



**University of
Nottingham**

UK | CHINA | MALAYSIA

Visual Analytics of User Search History Using VizHiz

Submitted May 2022, in partial fulfillment of
the conditions for the award of the degree **MSc Data Science**.

Shruti Dudharkar
20387235

Supervised by Kai Xu

School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the
text:

Signature Shruti Dudharkar

Date 23 / 09 / 2022

I hereby declare that I have all necessary rights and consents to publicly distribute this
dissertation via the University of Nottingham's e-dissertation archive.

Public access to this dissertation is restricted until: DD/MM/YYYY

Abstract

People are said to engage in the process of analytics when they gather, organise, and generate representations of knowledge with the goal of solving a problem in which they are interested in gaining an understanding. When trying to solve difficult problems utilising large datasets over extended periods of time requiring research and analysis, people often get disoriented. It's possible that they forget what they've already accomplished, are clueless about where they are in relation to the larger goal at hand, and are unclear how to proceed. Within the framework of browser-based online analysis, I will present a tool called VizHiz here in this project. Its purpose is to solve the a forementioned challenges. I used six participants in a user research that was semi-structured to investigate the participants' actions in online search history analysis using the capability of current browsers. You may browse all of the websites that you've visited in the past using Google Chrome's built-in history manager, as well as revisit URLs that you didn't have time to save in your bookmarks.

This feature is standard on all versions of Google Chrome. However, have you ever considered doing an analysis of that data to assist you in determining which websites you go to most often during the weekdays or the number of times that you searched for a certain keyword? VizHiz is a python script that can do an in-depth analysis of your complete browsing history and present the findings in a variety of charts and statistics. Here I have used an extension to gather user's browser data for the recorded period and using python script visual analysis will be done. VizHiz delivers all of the interactive charts and statistical capabilities of the original History Trends. The data that are important to the work will be gathered from users in an iterative manner, then analysed with regard to certain limitations, and lastly the results will be communicated to others. I did a user research in a realistic work situation with six participants who used the extension to obtain most visited sites that comprises unique aspects of the sites and how many times visited by them. The purpose of this study was to analyse visually, the trends between user search history using VizHiz . I was able to get a visual representation for the browsing history of each and every participant. This functionality might assist large organisations control redirection on a certain website as well as server load by providing information about the number of users in a particular time scale.

Acknowledgements

Firstly, I would like to thank my supervisor, Dr. Kai Xu, for all of his assistance, recommendations, and direction throughout the creation of this project. This project would not have been feasible without his expertise and cooperation.

I am also thankful to all the professors who taught me throughout the taught phase, since they provided me with the needed academic information for writing my dissertation.

I am grateful to everyone who participated the workshop despite the fact that it required them to rearrange their calendars. The information that was required for the outcomes could not have been obtained without their contributions.

Lastly, I would want to thank my family, friends, and colleagues for their encouragement and support during my studies and the creation of this project.

Contents

Abstract	i
Acknowledgements	iii
1 Introduction	1
1.1 Motivation	2
1.2 Workshop	3
1.2.1 The Accumulation of Data	3
1.2.2 Dataset	3
1.3 Aims and Objectives	4
2 Background and Related Work	6
2.1 A Resource Utilization Analytics Platform Using Grafana and Telegraph for Savio Supercluster	6
2.1.1 Implementation	8
2.1.2 Analysis	9
2.1.3 Conclusion	10
2.2 Model of Computer Architecture for Online Social Networks Flexible Data Analysis	10
2.2.1 Data Collecting	12
2.2.2 Analytics tool	14
2.2.3 Twitter data Analysis Platform	15
2.2.4 Implementation	16
2.2.5 Conclusion	18

2.3	Interactive Visualisation techniques for the web data	19
2.3.1	Proposed Approach	20
2.3.2	S Paths	20
2.3.3	Linked Dataset	22
2.3.4	Results	23
2.4	Next Step Suggestions for Modern Interactive Data Analysis Platforms . .	24
2.4.1	Dataset	26
2.4.2	Analysis Tree	27
2.4.3	Single Parameter generalisation	27
2.4.4	System Workflow	28
2.4.5	Results and Evaluation	29
2.5	Visual exploration of machine learning results using data cube analysis . .	30
2.5.1	Visual Exploration	31
2.5.2	Implementation	31
2.5.3	Data Cubes	32
2.5.4	Leveragig data Pipeline	33
2.6	SenseMap: Supporting Browser-based Online Sensemaking through Ana- lytic Provenance	33
2.6.1	Capturing Data	35
2.6.2	Visualisation	35
2.6.3	Design	36
2.6.4	Conclusion	40
3	Design	42
3.1	Collection of Browser History using History Collector	42
3.2	Jupyter Notebook	43
3.3	Jupyter Notebook-Cell Dependency Network	44
4	Implementation	46
4.1	Data Collection using chrome extension-History Collector	46

4.1.1	Installing Extension	46
4.1.2	Getting Data	47
4.2	Data Analysis	47
4.2.1	Python Packages	48
4.2.2	Analysing browser history	49
5	Evaluation	54
5.1	User Evaluation-Confidentiality and Reassuring Remarks	54
5.2	Browser History Analysis Evaluation	55
6	Conclusion and Future Work	57
6.1	Results and Conclusion	57
6.2	Options for Additional Strides Towards Improvement	58
	Bibliography	58

List of Tables

List of Figures

2.1	Data flow from Telegraf,Slurm to nodes and stored InfluxDB or PostgreSQL	7
2.2	Architechture of Twitter Data Analysis	12
2.3	S-path Process	22
2.4	REACT system Architechture	28
2.5	General Machine Learning Pipeline	32
2.6	Analysis by ML Cube	32
2.7	Design Architechture	37
2.8	Sensemaking Model	38
2.9	One Highlight and One note	39
4.1	User history files	47
4.2	Chrome Extension Output Screen	48
4.3	Top ten accessed websites by user	49
4.4	Unique ten most accessed websites by user	50
4.5	Analysis of domain count per day	51
4.6	Analysis of domain count per month	51
4.7	Average Loading Time for Websites	52
4.8	Comparing user 1 and user 2 domain search	53

Chapter 1

Introduction

Data analysis is a process that requires novice researchers to learn as they go, and it is also a process that requires experienced researchers to negotiate and adapt to the study they are undertaking and the data they acquire. Utilizing technology that is capable of effectively analysing vast volumes of data is essential to the process of data analysis. In addition, in order to make sound decisions, both individual intellectual capabilities and group knowledge are essential components of analytical work [17]. In order to continually gain new insights, it is necessary for individuals possessing a variety of specialised knowledge to work together in order to achieve a detailed understanding of the associated outcomes. Because of this, the technology that is used for data analysis has to be able to support both technological innovation and the productive participation of a large number of humans and artificial intelligence systems in the study of massive amounts of data[18]. The purpose of this study is to demonstrate how researchers can employ visual analytics to diagnose and explain occurrences in everyday contexts, as well as how it can be added to their toolbox as a method of interpretation and analysis. Visual analytics can be used to diagnose and explain occurrences in everyday contexts. This study addresses the use of visual analytics as a tool for diagnosis of the process that takes place with the user's online search history, how many sites user has visited the most and gives the stat of browsing history by day, month, year etc. It is proven that adopting visual analytics as a diagnostic tool in qualitative analysis and interpretation may be beneficial, and evidence to support this claim is offered.

A user study was conducted for data analysis were used to influence the design of generic scale modelling. This was done as a manner of defining the demands of such an analysis and placing the roles that technology and people play in this symbiotic interaction [44]. In order to evaluate the practicability of the proposed framework for the production of a data analysis enabling platform, two real-world use case studies that contrast with one another were carried out. Because of this result, our awareness of the complexity of individual and team models for data analytics has increased, as has our excitement about their potential.

1.1 Motivation

The built-in history manager of Google Chrome allows users to explore all of the websites that they have visited in the past. Users may even revisit URLs that they did not have time to store in their favourites. This functionality is included by default in each and every version of Google Chrome. On the other hand, as part of this project, I've been thinking of doing an analysis of those data in order to provide you with assistance in establishing which websites visitors visit the most frequently on weekdays or throughout a certain month. The trend in the industry is increasingly shifting away from one-sided service from the present service provider and toward the automated execution of adequate tasks that are acceptable for user choice as the internet trends continue to increase in popularity [17]. Recent years have seen a rise in the amount of research and services that examine the lifestyle patterns of the user, provide tailor-made information, or advertise in an effort to keep up with the rapid speed of this change. In most cases, the analysis of the user's search history was contingent upon the activities that were now taking place in the user's life[17]. In addition to being able to see all of the history from the last four months on a single page, we are also able to identify individual browsing tendencies thanks to the clear and concise visual display of your life online.

JupyterLab gives you the ability to deal with documents and activities in a way that is flexible, integrated, and extendable. Some examples of these are python notebooks, , terminals, and customized components. JupyterLab provides users with a standardised paradigm for viewing and working with various data types. JupyterLab is capable of

displaying rich kernel output in many different file forms, including pictures, CSV, JSON, Markdown, and among many others, and recognises a wide variety of file formats [44]. One of the most popular tools for developing algorithms and the accompanying models is called Jupyter Lab. The primary objective of Jupyter lab is to generate Python code, then compile that code, and then provide output depending on a set of callbacks, variable modifications, and library usages that have been provided. Any kind of data may be produced as an output from the code [41].

1.2 Workshop

I led a workshop with six students from Nottingham in order to obtain a better grasp of what search history data I might use for visual analysis in a Jupyter lab. This was done in order to get a better understanding. The following list provides the activities to be completed as well as the results of the workshop.

1.2.1 The Accumulation of Data

The intention is to compile the users' internet search histories for further visual examination. An extension for Google Chrome was made available to the consumers. This Chrome addon essentially gathers together two different items. To begin, it provides users with their whole search history of four months [e.g. user1-full.csv], which includes the URLs and domain names of websites, as well as a time stamp underneath it. Second, it provides a list of the user's most frequently visited URLs [e.g. user1-mv.csv] together with the number of times each site was browsed.

1.2.2 Dataset

Every user, after reading the instructions, downloaded and installed the plugin and utilised it. After using the plugin, they obtained two CSV files that included the data for their most frequented sites as well as their complete search history. This search history for the previous four months is collected by a Chrome plugin. Following the session, I had access

to the whole search history data for all six users, which brings the total number of rows in the dataset to 48827. These rows were utilised in the visual analysis that I performed.

1.3 Aims and Objectives

The requirements that were gathered from the participants served as a significant source of motivation for me to develop the different features. The requirement of this project is to analyse the collected browser history data and obtain some insights that could eventually assist large organisations in knowing a particular website's redirects and managing server load for a specific domain during a specific time period. Now that I have more than enough data to begin analysis, the requirement of this project is to analyse the collected browser history data. When taking into account the root of the problem, there may be a great number of facets that may be included into the investigation via the modification of various factors. Because I have a much extensive need list and have a very limited amount of time to finish the project from the ground up. The evaluation of the requirements and the development of a feature that satisfies them will consume a significant amount of time. As a result, I have made the decision to take a few criteria into consideration and include them into VizHiz.

The following is a list of the primary characteristics that it was my intention to create:

- Display the user's most frequented websites and domains visually
 - This should include the most popular destinations.
 - Should take into account and display the facts for each user on an individual basis
 - Show the websites that were visited the most by all users combined
 - The page in a domain that receives the most visitors.
- Display the findings for a particular domain
 - divide the data into daily and monthly intervals using the time series graph
 - Illustrate how accessibility to websites is changing for all people

- The total amount of time that was spent to load the webpage
- patterns or routines of browsing that are characteristic of each person and how they compare to or vary from one another

Chapter 2

Background and Related Work

2.1 A Resource Utilization Analytics Platform Using Grafana and Telegraf for Savio Supercluster

[7] Clusters for high performance computing (HPC) are already very complicated systems, and their complexity is only going to increase over time. Understanding how the cluster is really being used, as well as making strategic and planning choices, gets more complex over time as users need new kinds of computing resources, increased storage and networking capabilities, and the usage of new applications. It is necessary for the data on the status and usage of the cluster to be clearly accessible and presented in a manner that is appropriate for the use-cases of users, system administrators, and managers in order to successfully make choices within this complicated environment. In general, the users of this project should be able to make better choices on how to make better use of the cluster resources as a result of the data that is gathered and provided by this project. The administrators of the system also need to be notified of potentially problematic circumstances, such as when the storage space is close to reaching its maximum capacity or when there is an abnormally high level of CPU activity on the login nodes. The metrics that are accessible may be broken down into four primary categories: hardware status, compute usage, storage utilisation, and network use. Initially, The team concentrated on compute usage, more specifically the collection of task data from the Slurm scheduler,

the acquisition of additional data from other sources, and the combination of these three types of data to build acceptable visualisations.

The system demonstrate the approach that was utilised by Berkeley Research Computing to combine existing data[12], obtain new usage statistics, and show the necessary charts for various use-cases on the Savio supercluster. In the article, data were gathered, integrated, and presented according to task information obtained from slurm. This included account association information as well as CPU metrics obtained using Telegraf. With this information, the Grafana visualisation framework is able to provide broad summary statistics, such as aggregated consumption by campus department, all the way down to the CPU utilisation on a single node throughout the length of a task, and all of this with flexibility for a wide range of different purposes and use-cases. Metabase is also being considered for the visualisation role in this project; however, at the time that this project was being developed, Metabase did not support InfluxDB, which is the high-performance time series database that Telegraf prefers, and it does not place a primary emphasis on time series data. It does have the ability to create public dashboards; however, these dashboards lack the interactive capabilities of Grafana, which enable users to pick different time frames.

