Radio 5 live, the BBC Sport website and the app.

Podcast: F1 2018 season preview

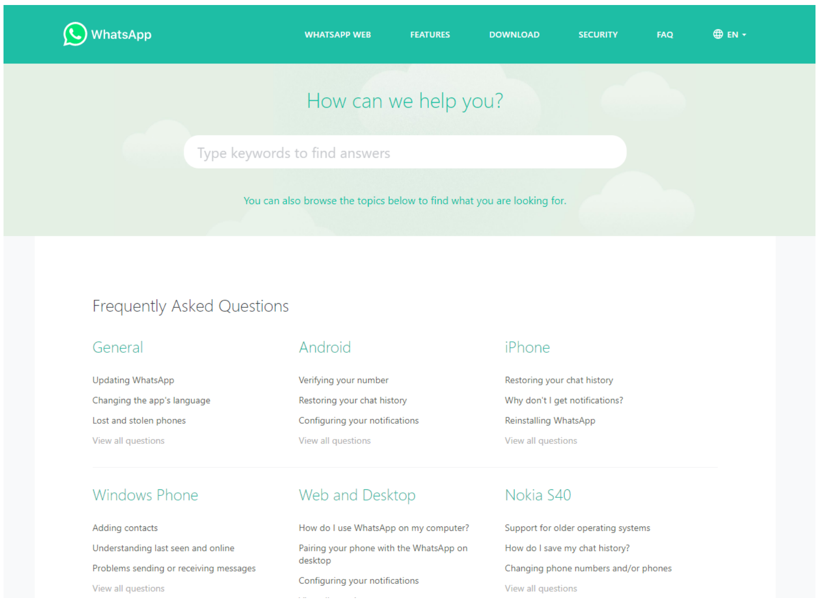
Today's Formula 1 gossip column

Schedule

Australian Grand Prix
23-25 March
2017 race report | Circuit guide

Bahrain Grand Prix
6-8 April
2017 race report | Circuit guide

Figure A.3 BBC web page[13]



How can we help you?

Type keywords to find answers

You can also browse the topics below to find what you are looking for.

Frequently Asked Questions

General

Updating WhatsApp
Changing the app's language
Lost and stolen phones
View all questions

Android

Verifying your number
Restoring your chat history
Configuring your notifications
View all questions

iPhone

Restoring your chat history
Why don't I get notifications?
Reinstalling WhatsApp
View all questions

Windows Phone

Adding contacts
Understanding last seen and online
Problems sending or receiving messages
View all questions

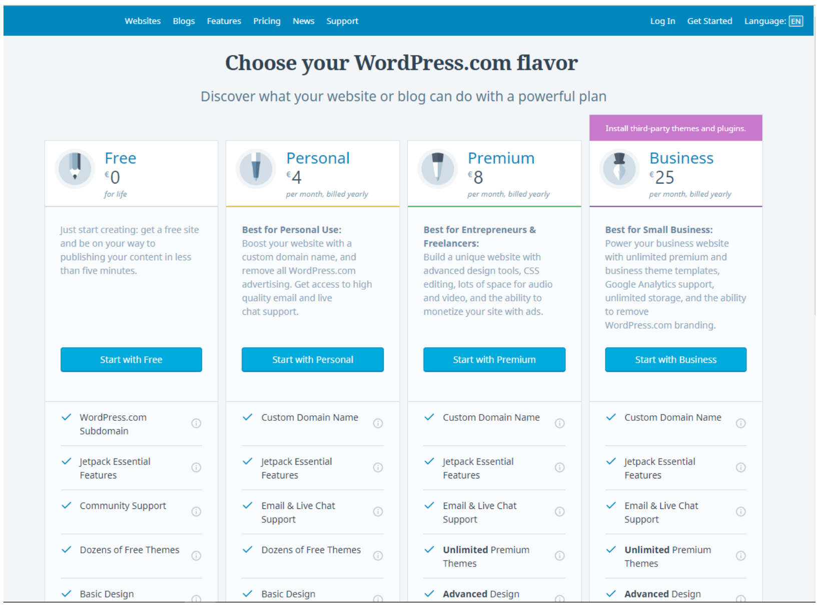
Web and Desktop

How do I use WhatsApp on my computer?
Pairing your phone with the WhatsApp on desktop
Configuring your notifications

Nokia S40

Support for older operating systems
How do I save my chat history?
Changing phone numbers and/or phones
View all questions

Figure A.4 WhatsApp web page [13]



Choose your WordPress.com flavor

Discover what your website or blog can do with a powerful plan

Install third-party themes and plugins.

Free
€0
for life

Just start creating: get a free site and be on your way to publishing your content in less than five minutes.

Start with Free

✓ WordPress.com Subdomain
✓ Jetpack Essential Features
✓ Community Support
✓ Dozens of Free Themes
✓ Basic Design

Personal
€4
per month, billed yearly

Best for Personal Use: Boost your website with a custom domain name, and remove all WordPress.com advertising. Get access to high quality email and live chat support.

Start with Personal

✓ Custom Domain Name
✓ Jetpack Essential Features
✓ Email & Live Chat Support
✓ Dozens of Free Themes
✓ Basic Design

Premium
€8
per month, billed yearly

Best for Entrepreneurs & Freelancers: Build a unique website with advanced design tools, CSS editing, lots of space for audio and video, and the ability to monetize your site with ads.

