Kingdom of Saudi Arabia
Majmaah University
Ministry of Higher
Education
College of Science Al Zulfi

المملكة العربية السعودية
جامعة المجمعه
وزارة التعليم العالي
كلية العلوم بالزلفي

Webpage Classification

Student Affairs System
For College of science Al Zulfi
Department of Computer Science and Information

Graduation Project (2)

Submitted in partial fulfillment of the requirements for the award of
Bachelor’s degree of the Majmaah University
(Semester 2, 2019-2020)

Submitted by:

Mohammed Hamad Al-Hamad.
361102504

Under the supervision of:
Dr. Wael Khedr

Abstract

Nowadays, a webpage searching is no longer hard or need experience because of help of the search engines as Google, Yahoo and Bing. Even though when we want to find a related webpages based on some sub-links or text contents, we found a hard sometimes to explore it. This research is going to simplify this problem through applying web mining techniques. Web page classification, also known as a web page categorization, to classify webpages categories based on their structure and un-structure content. This will help to make finding similar websites easier and more efficient by using some web mining techniques and algorithms.

Keywords: Web Mining, Classification algorithms, MATLAB, Webpage classification.

Acknowledgement

First, I would like to thank ALLAH for all the successes happened to me then I would like to thank my supervisor of this research project Dr. Wael Khedr, for his guidance and assistance in work time and non-work time to do this. And I would also like to thank the project-based committee for their clear effort in helping and guiding students throughout the whole semester, and Dr. Ashraf Haider for his kindness with every help I needed, and special thanks to Dr. Yousef quaqzeh for his help and guidance in my journey in the past 5 years in Computer science.

Certification
MAJMAAH UNIVERSITY,

COLLEGE OF SCIENCE AL ZULFI,
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION

(CERTIFICATE BY STUDENT)

This is to certify that the project titled **“Web page classification”** submitted by me (**Mohammed Hamad Ahmad Al-Hamad**, **361102504**) under the supervision of **Dr. Wael Khedr** for award of Bachelor’s degree of the Majmaah University carried out during the Semester 2, 2019-2020 embodies my original work.

Signature in full: --

Name in block letters: Mohammed Hamad Ahmad Al-Hamad

Student ID: 361102504

Date: April 10, 2020

Table of contents

List of figures

List of tables

Chapter 1: Introduction

Over the past decade we have witnessed an explosive growth on the Internet, with millions of web pages on every topic easily accessible through the Web [11]. The Internet is a powerful medium for communication between computers and for accessing online documents all over the world, but it is not a tool for locating or organizing the mass of information. Tools like search engines assist users in locating information on the Internet. They perform excellently in locating but provide limited ability in organizing the web pages. Internet users are now confronted with thousands of web pages returned by a search engine using simple keyword search. Searching through those web pages is becoming impossible for users. Thus, it has been of more interest in tools that can help make a relevant and quick selection of information that we are seeking.

Web contents of the web page include online documents, e-books, articles, technical reports, digital libraries which have been rapidly exploring all time. Categorizing the web pages is very useful for efficient contents browsing, managing and spam filtering. This process is also difficult for search engine to identify a particular web page [7]. The web pages are retrieved based on the content and structure of a web page using web mining techniques. Web mining is used to extract useful patterns from web pages. It can be classified into three types, namely content mining, structure mining and usage mining.