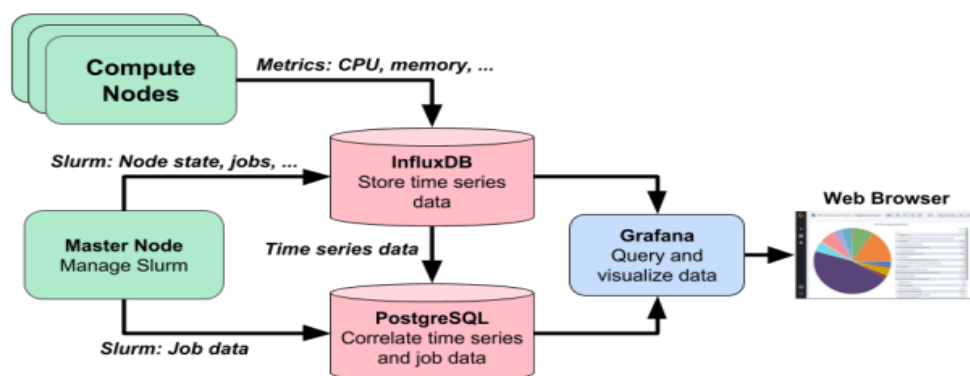


Figure 2.1: Data flow from Telegraf,Slurm to nodes and stored InfluxDB or PostgreSQL

2.1.1 Implementation

An instance of Telegraf was installed on each computing node and login node in the cluster so that metrics such as CPU use, memory consumption, and disc usage could be gathered about each individual node.[12] It would seem that Telegraf's resource use on the compute nodes is low. The amount of CPU time that it uses is comparable to that used by the Slurm daemon.[32] The data collected by Telegraf, which is executing on the compute nodes, is stored in an InfluxDB time series database which can be accessed by using Grafana. It is not difficult to install additional Telegraf input plugins that operate at a variety of time intervals. PostgreSQL was used for this function because it is a relational database, which was necessary for performing more complicated queries like those that correlate CPU statistics with data taken from a task. PostgreSQL ingests CPU data from Telegraf and gathers Slurm job data for the previous day using a script that runs everyday in a crontab.[22] Both of these processes take place every day. The majority of the data came from Slurm,[32] and it was processed by a combination of scripts written in Bash and Python. It was necessary to gather and visualise the current condition of the nodes and queues in order to deliver the required information.

a. Collecting Node

It states The current node states may be seen by using the `sinfo`[32] command on the Slurm[32] command line interface. After gathering the node statuses and partition allocation data, the `main.sh` Bash script, which is invoked at periodic intervals, formats the information in accordance with the InfluxDB Line Protocol. This information is then ingested by Telegraf[7] and stored by InfluxDB. allocation makes it possible to show statistics on a partition over time, while node state makes it possible to monitor the state of each individual node.

b. Visualising the Node State

A bar chart displaying the number of nodes that are now idle is the quickest and easiest method to get a sense of the condition of the nodes. Users are able to choose between

divisions and estimate waiting lists thanks to this feature.

c. Collecting Queue State

Using the `squeue` command on the Slurm command line interface, it is currently possible to quickly get information on the current status of the Slurm queue. This information on the queue and the tasks that are presently being processed is gathered by the `main.sh` Bash script, which is invoked at regular intervals.[22] It contains the information required for filtering the queue, such as the work ID, the job status, and the kind of resources that were requested by the user. Another script gets the raw data from the queue and organises it into groups according to account and partition type. The information is then prepared in accordance with the InfluxDB Line Standard before being ingested by Telegraf and saved in InfluxDB[13, 12].

d Visualising the Queue

By stipulating that the node's reported status must be `PENDING`, a simple InfluxDB query may restrict the data from the queue to just the nodes that are now part of the queue. A historical perspective of the number of requested nodes in the queue may be obtained by first grouping the nodes by partition and then summing over the relevant time span.

2.1.2 Analysis

Slurm keeps a record of every piece of work information for its own accounting needs; this record may be accessed with the `sacct` command.[13] Because processing Slurm data might take a few minutes at a time, the job information is updated to include the department name and is executed at more frequent intervals (such as once per day). [22]A script known as `lookup.sh` is responsible for determining the department name. This script acts as an abstraction for the purpose of seeking up the department title based on the online handle that is provided by Slurm. If `lookup.sh` were to be placed on a separate cluster, the appropriate adjustments would need to be made. The following information is included

in the data that InfluxDB[25] gets as tags: job ID, username and password, account type, department, CPUs sought, partition, and final task status. This information is structured according to the InfluxDB Line Protocol (such as CANCELLED or COMPLETED).[36] A numerical field is used to hold the information on the number of requested nodes, the number of allocated nodes, the raw time spent on the work, and the CPU time spent on the job. When determining which partitions would be the quickest to utilise when submitting test tasks, the display of the availability of the partitions proved to be quite helpful. It has become easier to spot difficulties with nodes thanks to the visualisation of the node status over time. The management is able to have a broad understanding of how the clusters is being used on a high level thanks to broad usage by cluster department.

2.1.3 Conclusion

When the task data from Slurm is presented, in particular when it is enhanced with other metrics and tagged with other information, numerous different sorts of users are afforded the opportunity to engage with the cluster in a manner that is more effective. Even though it's still in its early phases, this analytics tool has already shown to be valuable to users inside the business. The stack is made up of open source software, making it very adaptable. It may be installed in other businesses and customised to work with the infrastructure that they already have in place.

2.2 Model of Computer Architecture for Online Social Networks Flexible Data Analysis

The importance of OSNs, also known as online social networks, is steadily growing in today's culture. But at initially, academics saw these technologies as little more than a curiosity since they thought of them as being either irrelevant or fun. This perspective has shifted over time as a consequence of the development of the socio-technical environment, which has led to the current scenario in which investigations linked to OSNs have drastically risen over the last few years and include a wide variety of fields.[22] In point of fact,

the growing number of users combined with the democratisation of mobile technology has caused the data collected by OSNs to become an accurate reflection of society. Indeed, exchanges in text mode, images, or videos reveal not only the preferences, worries, and social, political, or economic leanings of people, but also the modes of communication that individuals prefer. This is especially evident in the case of Twitter, which generates a nearly instantaneous echo of all significant events taking place everywhere in the globe. Even if it is less organised, this reaction seems to be more intense than the one shown by the press. For instance, Paul S. Earle and his colleagues demonstrate, with the use of data from Twitter, that around 75percent of earthquake detections take place within two minutes of the event's originating time.

These examples illustrate why the potential of OSNs[25] began to mobilise academics and corporations, who have sought to harness the abundance of "big data" related to user interactions. These individuals are looking to capitalise on the potential of OSNs. The problem is one of both a scientific and a financial one. On the one hand, there is the matter of finances, since the data provided by OSNs pave the way for the improvement of already existing features at a cheaper cost. This is the situation with opinion polls, which, in an ideal world, might be conducted without face-to-face interviews, so offering a picture of collective sentiments in the quickest and most accurate way possible. On the other hand, these data are very important to the scientific community since they open the door to previously unimaginable ways of observing human behaviour.[11]The analysis of data relating to interpersonal acts of communication (such as tweets, Facebook status updates, short films on Vine, etc.) and the environment in which these interactions are carried out is the foundation for all of this potential (time, localization, popularity, etc.). The following example of generic architecture illustrates the aggregation of many fundamental elements, including the collection, storage, and analysis of tweets, as well as the management of the user interface.[14] This generic design not only allows for the combination of functions that complement one another, but it also makes it possible to emphasise the systemic qualities that result from the relationship. For instance, the system may exercise some degree of independence by gathering tweets that were not specifically re-

requested but do have a connection, either chronologically or semantically, [11] with the original request. If The team took into account that this kind of platform may be utilised by a group of analysts, then this might be considered a community connection.

Due to the need of maintaining this continuity, the platform must be the focus of particular care. Both the design and its capabilities need to be scaled up significantly. In addition, its functioning has to be monitored so that any issues that arise may be located and resolved as soon as possible so that the data collection process can continue.[36] The data collection from Twitter is the primary emphasis of this article. It explains the format of this data as well as the methods that the social network suggests using in order to get them. After that, it provides a list of technical tools for analysing tweets. Following this, it provides an overview of the current state of the art regarding systems for collecting and analysing data from Twitter. In order to emphasise the strengths and weaknesses of the databases, a summary of the kind of database that was utilised as well as the characteristics of their interface were provided. In later sections,[13] it delves into the topic of user interfaces in data analysis. Last but not least, it provides a case study of the platform that The team designed for the analysis of Twitter's data.

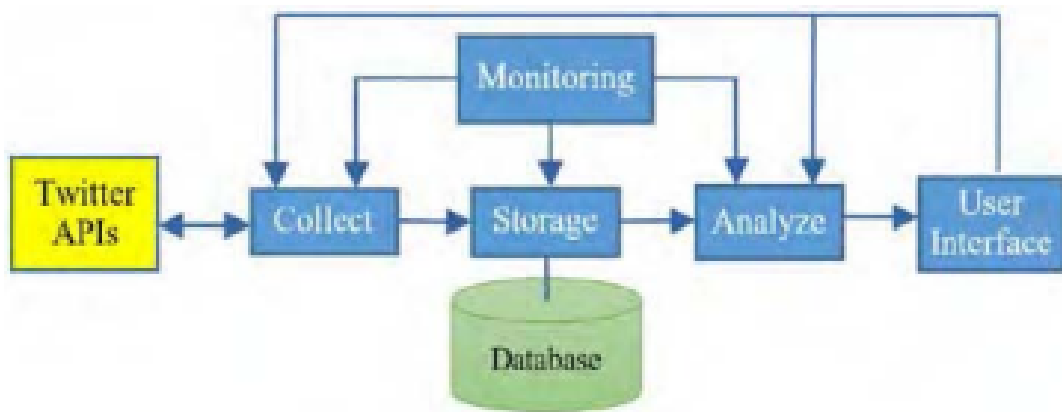


Figure 2.2: Architecture of Twitter Data Analysis

2.2.1 Data Collecting

Twitter is a free latency network that has been around since March 21, 2006. For the time being, users are only allowed to post messages (tweets) that are a maximum of

140 characters long; nevertheless, the firm often debates the validity of this artificial constraint. In point of fact, Twitter is planning to relax its strict adherence to the 140-character limit as soon as May 2016.[25, 4]This discussion is not objective since sending brief messages encourages a specific kind of usage that is characterised by spontaneity and straightforwardness. Tweets may be directed to a closed group of readers, but more often than not, [32]they are made available to the whole public. They are not possible to be edited; the only option is to delete them. Users have the ability to write them; they may retweet them (which means they are citing them) or add people to their list of favourites. Some scholars have attempted to provide explanations for these behaviours by looking at them from a social perspective. For example, the action of retweeting someone's post might be seen as agreeing with them, suggesting they read something, giving knowledge, being flattering, providing a picture of an event, providing payback, or increasing their visibility. [19, 36] It is possible to acquire and evaluate this data using a variety of different approaches. Since 2006, Twitter has made its Application Programming Interfaces (APIs) available to developers. According to Twitter's rules about them, only a small portion of users get access to all of the tweets at once (through the Firehose), while the other users only have restricted access (Public APIs). They only offer tweets that have been made available to the public. REST APIs, [14]which stand for "representational state transfer," and streaming APIs are the two types of application programming interfaces (APIs) [4, 14] that are available via the limited access. The REST APIs are often used for research that concentrate on entities like as hashtags, phrases, or keywords in tweets. On the other hand, the public Streaming APIs are typically used for studies that aim to analyse longitudinal trends in topics such as movies or politics. These APIs are free to use, but you will need a free Twitter account to use them. Because Twitter sets download rate restrictions, the data that may be accessed using the REST APIs are severely restricted. These limits are broken up into 15-minute periods. In a similar vein, Streaming APIs only provide for restricted access to the live stream of tweets, which amounts to less than 1percent of the overall flow. OAuth [4] is the protocol that Twitter employs in order to provide authorised access to its application programming interfaces (APIs). A link

between Twitter and an user is dynamically generated for each query using the client-server architecture, which is the foundation of the REST APIs. On the other hand, Streaming APIs require that a constant connection be maintained between Twitter and its users; moreover, they are intended for the transmission of significant amounts of data.

2.2.2 Analytics tool

In general, the gathered tweets need to be moulded and processed in order to unearth pieces of information that are buried in the signals and data, which may sometimes be rather faint. This is done in order to uncover patterns or warnings. Some of these aspects could already be covered by the data mining technologies that are now available. However, the format of tweets and the data that are linked with them (retweets, the author's id, and so on) have certain features that require a mostly linguistic pretreatment.[4] This is particularly significant if one desires to do an operative data analysis. In addition to the particular application, a number of developers have built plugins in order to modify the data mining tools already in use to work with the Twitter APIs. By way of illustration, the "Analytics module for Twitter" makes it possible to do Twitter queries from inside Microsoft Excel 2010. One is able to do analysis such as determining who the most active users are,[19] which tweets match to a certain hashtag, or which tweets are more positive or negative. Some writers make advantage of the reporting tools offered by Google Analytics in order to monitor the activities of online social networks, particularly Twitter. These methods are especially well-suited for use in marketing tactics that attempt, for example, to gauge the level of interest in a certain product, event, or television programme. It is more intriguing to undertake more complex analysis using specialist tools in statistical and data mining computing since these fields have developed significantly in recent years.[14] The majority of these solutions are equipped with a connection that can be used to establish a connection with the Twitter APIs.

a. Massive Online Analysis

It is a free software application that is specialised in data flow analytics and also enables the development of recommendation systems. Weka, a well-known data mining tool, is where MOA got its start (classification, etc.). These two tools combined give a tremendous deal of flexibility. The MOA Twitter[14] reader module makes it possible, in particular, to adapt these tools to the environment of Twitter. In addition to the gathering, it provides the ability to identify shifts in real time, like the identification of phrases whose frequency varied over time. In addition to this, it enables the examination of sensations in real time.

b. RapidMiner

It is one of the most well-known data analysis tools that has been around since 2006. There is currently a free version that can be found on sourceforge.net as well as a commercial version that costs 2,000dollars to purchase.[36] RapidMiner[14] studio has, more recently, provided functionality for evaluating activity on Twitter, as well as multilingual texts, emotion, and other such things .

TwitterR

The module for R is introduced for the very first time. R is open-source software that implements the computer language S for the purpose of data handling and statistical analysis.[14] R is available free of charge. Wrapper for high-level conversations with the Twitter APIs is what this module is all about. It makes the OAuth authorization process easier and converts requests written in S language into requests written in HTTP REST. Since February 23, 2014,[36] users of relational database management systems like as SQLite have had the ability to effortlessly capture tweets and other information.

2.2.3 Twitter data Analysis Platform

The overwhelming majority of the most recent science research on the topic uncovers that there is a plethora of solutions to claim back data from Twitter, and many papers make use of a relation database to record these solutions. This is the case for the vast

majority of the publications.[13, 25] This is the case despite the fact that there is currently a dearth of scientific research on the topic. Because different studies report their findings with varying degrees of clarity and precision, it is a difficult job to compare the many architectures that were addressed in the scientific literature. This makes the process more difficult. K. Makice's book "Twitter API: Up and Going - Learn how to create Apps with the Twitter API" was released into the public domain in March of 2009.[13] He describes how to store tweets in a MySQL database system as well as how to retrieve them using the programming language PHP and how to access the Twitter APIs.

A software architecture for the development of stochastic models to describe OSNs was presented by R. D.W. Perera, S. Anand, K. P. Subbalakshmi, and R. Chandramouli in the year 2010[19]. The concentrated on the lengths of time that pass between the posting of tweets and the frequency with which[4] one user retweets the tweets that were sent by another user. They accomplish this goal by using the Search API provided by Twitter REST APIs, as well as the programming languages Python and PHP, and a centralised MySQL database. Python is used for the scripting of the collection of tweets, and the Twython library is used.[25, 36, 14] The script that automates their captures runs every 5 minutes. They use a Yahoo web service that converts street addresses into GPS coordinates in order to ascertain the location of tweets. The MySQL database is used to hold tweets that have been captured by first obtaining the tweet's id, then its timestamp, and finally the author's id. The contents of the database are read by a PHP programme, which then displays those contents.