Start with Premium

✓ Custom Domain Name
✓ Jetpack Essential Features
✓ Email & Live Chat Support
✓ **Unlimited** Premium Themes
✓ **Advanced** Design

Business
€25
per month, billed yearly

Best for Small Business: Power your business website with unlimited premium and business theme templates, Google Analytics support, unlimited storage, and the ability to remove WordPress.com branding.

Start with Business

✓ Custom Domain Name
✓ Jetpack Essential Features
✓ Email & Live Chat Support
✓ **Unlimited** Premium Themes
✓ **Advanced** Design

Figure A.5 Wordpress web page [13]

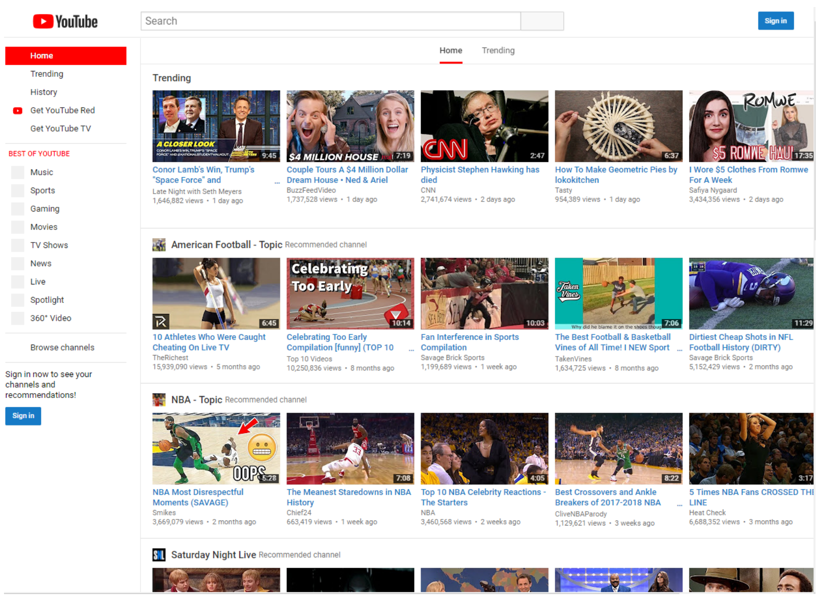

YouTube
Search
Sign in
Home
Trending
History
Get YouTube Red
Get YouTube TV
BEST OF YOUTUBE
Music
Sports
Gaming
Movies
TV Shows
News
Live
Spotlight
360° Video
Browse channels
Sign in now to see your channels and recommendations!
Sign in
Trending
Conor Lamb's Win, Trump's "Space Force" and
Couple Tours A $4 Million Dollar Dream House • Ned & Ariel
Physicist Stephen Hawking has died
How To Make Geometric Pies by lokokitchen
I Wore $5 Clothes From Romwe For A Week
American Football - Topic Recommended channel
10 Athletes Who Were Caught Cheating On Live TV
Celebrating Too Early Compilation [funny] (TOP 10
Fan Interference in Sports Compilation
The Best Football & Basketball Vines of All Time! I NEW Sport
Dirtiest Cheap Shots in NFL Football History (DIRTY)
NBA - Topic Recommended channel
NBA Most Disrespectful Moments (SAVAGE)
The Meanest Staredowns in NBA History
Top 10 NBA Celebrity Reactions - The Starters
Best Crossovers and Ankle Breakers of 2017-2018 NBA
5 Times NBA Fans CROSSED THE LINE
Saturday Night Live Recommended channel

Figure A.6 YouTube web page [13]

A.3 Full list of HTML tags

The Table A.1 shows all the HTML tags used in this study.

Table A.1 HTML Tags

Group	Features
HTML Tags	Div Count, Div, li count, Li, input count, input , form count, form, span count, span, a count, a, Img_count, image, h1 count, h1, h2 count, h2, h3 count, h3, h4 count, h4, I count, I, ul count, ul abbr count, abbr, button count, button, ol count, ol, dd count, dd, dt count, dt, dl count, dl td count, td, tr count, tr, tbody count, tbody, table count, table, g:plusone count, g:pluson, br count, br, strong count , strong, fieldset count, fieldset iframe count, iframe, b count, b, em count, em

A.4 Experiment Configuration Settings

The Table A.2 shows the configuration for the models used in this study.

Table A.2 Classification Models Configuration

Model	**Configuration**
SVM Linear	(pepenalty='l2', loss='squared_hinge', *, dual=True, tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=100)0
Logistic Regression	(penalty='l2',$*$, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
Decision Tree	(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0)
Random Forest	(n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None)
KNN	(*, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None)
GaussianNB	(*, priors=None, var_smoothing=1e-09)
MultinomialNB	(*, alpha=1.0, fit_prior=True, class_prior=None)
ComplementNB	(*, alpha=1.0, fit_prior=True, class_prior=None, norm=False)
BernoulliNB	(*, alpha=1.0, binarize=0.0, fit_prior=True, class_prior=None))

APPENDIX B

RESULTS

This chapter shows the results of all the classification algorithms conducted in this study for dataset-1 and dataset-2.