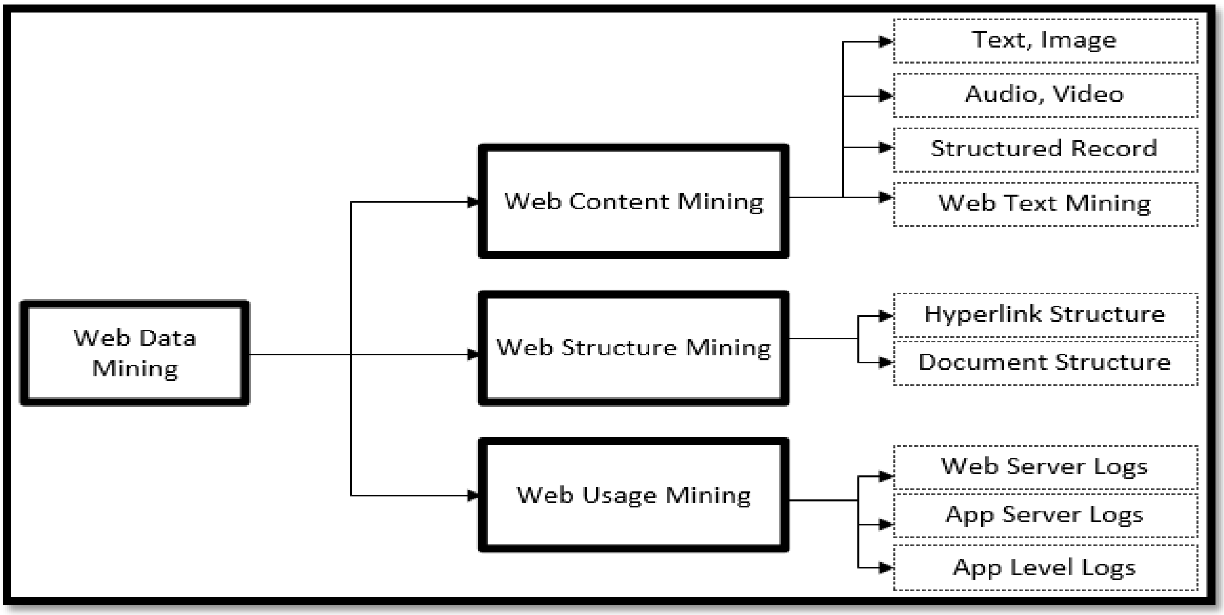


Figure 1 - Web mining types

Web content mining is used to extract the web information based on content which includes (text, images, audio, video, etc.) [11] Web structure mining is used to extract the information through hyperlinks. Web usage mining is used to find the behavior of online users. In web page classification, the information is extracted based on content, links and usage of web data using web mining techniques. It is a process of categorizing the web pages with meaningful predefined category labels. The main problem in web mining is to classify the web pages and it can be done using optimization algorithms, classification algorithms, feature selection and feature extraction algorithms. And its techniques are used to fetch knowledge from web data and can be used to extract relevant information from web pages or web documents.

Chapter 2: Literature Review (survey)

A survey on web distribution papers by S.Lassri, H.Benlahmar, and A.Tragha[2] for the latest methods to inform future classifier implementations, they created a synthesis matrix that helps them record the main points of each source and document how sources relate to each other. They generated this matrix automatically by writing a scraping script, which is the process of downloading data from ScienceDirect [4] and Springer [5] websites after introducing web page classification as the search keyword and extracting valuable information from that data. they develop this script with python and beautiful soup, which allows them to manually extract the elements needed for their study from the selected websites. Each matrix contains that information about each article: year of publication, title, link, type, authors, abstract, keywords, used classifiers, highlights for science direct articles and references for springer articles. they choose springer and ScienceDirect because they contain the largest number of articles related to the topic of web page classification. Post-processing is necessary to make data cleanest. **(Fig.2)** depicts the number of papers dealing with web page classification within each year. Here we can notice that the web page classification topic has moved from marginalization to mainstream in recent years. It is increasingly treated from 2004 and yields a peak in 2018. SVM and neural network are the most used classifiers, according to **(Fig.2)**. They marked a difference of a hundred articles with Naïve Bayes and decision tree.