2.2.4 Implementation

In order to provide a software as a service (SaaS)[36] to professionals in the fields of science and public policy, the project team developed a partly distributed architecture that was built on the Twitter Streaming APIs. This platform makes it possible to conduct longitudinal investigations of a wide range of topics in almost real time. The user may choose whatever terms they are interested in looking for. The software compiles all of their requirements into a list of keywords to monitor, and then it provides that list as a

parameter to the streaming APIs. In addition to this, it covers a broad range of subject areas. As a result, it provides a great degree of adaptability.[36] Because the data have already been gathered, users often do not have to wait while data are collected before they can begin querying the service.

The data storing function is handled by Elasticsearch, which is a distributed search engine. It is open source and is built on the Apache Lucene database.[14] This particular technical option offers a number of benefits when it comes to the collection of tweets. To begin, as the tweets are being indexed, it tokenizes them, which opens the door to real-time and full-text search possibilities. This allows for the observation of trends in real time through the user interface. e. Second, Elasticsearch[13, 25] ensures that it always has an up-to-date duplicate of each shard that makes up the tweets index in order to avoid data loss that may be caused by hardware failures. Thirdly, when there is a rapid spike in the number of tweets that are sent by Twitter APIs, it is very necessary to have a system that can do inserts in a very short amount of time in order to prevent being disconnected. When it comes to the insertion of new data, Elasticsearch may be up to twice as quick as MySQL. In conclusion, tweets are JSON objects when they are delivered by Twitter APIs.[3] Elasticsearch's JSON document-oriented side enables it to index these tweets without the need for conversion. Additionally, plug-ins are supported by Elasticsearch.

This platform is used by policymakers and scientists via the utilisation of a web application. This web application is composed of a server component (back-end) that is run by Node.js and a client component (front-end) that is mostly composed of AngularJS and jQuery.[25] Here between Elasticsearch cluster and connections coming from the Internet, the server side of this application mainly serves as a security layer. End users have access to a variety of tools, such as a comparing tool with charts, word clouds, a buzz observatories, and so on, through the client side, also known as the front-end. The request is sent to the server side whenever the frontend has to load or update an AngularJS directive[13] (for example, a graph, a list of retweets, or anything else along those lines). After then, the server sends queries to the Elasticsearch cluster, and while it is doing so, it fulfils the requests of other users up to the point when it has no more work to do. Because callback

functions are used in such large quantities, there is no blocking process.

The following methodology was used by the project to conduct an analysis of the performance of our SaaS.[25] This is a tool for comparing the front end application with Google Chrome on a local network, so that any potential delays caused by the internet may be avoided. The controller of the comparison tool will change the settings of the page's directives whenever a user populates the tool's input field with an expression that includes both a start and an end date. These modifications cause a large number of queries to be run in order to obtain all of the necessary data. In addition to the tools that are supplied by the front-end, there is a tool known as the exporter tool that provides our users with further versatility. Indeed, it makes it possible for us to reuse the data that we have acquired. They are able to download the tweets in their original form, which has been augmented with the additional information that has been provided by various analysis services. This metadata may include the gender of the writers or the polarity of the tweets. It is compatible with the JSON and CSV file formats and comes with a control panel that allows you to customise the tweets that are exported.[3]

2.2.5 Conclusion

This article highlights the problems that must be overcome in order to realise a platform that can collect and analyse tweets, as well as some potential solutions to those issues. First things first, let us point out that such architecture has a direct connection to the structure of Twitter. Indeed, even a little adjustment to Twitter's application programming interfaces (APIs) necessitates a modification to the platform's data collection procedure; failing to do so might result in the whole system being inoperable, with results that are either incorrect or lacking entirely.[14] User interfaces play a primary role in the process of designing a data analysis platform for an OSN because they are deeply related to the possibilities offered by the OSN's application programming interfaces (APIs) and their limits, the requirements of its future users and administrators, and the potential fields in which data mining can be applied.[19] However, despite the fact that The team have a platform that is excessively large and that The team have taken many technologi-

cal decisions, The team have found that the subject of performance is still relevant since The team are gathering tweets on a daily basis and over an increasingly extended period of time. In point of fact, the number of tweets The team have determines the level of our performance.

2.3 Interactive Visualisation techniques for the web data

The RDF file format enables computers to do advanced tasks such as reasoning and federated querying across related information. However, displaying RDF data to people is a particularly tough task since the structure of RDF itself makes it impossible for users to make sense of the information because it breaks it up into little parts. This makes traditional methods of data presentation ineffective. In the beginning of this work, S-Paths, which is a system that supportsexploration based on predefined sets of a dataset's content, is described. The work being done on the project suggests a strategy that places data inside a context in order to make them more comprehensible to people. It has been shown that it functions correctly on relatively easy designs, but that[6]its usefulness is limited by problems with its performance on more conceptually complex models. Next, The team established the groundwork for a second project, the objective of which is to take another step back and place these group of features in a larger context so as to provide a structural summary of linked data. This project's goal is to go one more step back than the first. Experts involved in the publication as well as reuse of RDF data need to improve their knowledge of their data, and many of the sets of data published as RDF could also be of involvement to lay users. Although the RDF format was designed to be digested by mechanics, there is a powerful necessity for imaging and exploration tools. On the other hand, due to the nature of RDF's structure, it might be challenging to develop effective visualisations.[14, 19]

A node-link diagram is the first representation that comes to mind, and it is also the one that most properly conveys the underlying structure of a directed graph. Such diagrams

are particularly effective at describing the data model or a restricted number of entities; but, if they include more than a few dozens of triple statements, they become illegible, despite the fact that even very tiny datasets contain many thousands of such assertions. Another method, which is used by the majority of Linked Data browsers,[16] consists of showing one page for each resource and presenting any statements that are directly connected to it in the form of clickable links when they consist of URIs. Although this allows users to hop through one resource to the other across graphs and datasets, regardless of their size, the first disadvantage is that offering high - quality to describe a resource may be a few hops away: when the user finally reaches a piece of data after several clicks, this data is displayed separately from the resource that was initially of interest to the user. A large amount of memorising work is required in order to keep track of the sequence of preceding pages.

2.3.1 Proposed Approach

Given that readability is a necessary but not sufficient condition to enable sense-making activities and that The team preferred to follow longer chains of properties rather than receive crumbs of information, it is hypothesised that chains of triple statements in the graph can be used as aggregate steps to reach readable ranges of values. No matter how many entities are in the set under consideration, values that come from following sequences of triple statements should always be in a readable dimension or aggregateable.[21]

2.3.2 S Paths

Its foundational paradigm is a mixed-initiative strategy, which implies the system will suggest novel viewpoints for a given set of resources but leave the door open for manual reorganisation by the user. S-Paths[4] defaults to what it believes is the most relevant viewpoint on that option, which users may, again, modify at their whim, making exploration of a collection of resources a cyclical activity. This takes place each time a user makes a pick.

a. Visualisation Status

When two views share entities, S-Paths is able to animate the transition between the views in a way that is smooth and seamless.[25] This helps to reduce the amount of mental effort required to make the connection between the present vision and the one that came before it by providing a fundamental degree of visual continuity. The system also allows for the juxtaposition of two successive views, as well as the brushing and connection between those views. When users pick items in one view, the system instantly highlights the matching elements in the other view, which further assists users in relating views. When it comes to brushing and connection between aggregates, the same space-filling method as described before is used. Additionally, S-Paths remembers all previous views and displays them in the interface in the form of dots arranged in a simple timeline, which can be found in the lower-left corner of the screen. Users have the ability to simply backtrack by clicking on one of these dots, which takes them to the view that corresponds to that dot. The team relied on a set of views because the conditions of readability are not absolute and instead depend on the type of visualisation being used. Each view declared how many dimensions it can display,[16] which categories it can display for each dimension, and under what conditions it can display those categories (minimum, maximum and optimal). The viewpoints are not mutually exclusive; multiple of them may be applicable to a certain situation. In this particular instance, the optimum circumstances will be utilised to offer the view that is the most effective.

b. Algorithms Used

When two views share entities, S-Paths is able to animate the transition between the views in a way that is smooth and seamless. This helps to reduce the amount of mental effort required to make the connection between the present vision and the one that came before it by providing a fundamental degree of visual continuity[38]. The system also allows for the juxtaposition of two successive views, as well as the brushing and connection between those views. When users pick items in one view, the system instantly highlights the matching elements in the other view, which further assists users in relating views. When

it comes to brushing and connection between aggregates, the same space-filling method as described before is used. Additionally, S-Paths remembers all previous views and displays them in the interface in the form of dots arranged in a simple timeline, which can be found in the lower-left corner of the screen. Users have the ability to simply backtrack by clicking on one of these dots, which takes them to the view that corresponds to that dot.[4]

The team relied on a set of views because the conditions of readability are not absolute and instead depend on the type of visualisation being used. Each view declared how many dimensions it can display, which categories it can display for each dimension, and under what conditions it can display those categories (minimum, maximum and optimal).[6] The viewpoints are not mutually exclusive; multiple of them may be applicable to a certain situation. In this particular instance, the optimum circumstances will be utilised to offer the view that is the most effective.

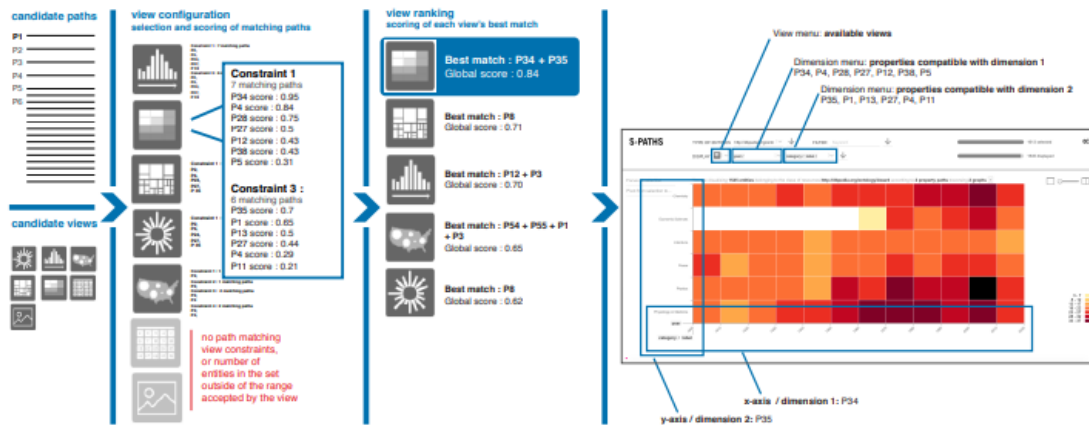


Figure 2.3: S-path Process

2.3.3 Linked Dataset

A linked data publisher, a linked data reuser, and a lay user were the three persona roles that the team decided to develop. Throughout the process of developing roofs of concept, these three persona characters served as a guiding thread. Their purpose was to ensure that key user duties and concerns were not forgotten. The project built high-fidelity prototypes in order to provide proofs of concept for the technique and to assess those prototypes. In order to identify and characterise groups of things and the accompanying

semantic routes, an application programming interface (API) was first developed for S-paths. After then, a simplified[21]version of the visualisation system consisting of two view components was constructed. This was done in order to establish the general query and transition mechanism, after which further views were gradually added. The Nobel Prize Linked Dataset was the one that was used for the development process. This dataset contains 85 797 triple statements that represent 15 different categories of resources. In order to illustrate that the tool is capable of working across many networks, the team constructed a tiny graph that was taken from DBpedia. This graph includes photos and geolocations[19, 4] for entities that are referenced in the Nobel Dataset. The Nobel Dataset has 2,234 triple statements.

2.3.4 Results

Users who were not familiar with graph databases or the concept of a route in a graph were nonetheless able to utilise the application successfully. They anticipated the interface to be more responsive and complained about not obtaining previews and rollovers for options; yet, the constraints that were caused by Linked Data made navigating difficult for them. On the other hand, the specialists had little trouble understanding that these limits are the result of the structure and scattered nature of the data, and they voiced their excitement about the potential of obtaining overviews. Twenty minutes seemed like a very small amount of time to them considering how intently they were all focusing on the navigation. They would have benefited from having extra time to hone their skills with the instrument and to investigate the data. S-Pathways,[6] following the paths, and being able to associate geolocations from DBpedia with places across the two graphs made this process very easy to understand and carry out.

S-Paths would also benefit from the discovery of an effective method for detecting linkages between datasets,[16]incesince this would make it possible to transform the programme into a comprehensive browser for linked data. Exploration of many graphs that are housed behind a single endpoint is all that can be done at this time due to technical constraints. The capability of pivoting, which is currently included in the tool, might

be paired with paths that traverse databases to create a new feature. Making changes in the attention from a group of entities to other with ongoing way is what is meant by the term "pivoting." [21] One has the option of deciding whether or not to maintain the limitations of the current subset. This would make it possible to move quickly between different datasets. This would indicate that a standardised method of communicating the analysis that was conducted at the endpoint has to be discovered.

2.4 Next Step Suggestions for Modern Interactive Data Analysis Platforms

Interactive Data Analysis tool, such as Kibana, Splunk, and Tableau, are slowly replacing OLAP and SQL.[38] This is happening because IDA platforms allow for simple data exploration, visualisation, and mining, and they do so even for users who are not proficient in SQL or programming. Nevertheless, analysis of data is still a challenging undertaking, particularly for users who are not experts in the field. In light of this, the article introduces REACT, a peer support developed specifically for [28] contemporary IDA platforms. Analysis sessions on these platforms blend together high-level activities of many kinds and carry out their operations across a wide variety of datasets. REACT is able to recognise and generalise pertinent (past) sessions in order to provide the user tailored options for their future course of action.[2]

It does this by employing a generic tree-based model to reflect the user's analytical context. In this model, the edges indicate the user's most recent actions, and the nodes reflect the "screens" that resulted from those activities. In order to index and retrieve potentially useful candidate next-actions quickly and effectively, a specialised context-similarity measure is used.[1] Following this, they are extended into abstract actions that express common elements, which are then tailored to the particular user scenario. In order to demonstrate the usefulness of REACT, the team carried out a comprehensive experimental assessment both online and offline, using real world analytical logs from the cyber security [21] field as the subject matter. Data analysis is, at its core, an interactive

and iterative process. During this process, a user issues an analysis action (also known as a query), gets a results set, and then determines whether or not to issue another analysis action and, if so, which one. Up until very recently, doing analytical jobs required an in-depth knowledge of not just SQL and programming but also mathematics and statistics. On the other hand, ever since the beginning of the age of big data, the infrastructures and support for interactive data analysis (IDA) have significantly advanced: Tableau, Kibana (ELK), and Splunk are examples of new[2], frequently web-based platforms that are gradually replacing traditional tools. These platforms make data exploration, visualisation, and mining simple to use for users of all skill levels, including those who are unfamiliar with SQL and programming languages. However, IDA is still a challenging procedure, particularly for users with little prior knowledge, since it requires an in-depth comprehension of both the examined domain and the specific environment. Users have the possibility of skipping crucial analytic steps and missing essential features of the data as a result.