B.1 Browsing Task Results Dataset-1

B.1.1 KNN Results

Table B.1 Results of original browsing task in dataset-1

Page	K values	Distance	Recall	Precision	Accuracy	F1
Apple	13	Manhattan	0.7143	0.8333	0.8333	0.7692
AVG	2	Manhattan	1.0000	1.0000	1.0000	1.0000
Babylon	9	Chebyshev, Manhattan	0.8750	0.7778	0.8636	0.8235
BBC	30	All distances except Manhattan	1.0000	1.0000	1.0000	1.0000
GoDaddy	11	All distances	1.0000	1.0000	1.0000	1.0000
Yahoo	4	All distances	1.0000	0.7500	0.9000	0.8571
Average			0.9316	0.8935	0.9328	0.9083

Table B.2 Results of oversample task in dataset-1

Page	K values	Distance	Recall	Precision	Accuracy	F1
Apple	11	All distances	0.8571	0.8571	0.8889	0.8571
Avg	4	All distances	1.0000	1.0000	1.0000	1.0000
Babylon	3	Manhattan	0.8750	0.7778	0.8636	0.8235
BBC	8	All distances	1.0000	0.5714	0.8571	0.7273
GoDaddy	11	All distances	1.0000	1.0000	1.0000	1.0000
Yahoo	5	All distances	1.0000	0.6000	0.8000	0.7500
Average			0.9554	0.8011	0.9016	0.8597

Table B.3 Results of SMOTE browsing task in dataset-1

Page	**K values**	**Distance**	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	19	All distances	0.8571	0.8571	0.8889	0.8571
Avg	4	All distances	1. 0000	1. 0000	1. 0000	1.0000
Babylon	7	All distances	0.8750	0.7778	0.8636	0.8235
BBC	8	All distances	1.0000	0.5714	0.8571	0.7273
GoDaddy	5	All distances	1.0000	1. 0000	1. 0000	1. 0000
Yahoo	2	Euclidean, Chebyshev	1.0000	0.7500	0.9000	0.8571
Average			0.9554	0.8261	0.9183	0.8775

B.1.2 SVM Results

Table B.4 Results of SVM browsing task in dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0.4571	0.8222	0.7700	0.5464
AVG	0.8650	0.6250	0.9376	0.6952
Babylon	0.2225	0.3878	0.6700	0.2249
BBC	0.8275	0.5057	0.8424	0.6118
GoDaddy	0.8100	0.8710	0.9556	0.8261
Yahoo	0.8167	0.5905	0.7940	0.6575
Average	0.6665	0.6337	0.8283	0.5937

B.1.3 Gaussian Naive Bayes Results

Table B.5 Results of GaussianNB browsing task in dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0.1429	1.0000	0.6667	0.2500
AVG	1.0000	0.6667	0.9600	0.8000
Babylon	0	0	0.6364	0
BBC	1.0000	0.5714	0.8571	0.7273
GoDaddy	1.0000	0.7500	0.9375	0.8571
Yahoo	1.0000	0.4286	0.6000	0.6000
Average	0.6905	0.5695	0.7763	0.5391

B.1.4 Multinomial Naive Bayes Results

Table B.6 Results of MultinomialNB browsing task in dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0.5714	1.0000	0.8333	0.7273
AVG	1.0000	1.0000	1.0000	1.0000
Babylon	0.5000	0.5714	0.6818	0.5333
BBC	1.0000	0.4444	0.7619	0.6154
GoDaddy	1.0000	1.0000	1.0000	1.0000
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.8452	0.7693	0.8462	0.7710

B.1.5 Complement Naive Bayes Results

Table B.7 Results of ComplementNB browsing task in dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0.5714	1.0000	0.8333	0.7273
AVG	1.0000	1.0000	1.0000	1.0000
Babylon	0.5000	0.5714	0.6818	0.5333
BBC	1.0000	0.4444	0.7619	0.6154
GoDaddy	1.0000	1.0000	1.0000	1.0000
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.8452	0.7693	0.8462	0.7710

B.1.6 Bernoulli Naive Bayes Results

Table B.8 Results of BernoulliNB browsing task in dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0	0	0.6111	0
AVG	1.0000	0.6667	0.9600	0.8000
Babylon	0.0000	0.0000	0.6364	0.0000
BBC	1.0000	0.5000	0.8095	0.6667
GoDaddy	1.0000	1.0000	1.0000	1.0000
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.6667	0.4611	0.8028	0.5361

B.1.7 Random Forest Results

Table B.9 Results of Random Forest browsing task in dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0.2229	0.9600	0.6978	0.3563
AVG	1.0000	0.6240	0.9432	0.7556
Babylon	0.0000	0.0000	0.6359	0.0000
BBC	1.0000	0.5724	0.8576	0.7280
GoDaddy	0.6800	1.0000	0.9400	0.8080
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.6505	0.6261	0.8124	0.5663

B.1.8 Decision Tree Results

Table B.10 Results of Decision Tree browsing task in dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0.3143	0.9790	0.7267	0.4551
AVG	1.0000	0.6667	0.9600	0.8000
Babylon	0.6288	0.8332	0.8045	0.7004
BBC	0.5000	0.4710	0.7957	0.4839
GoDaddy	0.6667	1.0000	0.9375	0.8000
Yahoo	0.8867	0.5917	0.7880	0.6986
Average	0.6661	0.7569	0.8354	0.6563