In the information retrieval approach, systems generate recommendations based on the following presumption: if users are trying to pose similar sequences of queries, it is likely that they are interested in the same section of the dataset. This is accomplished by utilising a repository of (previous) queries that were submitted by the same or other users.[8] As a result, the questions asked by one user may be used to inform the suggestions given to another. The data-driven method involves systems doing an analysis of the available data and making suggestions depending on the degree to which the query result is likely to be interesting.

IDA platforms simplify the process of conducting composite analyses by allowing for the interweaving of activities of various sorts (such as SQL-like operators, OLAP multiple aggregations, and visualisation)[1] while maintaining a streamlined syntax. In contrast, the majority of the work done in the past concentrates on only one category of activities, and as a consequence, it fails to take into account how the effects of one action rely on those of other, more diverse actions.

This article begins by doing a search for the top k most comparable analytical "contexts."

The analytical context is modelled by REACT utilising a tree-based model, where the nodes reflect the user's results "screens" and the edges indicate the user's most recent actions (denoted displays).[33] Following this, context similarity is determined by using the tree edit distance, which is then injected with two innovative ground metrics. These metrics quantify the distance between analytic activities and displays. In the end, The team store contexts in an index structure that makes use of the metric space of the contexts as well as the features of the issue settings The team was working with. The article is able to efficiently extract a collection of possible 'next actions' by drawing parallels between these different situations. The purpose of this study is to investigate the following actions taken by other users in similar scenarios in order to obtain a suitable suggestion for the user's next action. However, since these actions might be quite different and work on a wide variety of datasets,[1] they are not usable in their raw form and need to be processed before they can be turned into suggestions. Because of this, the group came up with a process for creating numerous actions to abstract actions, which conveys common action fragments. The document selects the most relevant generalisations from among a large number of possibilities and presents them to the user as recommendations for the next action to take. Because of this procedure, REACT is able to circumvent the sparsity problem, which is a common issue in recommender systems that occurs when the goals of various users seldom match.[33]

2.4.1 Dataset

When a user imports a specific dataset into an analysis UI, the beginning of a standard IDA procedure has been reached (typically web-based). Then, it carries out a sequence of analytical activities, and after each one, it evaluates the data to determine whether or not it should carry out a new action[29].

In this paper, it was assumed that users perform analysis actions that fall into three main categories: retrieving data operations,[33] data representation operations,[1] and data mining tasks such as cluster The actions and their parameters are represented in this article by a collection of key-value pairs (KVP) with the notation k for the type of

the parameter and v for the value of the parameter. This notation is typical in online applications. The process of interactive analysis on IDA platforms operates in a manner that is intuitively similar to website browsing. At each stage, an individual may initiate an action or retrace to a previous[29]display and follow a different route of travel. Therefore, you should describe the IDA process over dataset D as an ordered labelled tree. The nodes of this tree should represent displays, and the edges that leave each node should be labelled with the action that was done and lead to the subsequent display node. The order accurately depicts the timing for the execution.

2.4.2 Analysis Tree

The conditions that exist at a particular time and place that serve as the backdrop for an occurrence are referred to as the context. The team define the n -context of an action q by the previous n displays that came before q . This definition was inspired by the well-known n -gram model. Certain a dataset D and a series of analysis actions q_1, q_2, \dots, q_m of a given user, which begins with an initial display d_0 and produces subsequent displays d_1, \dots, d_m , an analysis tree may be constructed as follows: $T = (r, V, E,)$ is a tree that has $m + 1$ nodes and is structured so that each node $v_i \in V$ represents a display d_i and each edge $e_i = (v_j, v_i) \in E$ represents an action q_i that operates on d_j and produces d_i . The notation signifies the preorder traversal[39] which is responsible for capturing the execution chronology. More specifically, $v_i v_j$ determines whether or not q_i was carried out before q_j . [29, 8, 23]

2.4.3 Single Parameter generalisation

It was possible to generalise a single key-value parameter pair, denoted by the notation k, v , into the notation $,v$, or to the notation $k,$, where represents the generic variable. These may be further expanded to generate a generalisation for all possible parameters using the notation $,.$ [28] Therefore, this "generalisation" produces a partial order over the template parameters: $,v$ comes before $'k,v'$; $'k,$ comes before $'k,v'$; and so on [28] Lifting these definitions to actions, or sets of parameters, the paper defines an abstract action, or abstraction, as an action whose parameters may consist of other abstract parameters.

This definition is a shorthand way of saying that the action's parameters may be abstract.

2.4.4 System Workflow

The article shows how the system creates a collection of appropriate next-step ideas for the user at each phase in the user's analytical process. An investigation of a dataset is carried out by a user using an IDA interface, as was described in the introduction. At each stage, the most recent analysis action and a condensed description of the display that was produced are both recorded. These are then utilised to create her analysis tree in an incremental fashion. After being taken from the analysis tree, the user's current n-context is then supplied as an input to the REACT recommendation engine.[39] This process ensures that the context repository is effectively indexed. The recommendations engine will first carry out a quick kNN search on the context repository in order to extract the n-contexts that are the most comparable to the one being used by the present user. After this, the system will choose candidate actions from the retrieved collection of n-contexts that are quite similar to one another. In the last step, the system will generalise the potential actions and analyse them in order to provide a set of next-action recommendations that are specific to the user who is currently logged in.

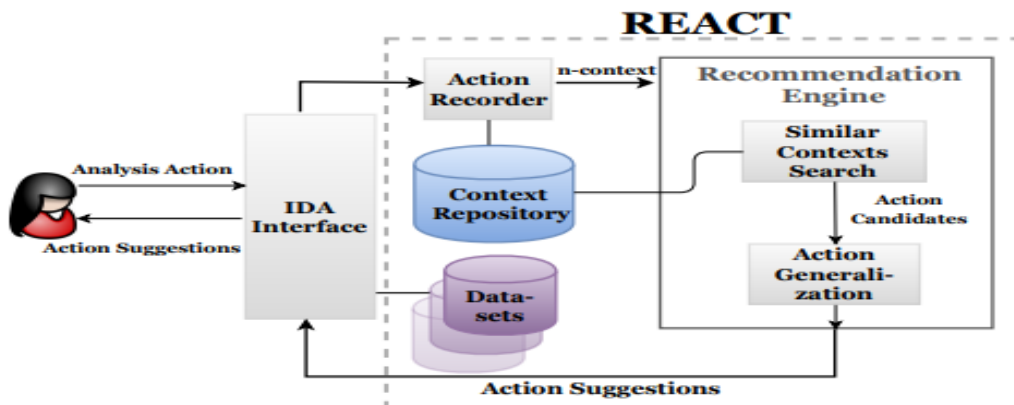


Figure 2.4: REACT system Architecture

2.4.5 Results and Evaluation

The group carried out a comprehensive series of tests using real-world data analysis logs as the basis. In addition to this, they began by conducting an offline assessment, which is standard practise for recommender systems[27]. This evaluation included analysing the capability of produced suggestions to forecast the analysis activities of users. Because of this, they were able to fine-tune the system settings, after which they compared the prediction performance of the system with other baseline methods. Second, the group conducted a "live" experiment with real users to see whether or not REACT can really be utilised in the real world to cut down on analysis times. In the article, the scalability of REACT was investigated using large, simulated analytical workloads. Lastly, the project analysed the influence of the system's settings on both the prediction performance and the execution durations of the model.[24]

The predictive accuracy of a recommender system[2] is a popular metric that may be used to measure the value of such a system. This project recreated the record analysis sessions one at a time, and at each point in a session, react was used to produce suggestions. After that, the project assessed if the recommendations truly correlate to the actual next-action taken by the user at this moment. In every scenario including respond, the number of analysis actions necessary to do the job was cut by around half a percent. In direct proportion, the total amount of time required for analysis was cut by an average of thirty percent, and this occurred irrespective of the dataset used or the sequence in which activities were performed.

This work introduces REACT, a recommender system[24] designed to aid users of contemporary web-based IDA platforms. The work was carried out by REACT. The generic data architecture that REACT employs provides support for a wide variety of high-level action kinds, and it may be simply modified to include more action types. The n-context similarity measure considers not only the syntax of the actions themselves but also the multi-layered displays that correlate to those actions.

2.5 Visual exploration of machine learning results using data cube analysis

The use of machine learning systems is leading to an increase in their complexity, which is a negative consequence of this trend. Extraction of features, transformation of features, model selection, and assessment of models are only few of the stages that are often required when applying machine learning approaches to large-scale, real-world issues.[45]The significance of providing assistance to users in interpreting machine learning models has garnered a growing amount of attention. The instance-based explanation is mostly supported by the visualisation options now available. This article discusses our continuing effort on constructing an interactive visualisation tool for comparing the performances of machine learning models and studying the outcomes of models using data cube analysis. The development of this tool is presented in this paper. Comparing two models based on their overall levels of accuracy is often too coarse and does not facilitate the discovery of contributing causes. On the other hand[28], inspecting individual examples within a large set of data is too contribute to the overall and may not scale well. The goal of this project is to assist users in interactively exploring and determining the appropriate abstraction level of analysis.

Users are able to construct instance subsets by employing relational selects over features inside the MLCube framework[8] that has been presented. Additionally, users are able to calculate aggregate stats and evaluation measures over the subsets. Users have the ability to visually examine these subsets and actively define operators in order to do additional analysis of the findings using the MLCube Explorer. Users of MLCube now have access to a novel method for selecting instance subsets that makes use of both data properties. MLCube Explorer enables users to visually discover aggregate statistics over subgroups of data instances and dynamically drill down into models,[45]which helps users quickly get an overall view of the data as well as model results and spot intriguing patterns and anomalies. This is accomplished by allowing users to visually explore aggregate data over subsets of data instances. This allows users to detect intriguing patterns between the

outcomes of the model and the properties of the system, which leads to the discovery of insights that assist users in comprehending the workings of the models and then further enhancing their capabilities.

2.5.1 Visual Exploration

By accurately portraying the data and enabling people to engage with it, interactive visualisations have been shown to be particularly successful at uncovering intriguing patterns and detecting abnormalities in massive amounts of multidimensional data. This is accomplished by properly expressing the data. In the beginning of the article, The team provided the [10] user interface of our visualisation tool and detail the several ways in which users may interact with the interface. After that, a use example is shown, which explains how the tool may assist a machine learning engineer in better understanding models.

2.5.2 Implementation

The process is followed by the implementation of a straightforward declarative machine learning framework by the project. MLCube[23, 24, 26], which is an application that operates on top of the framework, as well as MLCube explorer. Python, scikit-learn, and PostgreSQL are the underlying technologies that make up the framework. Within the confines of the framework, the project incorporates a number of the learning characteristics that were outlined in the report submitted by the winner of the KDD competition[28]. Cup nd also put into practise a number of models[35], such as logistic regression, decision tree, and boosted tree, all of which were founded on the report. A algorithm contributes to the partial materialisation of MLCube. HTML, JavaScript, and D3.js are the three languages that were used to create MLCube Explorer. It is compatible with all of today's major web browsers. When a user provides two built models to compare, the server delivers the associated MLCube in JSON format to the client, and the client code is responsible for generating the display. This method is an improvement over previous work[33] since it gives users the ability to specify subsets over characteristics or any other

intermediate data and construct visualisations for the cubes in an interactive manner. This approach presents a number of fascinating difficulties regarding scalability.

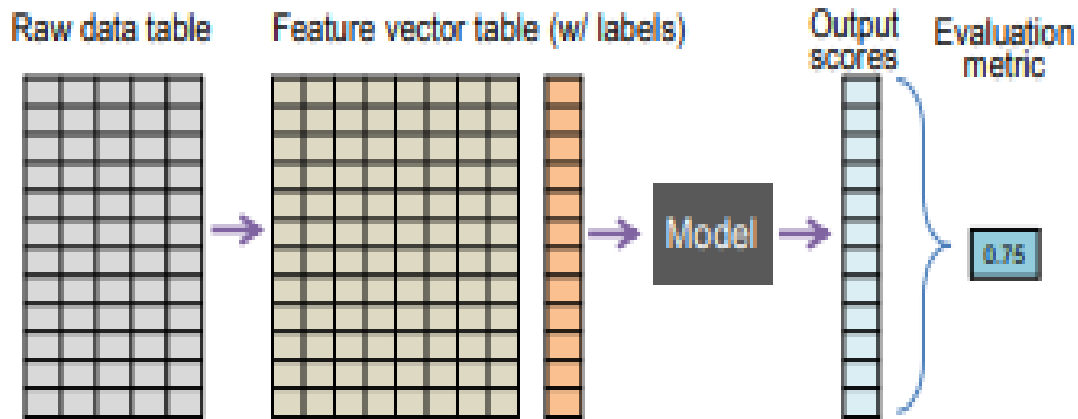


Figure 2.5: General Machine Learning Pipeline

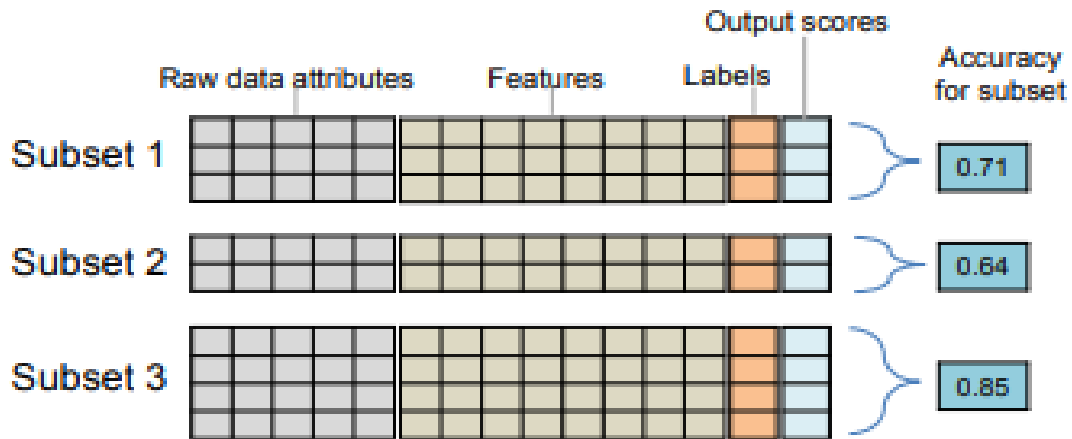


Figure 2.6: Analysis by ML Cube

2.5.3 Data Cubes

In this study, The team offers MLCube, a framework for examining the outputs of machine learning models that is inspired by data cubes. Through interactively exploring and producing a broad variety of instance subsets, the technique allows users to flexibly study and comprehend model outcomes at the subset level. This may be done at any level of the model. Despite the fact that subsets can be defined as any relational selection using a SQL-like expression[26], in practise, a set of dimension attributes (also known as categorical) is typically chosen because of scalability concerns. This is due to the fact that an infinite

number of subsets could be generated. MLCube will choose all categorical characteristics by default (i.e., those with a cardinality that is lower than a specific threshold),[41] and it will build discrete bins for chosen numerical (continuous) attributes and features. The MLCube is then partly materialised for the relevant subgroups in order to expedite the calculation of statistics across these subsets.[44]

2.5.4 Leveraging data Pipeline

The project gives a development platform for developers to create classification models. It has been suggested that interactive tools that support the whole of the machine learning process may help speed up the process of modelling comprehension. The community of database administrators recognises the significance of effectively controlling data flow. With this knowledge,[35] a lot of researchers have done research on how to assist machine learning engineers with feature engineering, but very few people have done research on how to pick models or analyse results.

2.6 SenseMap: Supporting Browser-based Online Sensemaking through Analytic Provenance

When trying to solve difficult problems utilising large datasets over extended periods of time requiring research and analysis, people often get disoriented. They could forget what they said before have done, are unable to locate the knowledge that they have found in the past. One method is to first capture and then depict the interactions of users in a manner that gives an overview of the situation. Regarding the user's involvement in the process of sensemaking.[3] The material that explains such visualization tools exploration and the way the human mind works the procedure that goes along with it is referred to as the analytic provenance.

The issue that was discussed before is referred to as the disorienting problem when it occurs in the setting of the World Wide Web.[9] Using a graphical internet history is one technique that may be used to remedy this issue. It does this by creating a visual repre-

sensation of the online sites that have been viewed as well as the connecting connections that exist between those pages. This allows users to easily determine where they are in the network and navigate to the page that they desire. However, while attempting to do a sensemaking assignment online, which entails collecting, sorting, and reconstructing a large amount of information in order to get insight, the disorientation issue becomes more acute and challenging to solve. They don't simply get disoriented in the hypertext area; they become disoriented in the task space as well.[44]

This article presents a programme called SenseMap[30], which is designed to assist in making sense of data in an online environment using a browser. In order to find a solution to the issue, the team used an iterative design approach that was user-centered. The first step is to conduct interviews in order to collect data on the activities of online sensemakers. After that, a simplified sensemaking model that is based on Pirolli and Card's model [34] is derived in order to better represent these behaviours. Users will iteratively collect sources of information that are relevant to the task, collate them in a manner that makes sense, and eventually communicate their results to others. The team performed a user research in a realistic work context with five participants who all completed the identical sensemaking task relevant to their regular job tasks.[20] The goal of the study was to get a better understanding of how SenseMap is used. The team acquired quantitative data about the activities of SenseMap users as well as qualitative data through semi-structured interviews. Everyone who participated considered both the tool's visual depiction and its interactivity to be very easy to understand and use.

As a free Chrome plugin, SenseMap [30] is readily accessible to users. The primary contribution consists of A user research was conducted to investigate how users make sense of information presented online using the capability of their browsers, and a series of workshops were held thereafter to develop needs and debate ideas. SenseMap is a visual analytics application that supports browser-based online sensemaking and addresses all of the needs that were developed from them. A user assessment that looks at how SenseMap is used in a realistic working environment, together with a discussion of the insights discovered and the design lessons learnt as a result of the evaluation.[41]

2.6.1 Capturing Data

Taking in low-level events is a reasonably simple process, but it offers very little in the way of semantics. It is more typical to capture provenance at the "activity" level since it may be done automatically while still having the potential to offer valuable information.[15] Nevertheless, collecting "sub-tasks" and "tasks" is more difficult since such information is often part of the user's thinking, which is something that computers do not have direct access to.[20] The information that is found via the use of a browser-based sensemaking tool may be gathered at several levels of granularity, such as a website's URL. Users are given the ability to record what they choose with a greater degree of precision thanks to finer-grained capture.

In addition to this manual capture, the history function of the browser makes it possible to automatically record websites that have been visited. Page connecting connections between sites, such as launching a website from a link or hitting the back button on a browser,[26] may also be collected. When to take the picture is an equally important decision that needs to be made, in addition to choosing what to capture. According to the findings,[43] there is not a discernible gap between the two choices in terms of the overall amount of time invested, the cognitive load, or the preferences of the respondents.

2.6.2 Visualisation

Typically, tree visualisation is used to comprehend an overview of provenance data. A vertex is a system state, while an edge is an operation that transitions between states. A branch signifies that the user returns to a previous state and does a different action.[34] To overcome this problem, large network visualisation methods[15] such as clustering or aggregation might be utilised. WindowTrails[15] creates an animation from a lengthy series of subsequent states. This SenseMap use a preexisting tight tree structure and offers semantic zooming. Temporal data may be represented using either color-coded vertices or edge length. A timeline is another frequent way to represent temporal information. WebComets[40, 15] displays browser history with an emphasis on the amount of time spent on every tab and page, as well as the links between sites.[43] Analytic provenance

facilitates the production of visual narratives, during which the user composes results into a coherent tale. A narrative may include provenance information at many levels, including an analytic result, user remarks, visualisations, and raw data. DIVA[40] enables users to develop a story based on recorded visualisation states and user annotations, and to revisit visuals in their captured state. SchemaLine facilitates the creation of narratives by organising user annotations along the timeline. In order to facilitate further analysis, advanced analytic systems often feature a reasoning workspace in which collected information may be freely spatially structured and linked.[31] Additionally, formal analytic techniques of reasoning may be provided. To assist Toulmin argumentation, POLESTAR[5] employs a graphical method; it displays argumentation as a tree structure of supporting/refuting assertions, each supported by at most one piece of evidence. Sandbox facilitates the study of competing hypotheses by giving each supporting/contrary evidence of all hypotheses a score depending on its importance and calculates the final score to aid in user decision-making.

2.6.3 Design

The team used a consumer, iterative design method to create SenseMap, a tool that facilitates online sensemaking. Initially, it recognised current user sensemaking habits using existing browser capability. These online user behaviours resulted in the choice and creation of a sense - making model for web user behaviour.[9] The team conducted semi-structured interviews with nine individuals to investigate their online sensemaking habits for everyday job tasks. The interview was conducted during a typical workday to access the participants' open, active browsers as a representative artefact of their behaviour. As a result, the participant's browser served as the dialogue's skeleton and offered the continuous inquiries as the conversation progressed. This strategy also guaranteed that participants discussed what they really did, as opposed to what they believed they did or what they should do.[37]

There was substantial variation in the usage of browser bookmarks, but the majority had abandoned them in favour of tabs to maintain live, accessible information. Two

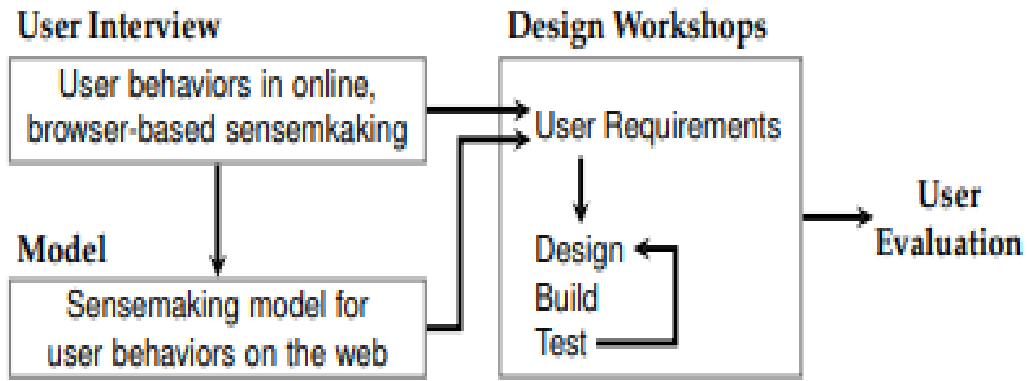


Figure 2.7: Design Architecture

participants did not have any bookmarks. The bookmarks of one participant were not categorised into groups, categories, or folders. At the completion of a project, one person bookmarks the tabs' content and organises it into folders with descriptive names.[31]

Sensemaking Model

The integration of our observed behaviours with the model of Pirolli and Card [34] suggests a browser-based sense - making process in which information sources are stored in a collection of browser tabs (foraging loops), with each tab storing the provenance of the source.[5] There is a continuous curation process (sense - making loop) in which tabs are classified and a story emerges inside such categorised groupings. These groupings and associations constitute the underlying schema. The results of the curation are then utilised as a direction for future, more focused searches and, once completed, as a tool for communicating the discoveries to others.

During the first design session of a workshops, all elicited criteria were examined and it was determined that SenseMap must: Record the web pages visited by the user, the sensemaking activities that occurred on those sites, and how the consumer arrived at those pages; Visualize the gathered data so that the user can comprehend what they've done, how things are related, and what they may do next. Assist the user in curating the acquired data based on its relevance, aid their thinking, and explain the results. Additionally, this should not affect the original connection between gathered data, so

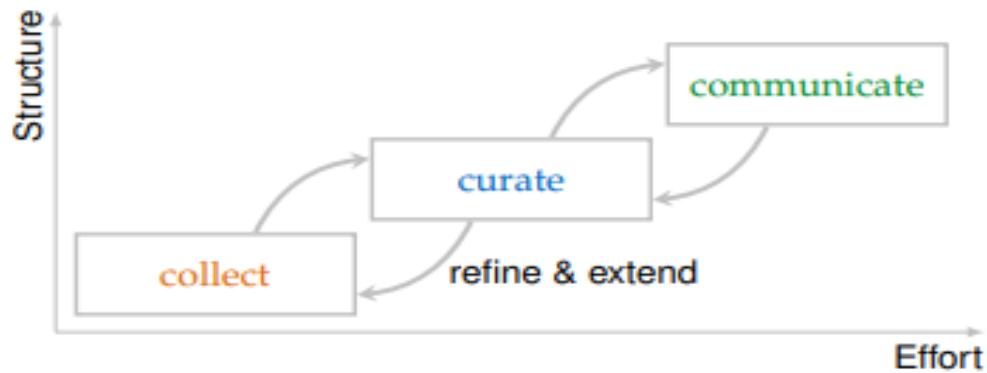


Figure 2.8: Sensemaking Model

that the user may always utilize it as a reference.

Browser View

The editing assistance tools of highlighting and annotating are very necessary. Users are able to annotate pertinent information and apply their own interpretations thanks to these tools. When the user selects a portion of text, a context menu appears with the option "Highlight" added to it. This gives the user the ability to highlight the chosen text. The user will have the option to either add a remark or remove the highlight when the text in question becomes clickable. When a user visits a web page, SenseMap will take a snapshot of the page and utilise that image as a representation of the page in the history map. Its purpose is to facilitate the user's ability to identify previously viewed web sites in a more expedient manner. On the other hand, it's possible that this snapshot may not accurately depict the primary content of the webpage,[40] particularly if there is a lot of information on the page. The editing assistance tools of highlighting and annotating are very necessary. Users are able to annotate pertinent information and apply their own interpretations thanks to these tools. When the user selects a portion of text, a context menu appears with the option Highlight added to it. This gives the user the ability to highlight the chosen text. The user will have the option to either add a remark or remove the highlight when the text in question becomes clickable. When a user visits a web page, SenseMap will take a snapshot of the page and utilise that image as a representation

of the page in the history map. Its purpose is to facilitate the user's ability to identify previously viewed web sites in a more expedient manner. On the other hand, it's possible that this snapshot may not accurately depict the primary content of the webpage,[43] particularly if there is a lot of information on the page.

Histor Map

A bar that has an icon that indicates the action's kind and text that displays the contextual information is used to symbolise an action. Users are able to more quickly understand action kinds with the assistance of icons, and the icon set that The team employed is the same one of the SensePath study [40]. If the action type is set to default surfing, then the preferred icon of the website's home page will be used instead. Because there is a restriction on the amount of space available, the contextual information is abbreviated up to a specific length. This is because the text is essential for comprehending what the action is about.



Figure 2.9: One Highlight and One note

Figure 9 shows how highlights and notes from the same website page have been grouped together. They may be found in distinct rows just under the title of the webpage. In order to maintain a suitable height for the page, the default setting only displays a small number of highlights and annotations. The menu that appears when you hover your

mouse pointer over any highlight or annotation will provide access to all of these options.

KnowledgeMap

The first step in the curation process is transferring nodes from the history map to the knowledge map. This is accomplished by clicking the Curate button that appears in the menu that appears when you hover over a node. The nodes in the knowledge map and the nodes in the history map both have the same visual representation. The limitation of ranking tabs in a single dimension from left to right is circumvented in the knowledge map by the use of a spatial arrangement for the nodes. The user has complete control over the movement of nodes and just needs to drag them in the desired direction[15] When new nodes are added to the history map, the provenance connections that connect those nodes are also transferred to the knowledge map. This provides an early comprehension of the relationships that are already in place. User-added links may be easily distinguished from provenance connections because to the usage of distinct colour schemes. At this time, SenseMap does not provide support for any formal argumentation techniques or procedures. On the other hand, The team believe that the adaptability of the spatial arrangement and the development of linkages may facilitate the user's application of their reasoning processes. Because only the most relevant and significant nodes are selected, the history map contains all of the nodes that are found in the knowledge map, but this may not be the case in the opposite way.[31]

2.6.4 Conclusion

In this work, the team proposes SenseMap, which is designed to assist online sensemaking that is conducted via a browser by using analytical provenance. The sensemaking activities of users are automatically captured by SenseMap in the browser view, and these actions are then shown in the history map. This provides users with an overview of their sensemaking processes, which helps users avoid being disoriented in the tasks. This gives users the ability to choose the information that is most relevant for inclusion in the knowledge map, which improves their overall comprehension of the tasks and may also

direct additional investigation.[30] In the end, users are able to share their discoveries by utilising any one of the three views, each of which provides a different degree of detail. These levels include the summary in the information map, the process in the past map, and the actual data in the browser view.

Chapter 3

Design

The workshop that will be given to the students will be used to collect information about the users' previous search histories, which will then be the primary focus of this project. In this part, the design process that was used to build the module as well as the obstacles that were encountered when creating the VizHiz python script for visual analysis are described in detail.