B.1.9 Logistic Regression Results

Table B.11 Results of Logistic Regression browsing task in dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0.2857	1.0000	0.7222	0.4444
AVG	1.0000	1.0000	1.0000	1.0000
Babylon	0.1250	0.5000	0.6364	0.2000
BBC	1.0000	0.5714	0.8571	0.7273
GoDaddy	1.0000	1.0000	1.0000	1.0000
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.7351	0.7786	0.8360	0.6870

B.1.10 Baseline Results

Table B.12 Results of baseline browsing task in dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0.7142	0.5000	0.6111	0.5882
Avg	0.5000	0.0833	0.5200	0.1429
Babylon	0.5000	0.3636	0.5000	0.4211
BBC	0.5000	0.1818	0.4762	0.2667
GoDaddy	0.6667	0.2222	0.5000	0.3333
Yahoo	0.3333	0.2000	0.4000	0.2500
Average	0.5357	0.2585	0.5012	0.3337

B.2 Searching Task Results Dataset-1

B.2.1 KNN Results

Table B.13 Results of original data for searching task dataset-1

Page	K values	Distance	Recall	Precision	Accuracy	F1
Apple	20	Manhattan	0.7143	1.0000	0.8889	0.8333
AVG	4	All distances	1.0000	1.0000	1.0000	1.0000
Babylon	8	Manhattan	1.0000	0.8889	0.9545	0.9412
BBC	7	All distances	1.0000	0.8571	0.9524	0.9231
GoDaddy	5	All distances	0.6000	1.0000	0.8750	0.7500
Yahoo	2	Manhattan	1.0000	0.7500	0.9000	0.8571
Average			0.8857	0.9160	0.9285	0.8841

Table B.14 Results of oversampled data for searching task dataset-1

Page	K values	Distance	Recall	Precision	Accuracy	F1
Apple	9	All distances	0.8571	0.8571	0.8889	0.8571
AVG	4	All distances	1.0000	1.0000	1.0000	1.0000
Babylon	3	All distances	1.0000	0.8000	0.9091	0.8889
BBC	12	All distances	1.0000	0.8571	0.9524	0.9231
GoDaddy	5	All distances	0.6000	1.0000	0.8750	0.7500
Yahoo	All values	All distances	1.0000	0.6000	0.8000	0.7500
Average			0.9095	0.8524	0.9042	0.8615

Table B.15 Results of SMOTE data for searching task dataset-1

Page	**K values**	**Distance**	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	19	All distances	0.8571	0.8571	0.8889	0.8571
Avg	4	All distances	1.0000	1.0000	1.0000	1.0000
Babylon	2	All distances	1.0000	0.8000	0.9091	0.8889
BBC	10	All distances	1.0000	0.8571	0.9524	0.9231
GoDaddy	5	All distances	0.6000	1.0000	0.8750	0.7500
Yahoo	All values	All distances	1.0000	0.6000	0.8000	0.7500
Average			0.9095	0.8524	0.9042	0.8615

B.2.2 SVM Results

Table B.16 Results of SVM searching task dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0.3857	0.7690	0.7439	0.4627
AVG	0.7700	0.6282	0.9400	0.6638
Babylon	0.2250	0.3815	0.6900	0.2383
BBC	0.7583	0.6971	0.8690	0.6947
GoDaddy	0.5120	0.9660	0.8462	0.6600
Yahoo	0.8567	0.5239	0.7610	0.6337
Average	0.5846	0.6609	0.8084	0.5589

B.2.3 Gaussian Naive Bayes Results

Table B.17 Results of GaussianNB searching task dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0.1429	1.0000	0.6667	0.2500
AVG	1.0000	0.6667	0.9600	0.8000
Babylon	0.0000	0.0000	0.6364	0.0000
BBC	1.0000	0.8571	0.9524	0.9231
GoDaddy	0.6000	0.7500	0.8125	0.6667
Yahoo	1.0000	0.4286	0.6000	0.6000
Average	0.6238	0.6171	0.7713	0.5400

B.2.4 Multinomial Naive Bayes Results

Table B.18 Results of MultinomialNB searching task dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0.7143	1.0000	0.8889	0.8333
AVG	1.0000	1.0000	1.0000	1.0000
Babylon	0.7500	0.7500	0.8182	0.7500
BBC	1.0000	0.6667	0.8571	0.8000
GoDaddy	0.6000	1.0000	0.8750	0.7500
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.8441	0.8361	0.8732	0.8139

B.2.5 Complement Naive Bayes Results

Table B.19 Results of ComplementNB searching task dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0.7143	1.0000	0.8889	0.8333
AVG	1.0000	1.0000	1.0000	1.0000
Babylon	0.7500	0.7500	0.8182	0.7500
BBC	1.0000	0.6667	0.8571	0.8000
GoDaddy	0.6000	1.0000	0.8750	0.7500
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.8441	0.8361	0.8732	0.8139

B.2.6 Bernoulli Naive Bayes Results

Table B.20 Results of BernoulliNB searching task dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0	0	0.6111	0
AVG	1.0000	0.2857	0.8000	0.4444
Babylon	0.2500	1.0000	0.7273	0.4000
BBC	1.0000	0.6667	0.8571	0.8000
GoDaddy	0.6000	1.0000	0.8750	0.7500
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.6417	0.5921	0.7784	0.5241

B.2.7 Random Forest Results

Table B.21 Results of Random Forest searching task dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0.5000	0.8995	0.7806	0.6129
AVG	1.0000	0.4810	0.9108	0.6471
Babylon	0.3875	0.8628	0.7518	0.5169
BBC	1.0000	0.7367	0.8948	0.8467
GoDaddy	0.5980	1.0000	0.8744	0.7482
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.7476	0.7633	0.8354	0.6870

B.2.8 Decision Tree Results

Table B.22 Results of Decision Tree searching task dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0.7143	0.7762	0.8067	0.7429
AVG	1.0000	0.5017	0.8944	0.6507
Babylon	0.7562	0.9463	0.8918	0.8311
BBC	1.0000	0.6487	0.8443	0.7865
GoDaddy	0.4660	1.0000	0.8331	0.6304
Yahoo	1.0000	0.4213	0.5690	0.5895
Average	0.8228	0.7157	0.8066	0.7052

B.2.9 Logistic Regression Results

Table B.23 Results of Logistic Regression searching task dataset-1

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Apple	0.4286	1.0000	0.7778	0.6000
AVG	1.0000	1.0000	1.0000	1.0000
Babylon	0.8750	0.7778	0.8636	0.8235
BBC	1.0000	0.7500	0.9048	0.8571
GoDaddy	0.6000	1.0000	0.8750	0.7500
Yahoo	1.0000	0.6000	0.8000	0.7500
Average	0.8173	0.8546	0.8702	0.7968

B.2.10 Baseline Results

Table B.24 Results of baseline for searching task dataset-1

Page	Recall	Precision	Accuracy	F1
Apple	0.7143	0.5000	0.6111	0.5882
Avg	0.5000	0.0833	0.5200	0.1429
Babylon	0.6250	0.4545	0.5909	0.5263
BBC	0.5000	0.2727	0.4762	0.3529
GoDaddy	0.6000	0.3333	0.5000	0.4286
Yahoo	0.3333	0.2000	0.4000	0.2500
Average	0.5454	0.3073	0.5164	0.3815

B.3 Browsing Task Results Dataset-2

B.3.1 KNN Model Results

Table B.25 Results of original browsing task in dataset-2

Page	K Value	Distance	Recall	Precision	Accuracy	F1
Adobe	3	All distances	0.4285	1.0000	0.6000	0.6000
Amazon	50	All distances	1.0000	1.0000	1.0000	1.0000
BBC	3	All distances	1.0000	1.0000	1.0000	1.0000
Netflix	7	Chebyshev	0.7500	0.6666	0.6428	0.7058
Outlook	2	All distances	1.0000	0.6666	0.9411	0.8000
Whatsapp	29	Chebyshev	0.7500	0.5000	0.6666	0.6000
Wordpress	2	All distances	1.0000	1.0000	1.0000	1.0000
YouTube	3	All distances	1.0000	0.8000	0.9642	0.8888
Average			0.8661	0.8292	0.8518	0.8243

Table B.26 Results of oversample browsing task in dataset-2

Page	K Value	Distance	Recall	Precision	Accuracy	F1
Adobe	11	All distances	0.8571	0.8571	0.8000	0.8571
Amazon	9	All distances	1.0000	0.5714	0.5714	0.7273
BBC	4	All distances	1.0000	1.0000	1.0000	1.0000
Netflix	5	Manhattan	0.7500	0.7500	0.7143	0.7500
Outlook	3	All distances	1.0000	0.6667	0.9412	0.8000
WhatsApp	2	All distances	1.0000	0.6667	0.8333	0.8000
WordPress	2	Chebyshev	1.0000	1.0000	1.0000	1.0000
YouTube	8	All distances	1.0000	0.5333	0.8750	0.6957
Average			0.9509	0.7557	0.8419	0.8288

Table B.27 Results of SMOTE browsing task in dataset-2

Page	K Value	Distance	Recall	Precision	Accuracy	F1
Adobe	11	Euclidean	0.8571	0.8571	0.8000	0.8571
Amazon	9	Euclidean	1.0000	0.5000	0.9024	0.6667
BBC	4	Manhattan	1	1	1	1
Netflix	7	Euclidean, Manhattan	0.8750	0.7000	0.7143	0.7778
Outlook	2	All distances	1.0000	0.6667	0.9412	0.8000
WhatsApp	2	Euclidean, Manhattan	1.0000	0.5000	0.6667	0.6667
WordPress	12	Chebyshev	1.0000	0.4286	0.7647	0.6000
YouTube	6	Euclidean, minkowski	1.0000	0.7273	0.9464	0.8421
Average			0.9665	0.6725	0.8420	0.7763

B.3.2 SVM Results

Table B.28 Results of SVM browsing task in dataset-2

Page	Recall	Precision	Accuracy	F1
Adobe	0.5328	0.6906	0.5739	0.5545
Amazon	0.895	0.5737	0.9207	0.6756
BBC	0.955	0.6526	0.9625	0.7493
Netflix	0.1225	0.3840	0.4328	0.1773
Outlook	0.9000	0.5973	0.9317	0.7177
WhatsApp	0.515	0.3057	0.5616	0.3391
WordPress	0.9200	0.5192	0.8205	0.642
YouTube	0.3637	0.5194	0.8780	0.3539
Average	0.6505	0.5303	0.7602	0.5262