3.1 Collection of Browser History using History Collector

The History Collector Chrome extension is meant to gather the whole of a user's browser history and save the information in two separate files once it has been compiled. The one and only constraint of using that tool is that it is not possible to gather data for periods longer than four months owing to restrictions imposed by Google Chrome itself. Because the goal of this research is not to develop an extension but rather to analyse data collected from users, I downloaded this extension from github.com after locating it via an anonymous source. Before I could participate in the workshop, I needed to make a few adjustments inside the extension code itself so that it would conform to my specifications. In the workshop, there were six students participated. After sharing History Collector extension with them, I received two files. First was named user a-full.csv. This file has

all the browsing history of user with URL of website, domain name and date time of the search was made. Second was user a-mv.csv which has domain urls and number of time the site was accessed.

3.2 Jupyter Notebook

The Jupyter Notebook has been updated to the next version with JupyterLab. Its primary focus is on enhancing the usability of the Notebook while also significantly broadening its functionality. For collaborative computing and data research in the browser, Since its inception, Jupyter Notebook has been an indispensable tool for every data scientist or data analyst who uses Python. In point of fact, the majority of Python and data science analysis classes offered online make use of Jupyter Notebook. Even while Jupyter Notebook is straightforward and easy to use for novices, it is the superior tool to use when you are engaged in more extensive data analysis tasks.

The only way for me to read CSV files in Jupyter notebook was as text files. On the other hand, we can see it in a tabular format in Jupyterlab, exactly as we would in Excel. In Jupyterlab, I have also attempted to open big CSV files with several million rows, and the programme performs faultlessly and without any issues whatsoever. Simply dragging and dropping elements around allows us to further subdivide the display. It is possible to see and work on several notebooks and files all at once with this feature. I really like how adaptable everything is, as well as how much like a true IDE it seems to be, and how much choice I have over how I want to organise my workflow.

This project uses jupyterlab to create python script for analysis since using code consoles is the simplest method for running code, we often use them whenever we want to try out a piece of code or check out what the return value of a function is. Because of the interaction that they provide, code consoles have proven to be the most effective location for testing code. In order to create a new console for the notebook in Jupyterlab, you only need to right-click anywhere in the notebook and pick the New Console for Notebook option. After that, you may go to the code console to experiment with your code in this environment.

3.3 Jupyter Notebook-Cell Dependency Network

After a cell in a Jupyter Notebook has been generated and validated, it is assigned a unique identification that it will maintain for the duration of its life. This identity cannot be changed. When we use this approach, we will be able to create references to more recent cells in a way that will continue to function normally even when the notebook is reopened. We are in a position to make trustworthy references to the outcomes that occurred in the past.

A dataflow is a set of computing modules that are connected together through connections. The outputs of one module are connected to the inputs of another module by means of these connections. The dataflow is carried out in a bottom-up fashion, with each modules being computed only after it has been verified that all of its reliant components have been up to date and after it has been subjected to the subsequent execution step.

At the present, the contents of each notebook are organised in a chronological order, for example, cell a-cell b-cell c. On the other hand, there are circumstances in which it is not required to calculate cell b in order to derive the answer from cell c, and doing so would take a considerable amount of time. Establishing a visualization of the dependencies as a possible solution to the issue is a possibility. If cell c is dependent on cell a, then the dependency graph may advise us that we are not required to analyse or re-review cell b since cell a has changed; but, it may also warn you that you're going to have a tough time analysing cell c in the event that cell c is dependent on cell a.

It may be challenging to communicate non-linear or branching narratives using the notebooks available today. Therefore, visualising cells in the form of a chart would make it feasible to expand the capacity to tell stories while also making it simpler to comprehend the connections between cells. With the aid of Dataflow notebooks, users are able to construct clear connections between cells. These notebooks are an extension of a Jupyter Notebook ecosystem and the IPython kernel. This enables the generation of a graph that is capable of describing all of the dependencies that have been indicated between cells.

In this context, the employment of discrete and enduring cell identification makes references between cells more trustworthy than the alternative that are now accessible. These

alternatives have been made available. Using the reliance, we are able to choose downstream units that rely on the current one or dynamically adjust the downstream dependencies that are linked with a cell.

Chapter 4

Implementation

JupyterLab's VizHiz is a Python script that offers data analysts a user-friendly interface to enable them manage code modifications, do data analysis, and visualise the findings in a specific Jupyter lab Notebook. This is made possible by JupyterLab's integration of the VizHiz script. This section provides specifics on the integration of the VizHiz Python script into the Jupyter Lab Notebook for the purpose of analysing the user's browsing history. In order for us to comprehend how the analysis of the data is carried out, it is necessary for us to first comprehend how the data was obtained from the users.

4.1 Data Collection using chrome extension-History Collector

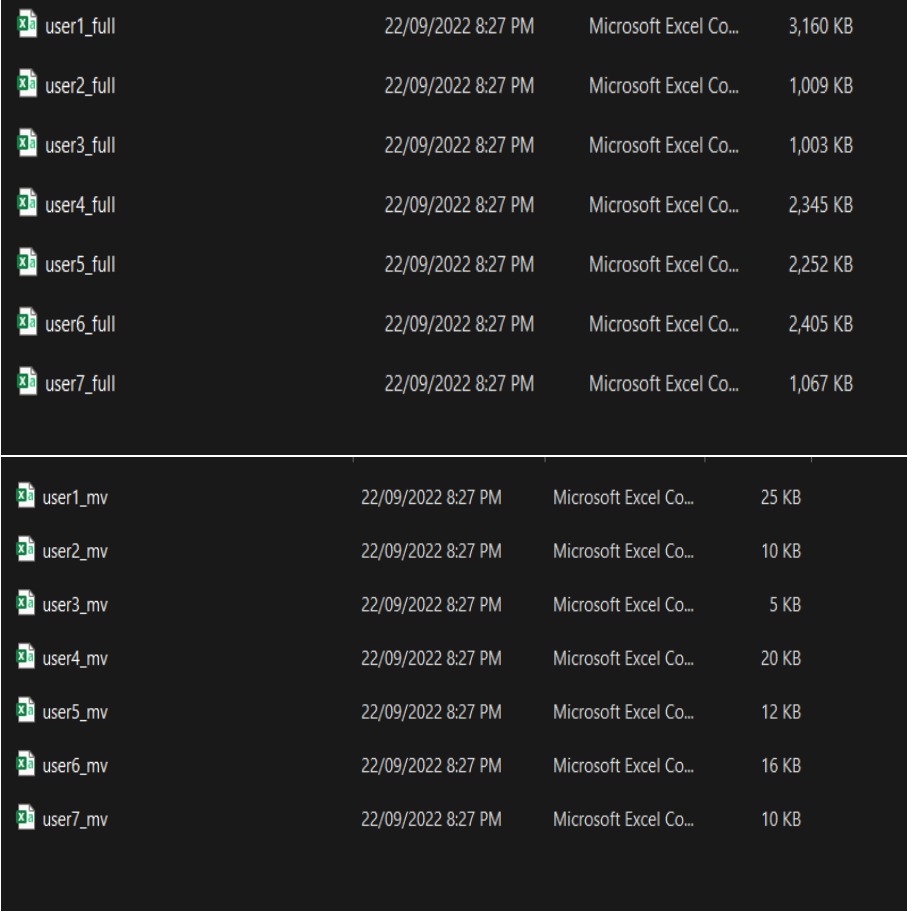
A Google Chrome plugin known as History Collector was used for the purpose of gathering user history. During the session that was done with students, users were shown how to access their internet chrome search history by using an extension that was provided.

4.1.1 Installing Extension

Users added the extension to their browser by installing it. After installing the extension, we were given the ability to see the time range covering when our search history began and ended.

4.1.2 Getting Data

While the user is using the extension, it does its own internal analysis and uploads the user's search history to two separate CSV files.



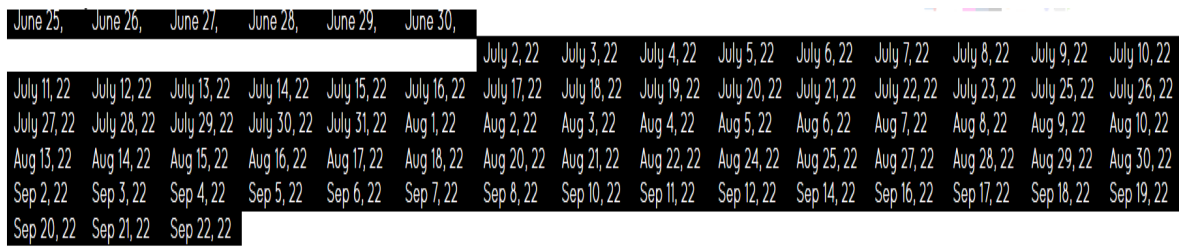
user1_full	22/09/2022 8:27 PM	Microsoft Excel Co...	3,160 KB
user2_full	22/09/2022 8:27 PM	Microsoft Excel Co...	1,009 KB
user3_full	22/09/2022 8:27 PM	Microsoft Excel Co...	1,003 KB
user4_full	22/09/2022 8:27 PM	Microsoft Excel Co...	2,345 KB
user5_full	22/09/2022 8:27 PM	Microsoft Excel Co...	2,252 KB
user6_full	22/09/2022 8:27 PM	Microsoft Excel Co...	2,405 KB
user7_full	22/09/2022 8:27 PM	Microsoft Excel Co...	1,067 KB
user1_mv	22/09/2022 8:27 PM	Microsoft Excel Co...	25 KB
user2_mv	22/09/2022 8:27 PM	Microsoft Excel Co...	10 KB
user3_mv	22/09/2022 8:27 PM	Microsoft Excel Co...	5 KB
user4_mv	22/09/2022 8:27 PM	Microsoft Excel Co...	20 KB
user5_mv	22/09/2022 8:27 PM	Microsoft Excel Co...	12 KB
user6_mv	22/09/2022 8:27 PM	Microsoft Excel Co...	16 KB
user7_mv	22/09/2022 8:27 PM	Microsoft Excel Co...	10 KB

Figure 4.1: User history files

The first collection of data comprises the urls and domain names of websites, as well as the date the search was conducted and the amount of time it took for the page to load. The second collection of files contains a list of the top urls of the most popular websites and a count of the number of times each website was viewed.

4.2 Data Analysis

After the search history was collected from the workshop, all user history was imported into JupyterLab for analysis. Over the course of multiple code snippets, I put together several python script for analyzing the browsing history.



June 25,	June 26,	June 27,	June 28,	June 29,	June 30,										
						July 2, 22	July 3, 22	July 4, 22	July 5, 22	July 6, 22	July 7, 22	July 8, 22	July 9, 22	July 10, 22	
July 11, 22	July 12, 22	July 13, 22	July 14, 22	July 15, 22	July 16, 22	July 17, 22	July 18, 22	July 19, 22	July 20, 22	July 21, 22	July 22, 22	July 23, 22	July 25, 22	July 26, 22	
July 27, 22	July 28, 22	July 29, 22	July 30, 22	July 31, 22	Aug 1, 22	Aug 2, 22	Aug 3, 22	Aug 4, 22	Aug 5, 22	Aug 6, 22	Aug 7, 22	Aug 8, 22	Aug 9, 22	Aug 10, 22	
Aug 13, 22	Aug 14, 22	Aug 15, 22	Aug 16, 22	Aug 17, 22	Aug 18, 22	Aug 20, 22	Aug 21, 22	Aug 22, 22	Aug 24, 22	Aug 25, 22	Aug 27, 22	Aug 28, 22	Aug 29, 22	Aug 30, 22	
Sep 2, 22	Sep 3, 22	Sep 4, 22	Sep 5, 22	Sep 6, 22	Sep 7, 22	Sep 8, 22	Sep 10, 22	Sep 11, 22	Sep 12, 22	Sep 14, 22	Sep 16, 22	Sep 17, 22	Sep 18, 22	Sep 19, 22	
Sep 20, 22	Sep 21, 22	Sep 22, 22													

Figure 4.2: Chrome Extension Output Screen

The extension does internal analysis for segregating data as full search history of user and most visited websites of user individually. And then downloads the files automatically. Afterwards, these files are imported to jupyter lab for analysis.

4.2.1 Python Packages

Because it provides access to an effective N-dimensional array object, the NumPy (Numerical Python) module is the foundational component of Python's numerical calculation capabilities. It was used for processing arrays that may be used for a variety of purposes and offers high-performance multidimensional objects known as arrays as well as tools for interacting with them. NumPy solved the issue of slowness in part by offering these multidimensional arrays as well as functions and operators that perform well on these arrays. In addition, NumPy provided the multidimensional arrays.

Along with NumPy and matplotlib, the Python library known as Pandas is often used throughout the data analysis process. Pandas offered highly quick and flexible data structures, such as data frame CDs, which were developed to deal with structured data in a very simple and straightforward manner.

Matplotlib offers visualisations that are both strong and aesthetically pleasing. Because it is capable of generating graphs and plots, it finds widespread use in the field of data visualisation. Additionally, it included an object-oriented application programming interface (API), which was used in order to incorporate such plots into other programmes.

4.2.2 Analysing browser history

This research categorised the websites that users visit in order to conduct an investigation of their browsing history online. During the course of this procedure, a number of objective characteristics were taken into consideration. The following are some of them:

Top Visited Sites

First, I focused on the top 10 websites that each user visited the most often. I was able to get a visual that shows the domain name as well as the amount of times a person browsed a certain website. Sample for User1 is shown in figure 4.3.

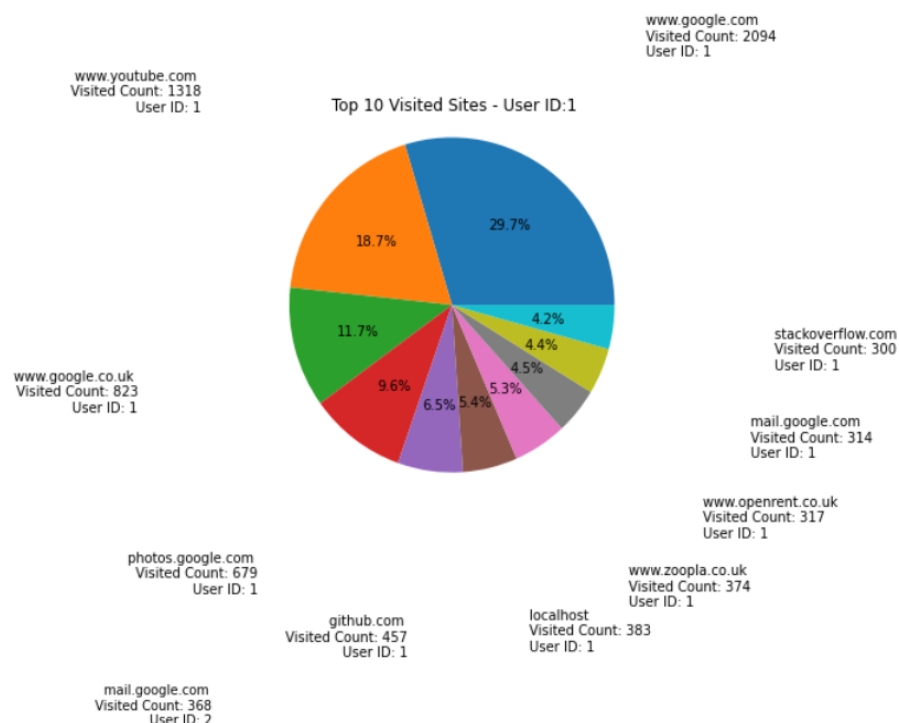


Figure 4.3: Top ten accessed websites by user

Common Domain visited by All Users

Further study was conducted based on the notion, which indicated which websites were most often frequented by each user. First, I needed to determine the total number of domains based on the user data. I discovered that users as a whole had searched for a total of 3690 different domains. After that, I performed a filtering process to identify

websites that were often visited by several users. I received a total of 2892 different website addresses from all of the users.

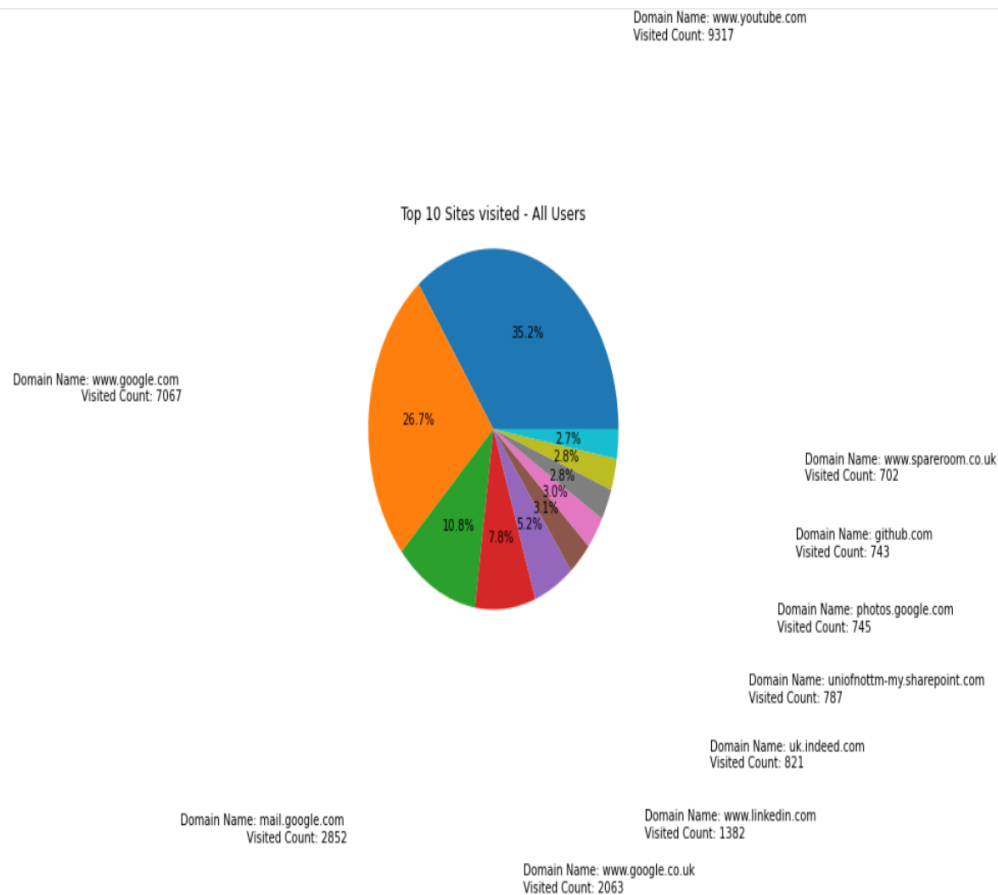


Figure 4.4: Unique ten most accessed websites by user

Analysis of domain count on Daily and Monthly basis

Because it was obvious in the past after utilising the extension that it was gathering search history from the present day back to the previous four months, I was able to organise the total number of searches conducted domain by domain throughout all four months. The same domain dataset, which included frequently visited websites from the histories of all users, was used for further analyses. It aggregates the results of all of the people who are now using a certain website. Figure 4.5, for instance, shows the number of times that YouTube.com was viewed by consumers on each individual day of the month. In a similar vein, analysis was performed on each individual domain in the user's search history.

In the same vein, the analysis was carried out on a regular basis, which provided monthly

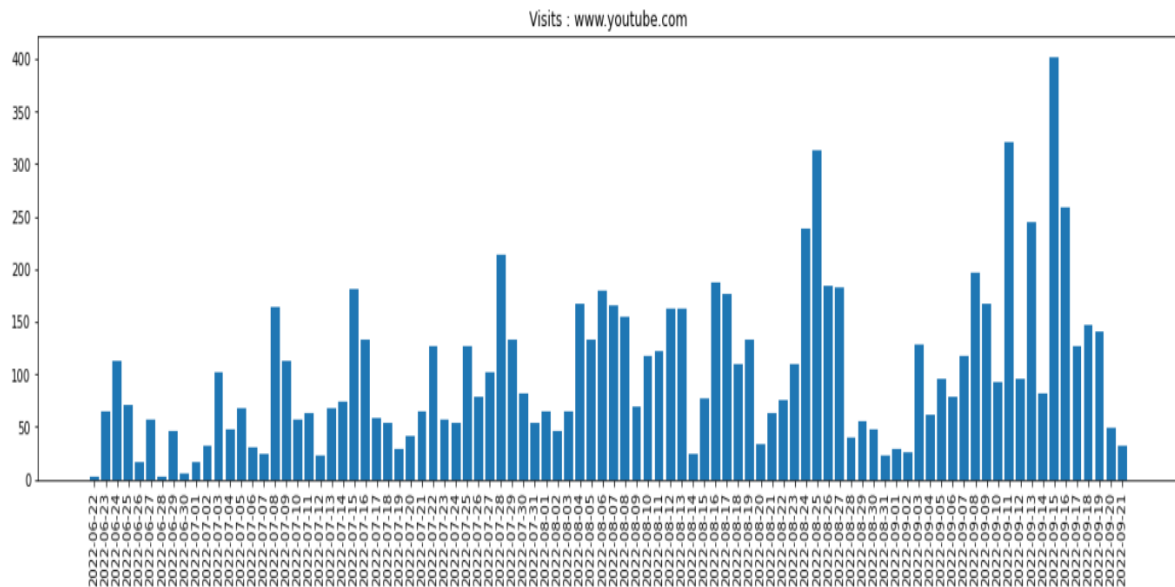


Figure 4.5: Analysis of domain count per day

counts of the websites that were viewed by every user. The example that can be found on the website linkedIn.com is shown in the figure 4.6.

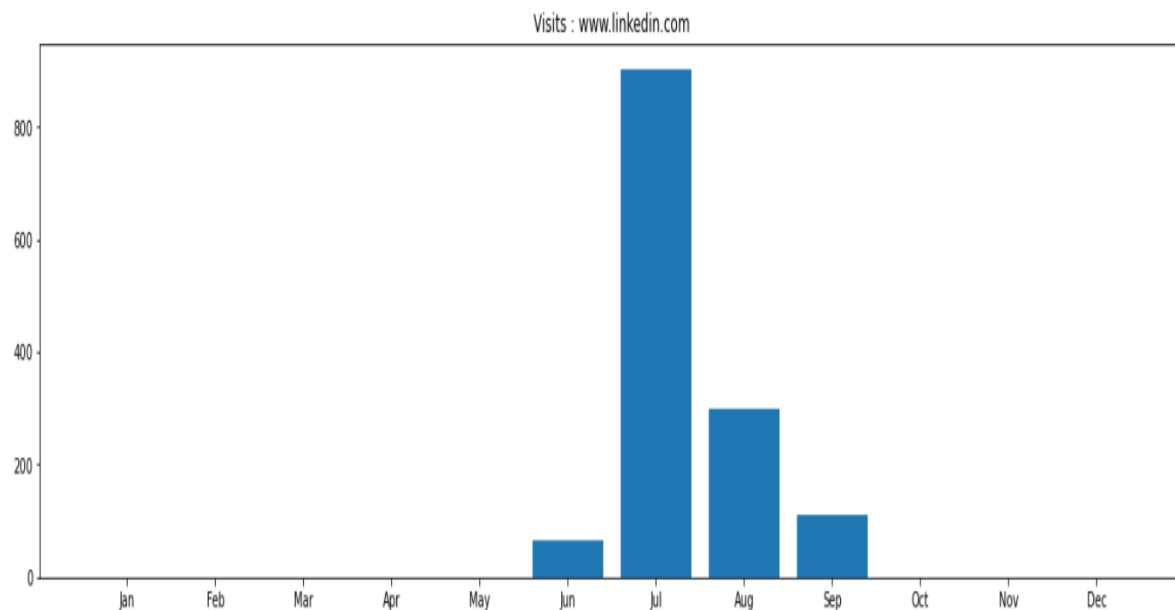


Figure 4.6: Analysis of domain count per month

Website Loading Time

The length of time required for a website or web page to completely load and become visible on screen is referred to as the website load duration or web page load time. This

encompasses everything that can be found on the page, including the text, photographs, and videos. To put it simply, it refers to how quickly all of the information on a web page loads. According to research conducted by Google in 2018, it takes an average of 15.3 seconds for a mobile web page to load. According to the same survey, the issue arises because the majority of portable websites have an excessive number of page components.[42] This issue continues to exist despite the fact that the majority of online traffic now occurs on 4G rather than 3G. When it comes to conversion rates, a load time of 0 to 4 seconds is optimal, and the first five seconds of a page's load time have the most influence on conversion rates. In point of fact, the sites with load times ranging from 0 to 2 seconds have the greatest rates of conversion for online shops. Conversion rates[42] on websites tend to decline by a mean of 4.42percent for every extra second that a page takes to load. Conversion rates on websites typically decrease by 2.11percent for every extra second that it takes for a page to appear.

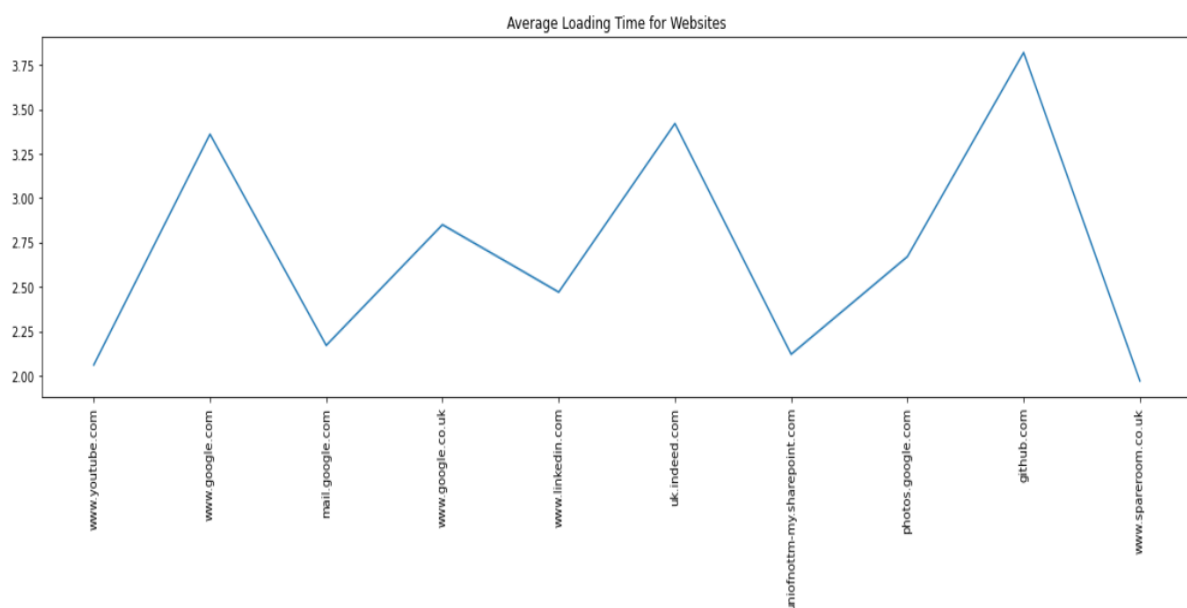


Figure 4.7: Average Loading Time for Websites

Comparing User Searches

At long last, I was able to visualise the results of an analysis of the top website searches performed by each individual user. A comparison between user one and user two's total number of visits to a certain website was made for the purpose of doing more research

and analysis. This provided the study with a surfing routine or habit from user to user, allowing them to compare and contrast the users and uncover differences and similarities. The figure 4.8 Shows a sample comparison done between top sites accessed by user 1 and user 2.

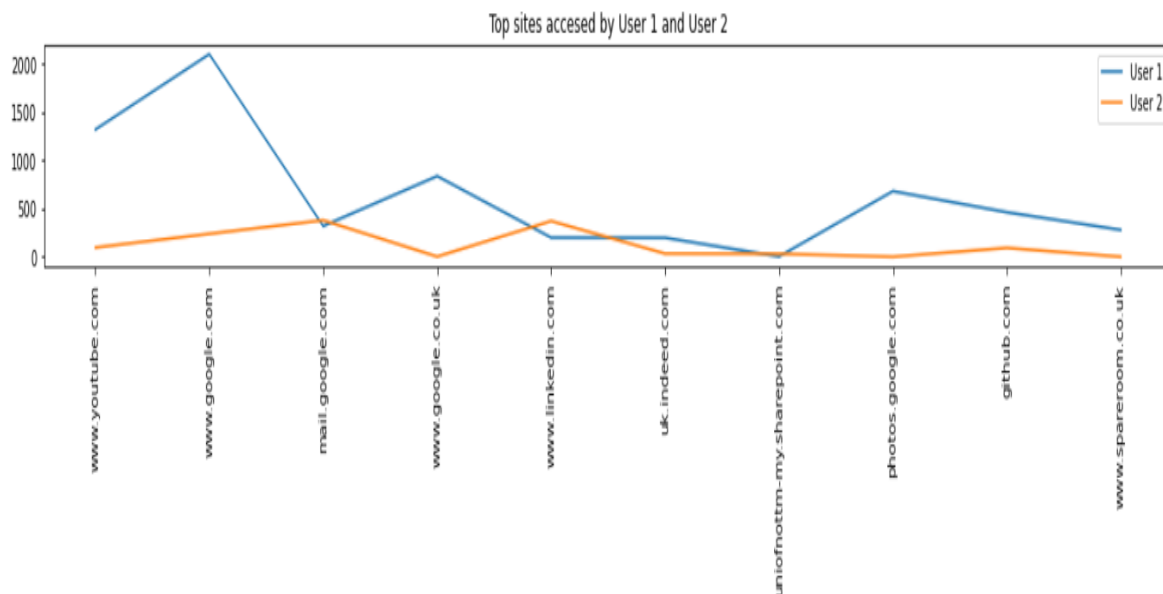


Figure 4.8: Comparing user 1 and user 2 domain search

Chapter 5

Evaluation

VizHiz, a tool that helps data analysts illustrate how researchers may use visual analytics to diagnose and explain occurrences in daily situations, as well as how it can be added to their toolkit as a way of interpretation and analysis, is one of the many applications that VizHiz offers. An explanatory assessment was carried out so that the appropriateness and applicability of the planned work could be determined. The following conclusions were drawn as a result of this evaluation.