B.3.3 Gaussian Naive Bayes Results

Table B.29 Results of GaussianNB browsing task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0	0	0.3000	0
Amazon	1.0000	0.6666	0.9512	0.8000
BBC	1.0000	0.6666	0.9743	0.8000
Netflix	0	0	0.4285	0
Outlook	1.0000	0.6666	0.9411	0.8000
WhatsApp	0.7500	0.5000	0.6666	0.6000
WordPress	1.0000	0.5000	0.8235	0.6666
YouTube	0.1250	0.5000	0.8571	0.2000
Average	0.6094	0.4375	0.7428	0.4833

B.3.4 Multinomial Naive Bayes Results

Table B.30 Results of MultinomialNB browsing task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	1.0000	0.7777	0.8000	0.8750
Amazon	1.0000	0.4444	0.8780	0.6153
BBC	1.0000	0.5000	0.9487	0.6666
Netflix	0.2500	0.6666	0.5000	0.3636
Outlook	1.0000	0.6666	0.9411	0.8000
WhatsApp	1.0000	0.3636	0.4166	0.5333
WordPress	1.0000	0.4285	0.7647	0.6000
YouTube	0.5000	0.6666	0.8928	0.5714
Average	0.8438	0.5643	0.7499	0.6282

B.3.5 Complement Naive Bayes Results

Table B.31 Results of ComplementNB browsing task in dataset-2

Page	Recall	Precision	Accuracy	F1
Adobe	1.0000	0.7777	0.8000	0.8750
Amazon	1.0000	0.4444	0.8780	0.6153
BBC	1.0000	0.5000	0.9487	0.6666
Netflix	0.2500	0.6666	0.5000	0.3636
Outlook	1.0000	0.6666	0.9411	0.8000
WhatsApp	1.0000	0.3636	0.4166	0.5333
WordPress	1.0000	0.4285	0.7647	0.6000
YouTube	0.5000	0.6666	0.8928	0.5714
Average	0.8438	0.5643	0.7499	0.6282

B.3.6 Bernoulli Naive Bayes Results

Table B.32 Results of BernoulliNB browsing task in dataset-2

Page	Recall	Precision	Accuracy	F1
Adobe	0	0	0.3	0
Amazon	1.0000	1.0000	1.0000	1.0000
BBC	1.0000	0.3333	0.8974	0.5000
Netflix	0	0	0.4285	0
Outlook	0	0	0.8823	0
WhatsApp	0.7500	0.5000	0.6666	0.6000
WordPress	1.0000	0.6000	0.8823	0.7499
YouTube	1.0000	0.5333	0.8750	0.6956
Average	0.5357	0.3708	0.7415	0.4432

B.3.7 Random Forest Results

Table B.33 Results of Random Forest browsing task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0	0	0.3000	0
Amazon	0.6775	0.5640	0.9192	0.6119
BBC	1.0000	0.6666	0.9743	0.7999
Netflix	0	0	0.4285	0
Outlook	0.4700	0.8800	0.9376	0.6066
WhatsApp	0.7825	0.5180	0.6758	0.6126
WordPress	0.9966	0.5507	0.8476	0.7049
YouTube	0.0725	0.5450	0.8662	0.1268
Average	0.4883	0.4542	0.7437	0.4328

B.3.8 Decision Tree Results

Table B.34 Results of Decision Tree browsing task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0.4285	0.7500	0.5000	0.5454
Amazon	0.7500	0.5000	0.9024	0.6000
BBC	0.7800	0.5799	0.9582	0.9582
Netflix	0	0	0.4285	0
Outlook	0.5000	1.0000	0.9411	0.6666
WhatsApp	1.0000	0.5000	0.6666	0.6666
WordPress	1.0000	0.7500	0.9411	0.8571
YouTube	0.4188	0.5519	0.8598	0.4603
Average	0.6369	0.5790	0.7747	0.5943

B.3.9 Logistic Regression Results

Table B.35 Results of Logistic Regression browsing task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0.14285	1.0000	0.3999	0.2500
Amazon	1.0000	0.3999	0.8536	0.5714
BBC	1.0000	0.5000	0.9487	0.6666
Netflix	0.2500	0.6666	0.5000	0.3636
Outlook	1.0000	0.6666	0.9411	0.7999
WhatsApp	1.0000	0.4444	0.5833	0.6153
WordPress	1.0000	0.3333	0.6470	0.5000
YouTube	0	0	0.8392	0
Average	0.6741	0.5014	0.7141	0.4709

B.3.10 Baseline Results

Table B.36 Results of baseline for browsing task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0.4286	0.6000	0.4000	0.5000
Amazon	0.5000	0.1053	0.5366	0.1739
BBC	0.5000	0.0526	0.5128	0.0952
Netflix	0.3750	0.375	0.2857	0.375
Outlook	0.5000	0.1000	0.4118	0.1667
WhatsApp	0.7500	0.5000	0.6667	0.6000
WordPress	0.6667	0.2000	0.4706	0.3077
YouTube	0.5000	0.1379	0.4821	0.2162
Average	0.5275	0.2589	0.4708	0.3043

B.4 Synthesis Task Results Dataset-2

B.4.1 KNN Results

Table B.37 Results of original synthesis task in dataset-2

Page	K Value	Distance	Recall	Precision	Accuracy	F1
Adobe	14	Euclidean	0	0	0.7000	0
Amazon	2	Euclidean	0	0	0.9268	0
BBC	2	Euclidean	0.5000	1.0000	0.9744	0.6667
Netflix	9	Euclidean	0.5000	0.7500	0.7143	0.6000
Outlook	3	Manhattan	0.5556	0.8333	0.7059	0.6667
WhatsApp	7	Euclidean	0.1429	0.5000	0.4167	0.2222
WordPress	15	Manhattan	0.6667	1.0000	0.9412	0.8000
YouTube	4	Chebyshev	0.4000	1.0000	0.9464	0.5714
Average			0.3457	0.6354	0.7907	0.4409

Table B.38 Results of oversample synthesis task in dataset-2

Page	K Value	Distance	Recall	Precision	Accuracy	F1
Adobe	25	Manhattan	1.0000	0.3333	0.4000	0.5000
Amazon	6	Euclidean	0.6667	0.4000	0.9024	0.5000
BBC	11	Euclidean	1.0000	0.5000	0.9487	0.6667
Netflix	17	Manhattan	1.0000	0.6667	0.7857	0.8000
Outlook	2	Chebyshev	0.7778	1.0000	0.8824	0.8750
Whatsapp	37	Euclidean	1.0000	0.7000	0.7500	0.8235
WordPress	2	Euclidean	1.0000	0.4286	0.7647	0.6000
YouTube	4	Chebyshev	0.6000	0.2500	0.8036	0.3529
Average			0.8806	0.5348	0.7797	0.6398

Table B.39 Results of SMOTE synthesis task in dataset-2

Page	K Value	Distance	Recall	Precision	Accuracy	F1
Adobe	25	Manhattan	1.0000	0.3333	0.4000	0.5000
Amazon	2	minkowski	0.6667	0.3333	0.3333	0.4444
BBC	2	Euclidean	1.0000	0.6667	0.9744	0.8000
Netflix	3,6,7	Chebyshev	1.0000	0.6667	0.7857	0.8000
Outlook	31	Euclidean	0.7778	1.0000	0.8824	0.8750
WhatsApp	47	Chebyshev	1.0000	0.7000	0.7500	0.8235
WordPress	3	Euclidean	1.0000	0.4286	0.7647	0.6000
YouTube	2	Manhattan	0.6000	0.3000	0.8393	0.4000
Average			0.8806	0.5536	0.7162	0.6554

B.4.2 SVM Results

Table B.40 Results of SVM synthesis task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0.6933	0.3487	0.5050	0.4069
Amazon	0.4733	0.1959	0.8441	0.2672
BBC	0.8300	0.6379	0.9346	0.6526
Netflix	0.2833	0.2433	0.4943	0.2304
Outlook	0.2444	0.6962	0.5976	0.3552
WhatsApp	0.6129	0.4139	0.5067	0.4887
WordPress	0.6900	0.2959	0.7235	0.4025
YouTube	0.3540	0.1882	0.8432	0.1982
Average	0.5227	0.3775	0.6811	0.3752

B.4.3 Gaussian Naive Bayes Results

Table B.41 Results of GaussianNB synthesis task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0.0000	0.0000	0.6000	0.0000
Amazon	0.6667	0.3333	0.8780	0.4444
BBC	1.0000	0.6667	0.9744	0.8000
Netflix	0.0000	0.0000	0.5714	0.0000
Outlook	0.3333	1.0000	0.6471	0.5000
WhatsApp	0.7143	0.8333	0.7500	0.7692
WordPress	1.0000	0.5000	0.8235	0.6667
YouTube	0.0000	0.0000	0.8750	0.0000
Average	0.4643	0.4167	0.7649	0.3975

B.4.4 Multinomial Naive Bayes Results

Table B.42 Results of MultinomialNB synthesis task in dataset-2

Page	Recall	Precision	Accuracy	F1
Adobe	1.0000	0.3333	0.4000	0.5000
Amazon	0.6667	0.2222	0.8049	0.3333
BBC	1.0000	0.5000	0.9487	0.6667
Netflix	0.1667	0.5000	0.5714	0.2500
Outlook	0.3333	1.0000	0.6471	0.5000
WhatsApp	0.5714	0.4444	0.3333	0.5000
WordPress	0.6667	0.2857	0.6471	0.4000
YouTube	0.4000	0.5000	0.9107	0.4444
Average	0.6006	0.4732	0.6579	0.4493

B.4.5 Complement Naive Bayes Results

Table B.43 Results of ComplementNB synthesis task in dataset-2

Page	Recall	Precision	Accuracy	F1
Adobe	1.0000	0.3333	0.4000	0.5000
Amazon	0.6667	0.2222	0.8049	0.3333
BBC	1.0000	0.5000	0.9487	0.6667
Netflix	0.1667	0.5000	0.5714	0.2500
Outlook	0.3333	1.0000	0.6471	0.5000
WhatsApp	0.5714	0.4444	0.3333	0.5000
WordPress	0.6667	0.2857	0.6471	0.4000
YouTube	0.4000	0.5000	0.9107	0.4444
Average	0.6006	0.4732	0.6579	0.4493