5.1 User Evaluation-Confidentiality and Reassuring Remarks

It is crucial to keep the confidence and comfort of users with the analysis, which will allow them to continue curating the material that they have acquired and profit from the process of curation. During the preliminary session, participant worry around keeping and managing chrome search history was noted. It is considered a severe occurrence when collections of browser history are lost. Users of VizHiz are required to have faith that the software will record their browsing habits properly and in a way that will allow them to continue to comprehend the results of the analysis throughout the process. In essence, users delegate control over the collecting portion of their search analysis process to the Python script, and the script then curates this collection in a manner that strives

to create better methods to comprehend and communicate this information (reassurance). VizHiz is designed to support and augment browser-based online visual analysis; as a result, it requires a change in practise from scripting through a collection of browsing history to analysing by engaging with the history. This change is necessary because VizHiz is designed to support and augment browser-based online visual analysis. According to the data, all of the participants who had a good engagement profile were able to make the required practise shift, were comforted by the tool's capacity to assist their work, and retained their faith in it, and ultimately delivered successful results for the project. This new understanding implies that devoting some of one's time to the process of curation is likely to be time well spent. The problem consists in figuring out how to increase trust and reassurance throughout the whole building process.

The screen that appears as a result of collecting the browser history from the user, the URL visiting record and time collected from the searches performed by the user, the title of the URL, and the preference classification that was determined by analysing the text contained in the body of the web pages are manifested. In addition to that, the outcome of a preference categorization that looked at the content of the web pages' bodies is shown.

5.2 Browser History Analysis Evaluation

In this research, a visual analysis of the browsing history of users is performed using 48827 rows of data. It provides the precise count of how many times a person has visited a given URL. Within the scope of this study, the access to a total of 3690 different domains was investigated. There were a total of 2892 websites that were looked through by all of the users. Due to the fact that those who attended the workshop were students, there are a number of patterns in the search history that make sense. According to the findings of the research, the websites linkedin.com, googlescholar.com, indeed.com, github.com, and myuniofnottm.my.sharepoint.com are the ones that get the most visits from people overall.

In addition, during the course, a few of attendees brought up the fact that they were working on their dissertations. A handful of them said that they would soon be moving

out of their student housing and would be looking for houses in the neighbourhood. Based on the visual analysis of search history by month, we can observe that the month of August alone brings in around 600 user visits to the website sparerroom.co.uk. August and September are the months with the highest visit numbers for Github sites. YouTube.com, Google.com, and Google Mail have been the websites that have received the most visits from people overall during whole period. The count is around 1200 to 3500. Since the month of July, employment websites such as Indeed.com and LinkedIn have seen an upsurge in the number of visitors.

At last, a comparison of users on a one-to-one basis was carried out in order to glean more insights from the data. It was discovered that User 5 had a greater propensity than any of the other users to visit amusing websites such as Netflix.com, torrents.com, and justwatch.com, amongst others. In the same amount of time, each user has logged a significant number of visits to the websites sparerroom.com and google.co.uk.

Chapter 6

Conclusion and Future Work

6.1 Results and Conclusion

This study analysed and visualised the user searches conducted on smart devices by accessing the browser history on such devices. It gathered the user's browser history data, which the user contributed via their online searches in their day-to-day life, and submitted it for study so that it could conduct an analysis of the user's preferences. The relevant online sites were then gathered using a Chrome extension, categorised by subject using a machine learning algorithm, and examined to establish which topic and category within the contents were most chosen by the user. In order to improve domain administration based on user preferences, it is recommended that beginning immediately, a visual analysis of the various user data that is created from browser history be performed. The visualisation of the user search results showing the most frequented websites was successfully completed. This section presents user statistics on an individual basis and lists the top 10 websites that users visit. The analysis was carried out by selecting the most popular search websites over the course of four months and tallying up the total number of visitors.

This makes it possible for major domains like Google, YouTube, and Google Scholar etc. to curate the information that is most pertinent to their operations, such as the user load on the server and the number of visits to the domain during a specific timescale. This information could potentially guide further investigation. It could be easier for the

management to implement user information sources, locate and navigate to the sources they desire with less effort, and successfully explain their results with the use of this tool.

6.2 Options for Additional Strides Towards Improvement

According to the results of the assessment, History Collector does a relevant data collection about the search history that was created by the user over the course of the previous four months. In contrast, the effort of collecting browsing history in the actual world might extend over a period of more than four months. Because VizHiz is written as a Python script, we have the ability to carry out an investigation of a more extensive nature and over a longer period of time in order to get a deeper comprehension of how the analysis is used. Second, taking into consideration a greater number of characteristics and a wider range of analytical goals will allow for the extraction of a greater number of insights from the browsing history.

In this research, some beginning-level analysis was done on the amount of time it takes for webpages to load. The total number of page kilobytes, including those for pictures, is no longer the most important element in determining how long a website takes to load. A great number of websites have simplified their programming by minimising their code and compressing it using GZIP.[42] Therefore, the setup of the server and the page is, in many instances, the most important component. When we timed it, 82percent of the pages we tested loaded in less than 5 seconds. The disparity in conversion rates between sites that load very quickly and those that take a long time to load is much more noticeable. The conversion rate of a website that loads in one second is five times greater than that of a website that loads in ten seconds. The findings of my investigation led me to arrive at this conclusion. Additional research and analysis on this topic may be carried out visually.

Bibliography

- [1] ADOMAVICIUS, G., AND TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17, 6 (2005), 734–749.
- [2] AMAR, R., EAGAN, J., AND STASKO, J. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (2005), IEEE, pp. 111–117.
- [3] BAUER, R. S., JADE, T., HEDIN, B., AND HOGAN, C. Automated legal sense-making: the centrality of relevance and intentionality.
- [4] BERNERS-LEE, T., CHEN, Y., CHILTON, L., CONNOLLY, D., DHANARAJ, R., HOLLENBACH, J., LERER, A., AND SHEETS, D. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd international semantic web user interaction workshop* (2006), vol. 2006, Athens, Georgia, p. 159.
- [5] BONCHI, F., CASTILLO, C., GIONIS, A., AND JAIMES, A. Social network analysis and mining for business applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 1–37.
- [6] BRAȘOVEANU, A. M., SABOU, M., SCHARL, A., HUBMANN-HAIDVOGEL, A., AND FISCHL, D. Visualizing statistical linked knowledge for decision support. *Semantic Web* 8, 1 (2017), 113–137.

- [7] CHAN, N. A resource utilization analytics platform using grafana and telegraf for the savio supercluster. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*. 2019, pp. 1–6.
- [8] CHATZOPOULOU, G., EIRINAKI, M., AND POLYZOTIS, N. Query recommendations for interactive database exploration. In *International Conference on Scientific and Statistical Database Management* (2009), Springer, pp. 3–18.
- [9] CONKLIN, J. Hypertext: An introduction and survey. *computer* 20, 09 (1987), 17–41.
- [10] DAWKES, H., TWEEDIE, L. A., AND SPENCE, B. Vicki: the visualisation construction kit. In *Proceedings of the workshop on Advanced visual interfaces* (1996), pp. 257–259.
- [11] DORSEY, J. Just setting up my twttr. *Twitter. Oldest Tweet ever from Twitter founder* (2006).
- [12] FURLANI, T. R., SCHNEIDER, B. L., JONES, M. D., TOWNS, J., HART, D. L., GALLO, S. M., DELEON, R. L., LU, C.-D., GHADERSOHI, A., GENTNER, R. J., ET AL. Using xdmmod to facilitate xsede operations, planning and analysis. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery* (2013), pp. 1–8.
- [13] GIOVANETTI, R., AND LANCIERI, L. Model of computer architecture for on-line social networks flexible data analysis: The case of twitter data. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2016), IEEE, pp. 677–684.
- [14] GIOVANETTI, R., AND LANCIERI, L. Model of computer architecture for on-line social networks flexible data analysis: The case of twitter data. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2016), IEEE, pp. 677–684.
- [15] GOTZ, D., WHEN, Z., LU, J., KISSA, P., CAO, N., QIAN, W. H., LIU, S. X., AND ZHOU, M. X. Harvest: an intelligent visual analytic tool for the masses. In

- Proceedings of the first international workshop on Intelligent visual interfaces for text analysis* (2010), pp. 1–4.
- [16] GRAVES, A. Creation of visualizations based on linked data. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics* (2013), pp. 1–12.
- [17] GU, Y. H., YOO, S. J., PIAO, Z., LIN, Y., YAN, J., AND PARK, J. H. User preference analysis and visualization through the browser history of smart devices. In *Proceedings of the 2015 International Conference on Big Data Applications and Services* (2015), pp. 264–267.
- [18] GU, Y. H., YOO, S. J., PIAO, Z., LIN, Y., YAN, J., AND PARK, J. H. User preference analysis and visualization through the browser history of smart devices. In *Proceedings of the 2015 International Conference on Big Data Applications and Services* (New York, NY, USA, 2015), BigDAS '15, Association for Computing Machinery, p. 264–267.
- [19] GUPTA, V., LEHAL, G. S., ET AL. A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence* 1, 1 (2009), 60–76.
- [20] HEER, J., MACKINLAY, J., STOLTE, C., AND AGRAWALA, M. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE transactions on visualization and computer graphics* 14, 6 (2008), 1189–1196.
- [21] HEIM, P., LOHMANN, S., TSENDRAGCHAA, D., AND ERTL, T. Semlens: Visual analysis of semantic data with scatter plots and semantic lenses. In *Proceedings of the 7th International Conference on Semantic Systems* (2011), pp. 175–178.
- [22] JAMES, J., ET AL. Data never sleeps 3.0. *Retrieved Syyskuu 28* (2016), 2016.
- [23] KAHNG, M., FANG, D., AND CHAU, D. H. Visual exploration of machine learning results using data cube analysis. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (2016), pp. 1–6.

- [24] KRAUSE, J., PERER, A., AND NG, K. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI conference on human factors in computing systems* (2016), pp. 5686–5697.
- [25] KRIKORIAN, R. Map of a twitter status object. *The Wall Street Journal* 18 (2010).
- [26] KULESZA, T., BURNETT, M., WONG, W.-K., AND STUMPF, S. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces* (2015), pp. 126–137.
- [27] LIU, J., WILSON, A., AND GUNNING, D. Workflow-based human-in-the-loop data analytics. In *Proceedings of the 2014 Workshop on Human Centered Big Data Research* (2014), pp. 49–52.
- [28] MILO, T., AND SOMECH, A. React: Context-sensitive recommendations for data analysis. In *Proceedings of the 2016 International Conference on Management of Data* (2016), pp. 2137–2140.
- [29] MILO, T., AND SOMECH, A. React: Context-sensitive recommendations for data analysis. In *Proceedings of the 2016 International Conference on Management of Data* (2016), pp. 2137–2140.
- [30] NGUYEN, P. H., XU, K., BARDILL, A., SALMAN, B., HERD, K., AND WONG, B. W. Sensemap: Supporting browser-based online sensemaking through analytic provenance. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2016), IEEE, pp. 91–100.
- [31] NGUYEN, P. H., XU, K., WALKER, R., AND WONG, B. W. Timesets: Timeline visualization with set relations. *Information Visualization* 15, 3 (2016), 253–269.
- [32] PALMER, J. T., GALLO, S. M., FURLANI, T. R., JONES, M. D., DELEON, R. L., WHITE, J. P., SIMAKOV, N., PATRA, A. K., SPERHAC, J., YEARKE, T., RATHSAM, R., INNUS, M., CORNELIUS, C. D., BROWNE, J. C., BARTH, W. L., AND EVANS, R. T. Open xdmmod: A tool for the comprehensive management of

- high-performance computing resources. *Computing in Science Engineering* 17, 4 (2015), 52–62.
- [33] PIETRIGA, E. Semantic web data visualization with graph style sheets. In *Proceedings of the 2006 ACM symposium on Software visualization* (2006), pp. 177–178.
- [34] PIROLI, P., AND CARD, S. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis* (2005), vol. 5, McLean, VA, USA, pp. 2–4.
- [35] PLAISANT, C., MILASH, B., ROSE, A., WIDOFF, S., AND SHNEIDERMAN, B. Lifelines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1996), pp. 221–227.
- [36] RUSSELL, M. A. *Mining the social web: data mining Facebook, Twitter, LinkedIn, Google+, GitHub, and more.* ” O’Reilly Media, Inc.”, 2013.
- [37] SHRINIVASAN, Y. B., AND VAN WIJK, J. J. Supporting the analytical reasoning process in information visualization. In *Proceedings of the SIGCHI conference on human factors in computing systems* (2008), pp. 1237–1246.
- [38] THELLMANN, K., GALKIN, M., ORLANDI, F., AND AUER, S. Linkdavis—automatic binding of linked data to visualizations. In *International Semantic Web Conference* (2015), Springer, pp. 147–162.
- [39] VARTAK, M., RAHMAN, S., MADDEN, S., PARAMESWARAN, A., AND POLYZOTIS, N. Seedb: Efficient data-driven visualization recommendations to support visual analytics. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases* (2015), vol. 8, NIH Public Access, p. 2182.
- [40] WALKER, R., SLINGSBY, A., DYKES, J., XU, K., WOOD, J., NGUYEN, P. H., STEPHENS, D., WONG, B. W., AND ZHENG, Y. An extensible framework for provenance in human terrain visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2139–2148.

- [41] WANG, A. Y., WANG, D., DROZDAL, J., MULLER, M., PARK, S., WEISZ, J. D., LIU, X., WU, L., AND DUGAN, C. Documentation matters: Human-centered ai system to assist data science code documentation in computational notebooks. *ACM Transactions on Computer-Human Interaction* 29, 2 (2022), 1–33.
- [42] WIEGAND, M. Site speed is (still) impacting your conversion rate, 2022.
- [43] XU, K., NGUYEN, P., AND FIELDS, B. Visual analysis of streaming data with savi and sensemap. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2015).
- [44] ZASTRE, M. Jupyter notebook in cs1: An experience report. In *Proceedings of the Western Canadian Conference on Computing Education* (2019), pp. 1–6.
- [45] ZHANG, K., AND SHASHA, D. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing* 18, 6 (1989), 1245–1262.