B.4.6 Bernoulli Naive Bayes Results

Table B.44 Results of BernoulliNB synthesis task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0.0000	0.0000	0.4000	0.0000
Amazon	0.6667	0.5000	0.9268	0.5714
BBC	1.0000	0.4000	0.9231	0.5714
Netflix	0.0000	0.0000	0.5714	0.0000
Outlook	0.0000	0.0000	0.4706	0.0000
WhatsAapp	0.0000	0.0000	0.3333	0.0000
WordPress	1.0000	0.6000	0.8824	0.7500
YouTube	0.8000	0.2353	0.7500	0.3636
Average	0.4333	0.2169	0.6572	0.2821

B.4.7 Random Forest Result

Table B.45 Results of Random Forest synthesis task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0.0000	0.0000	0.6850	0.0000
Amazon	0.1933	0.1230	0.8690	0.1486
BBC	0.1750	0.1670	0.9277	0.1669
Netflix	0.0000	0.0000	0.5714	0.0000
Outlook	0.0000	0.0000	0.4706	0.0000
WhatsApp	0.0014	0.0050	0.3383	0.0022
WordPress	0.1733	0.1307	0.7100	0.1454
YouTube	0.3040	0.1735	0.8389	0.2156
Average	0.1059	0.0749	0.6764	0.0848

B.4.8 Decision Tree Results

Table B.46 Results of Decision Tree synthesis task in dataset-2

Page	Recall	Precision	Accuracy	F1
Adobe	0.1267	0.0817	0.4390	0.0964
Amazon	0.3267	0.0836	0.7383	0.1330
BBC	0.8500	0.2169	0.8341	0.3435
Netflix	0.0000	0.0000	0.5714	0.0000
Outlook	0.2222	1.0000	0.5882	0.3636
WhatsApp	0.6429	0.5605	0.5150	0.5952
WordPress	0.4800	0.2309	0.6200	0.3108
YouTube	0.0000	0.0000	0.7275	0.0000
Average	0.3311	0.2717	0.6292	0.2303

B.4.9 Logistic Regression Results

Table B.47 Results of Logistic Regression synthesis task in dataset-2

Page	Recall	Precision	Accuracy	F1
Adobe	0.3333	0.2500	0.5000	0.2857
Amazon	0.6667	0.2222	0.8049	0.3333
BBC	1.0000	0.4000	0.9231	0.5714
Netflix	0.0000	0.0000	0.5714	0.0000
Outlook	0.3333	1.0000	0.6471	0.5000
WhatsApp	0.7143	0.6250	0.5833	0.6667
WordPress	1.0000	0.5000	0.8235	0.6667
YouTube	0.0000	0.0000	0.9107	0.0000
Average	0.5060	0.3747	0.7205	0.3780

B.4.10 Baseline Results

Table B.48 Results of baseline synthesis task in dataset-2

Page	**Recall**	**Precision**	**Accuracy**	**F1**
Adobe	0.3333	0.2000	0.4000	0.2500
Amazon	0.3333	0.0526	0.5121	0.0909
BBC	1.000	0.1053	0.5641	0.1905
Netflix	0.5000	0.375	0.4286	0.4286
Outlook	0.5556	0.5000	0.4706	0.5263
WhatsApp	0.4286	0.5000	0.4167	0.4615
WordPress	0.3333	0.1000	0.3529	0.1538
YouTube	0.8	0.1379	0.5357	0.2343
Average	0.5355	0.2464	0.4601	0.2920

TEZ İZİN FORMU / THESIS PERMISSION FORM

PROGRAM / PROGRAM

Sürdürülebilir Çevre ve Enerji Sistemleri / Sustainable Environment and Energy Systems ☐

Siyaset Bilimi ve Uluslararası İlişkiler / Political Science and International Relations ☐

İngilizce Öğretmenliği / English Language Teaching ☐

Elektrik Elektronik Mühendisliği / Electrical and Electronics Engineering ☐

Bilgisayar Mühendisliği / Computer Engineering [✓]

Makina Mühendisliği / Mechanical Engineering ☐

YAZARIN / AUTHOR

Soyadı / Surname : Shekh Khalil

Adı / Name : Naziha

Programı / Program : Computer Engineering

TEZİN ADI / TITLE OF THE THESIS (**İngilizce** / English) : Predicting Trending Elements on Web Pages Using Eye-Tracking Data

TEZİN TÜRÜ / DEGREE: **Yüksek Lisans** / Master [✓] **Doktora** / PhD ☐

1. **Tezin tamamı dünya çapında erişime açılacaktır. /** Release the entire work immediately for access worldwide. [✓]

2. **Tez <u>iki yıl</u> süreyle erişime kapalı olacaktır.** / Secure the entire work for patent and/or proprietary purposes for a period of **two years**. * ☐

3. **Tez altı ay süreyle erişime kapalı olacaktır.** / Secure the entire work for period of **six months**. * ☐

Yazarın imzası / Author Signature .... **Tarih** / Date 31/08/2022

Tez Danışmanı / Thesis Advisor Full Name: Assoc. Prof. Dr. Yeliz Yeşilada

Tez Danışmanı İmzası / Thesis Advisor Signature: ....

Eş Danışmanı / Co-Advisor Full Name: Dr. Şükrü Eraslan

Eş Danışmanı İmzası / Co-Advisor Signature: ....

Program Koordinatörü / Program Coordinator Full Name: Assoc. Prof. Dr. Enver Ever

Program Koordinatörü İmzası / Program Coordinator Signature: ....