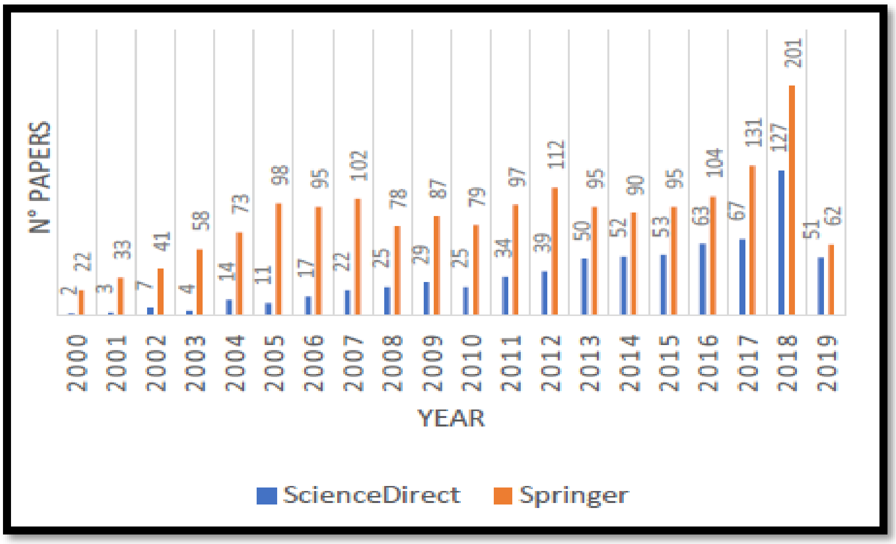


Figure 2- Year-wise distribution of papers

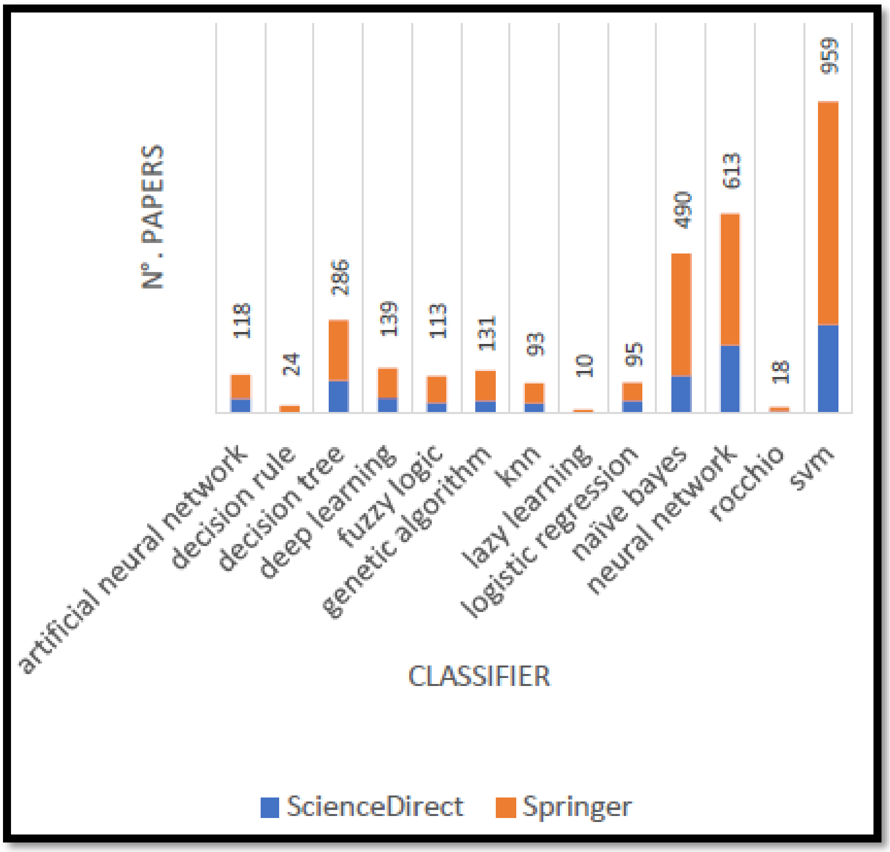


Figure 3- Distribution of papers for classifiers.

Figure (3) describes the evolution of the percentage of classifiers used over the years, which reinforces the last remark and shows that **Support vector machine** (**SVM**) and **artificial neural network (ANN)** represent more than **50%** of the classifiers used. It has also been noted that **deep learning** is increasingly being chosen in the last three years.

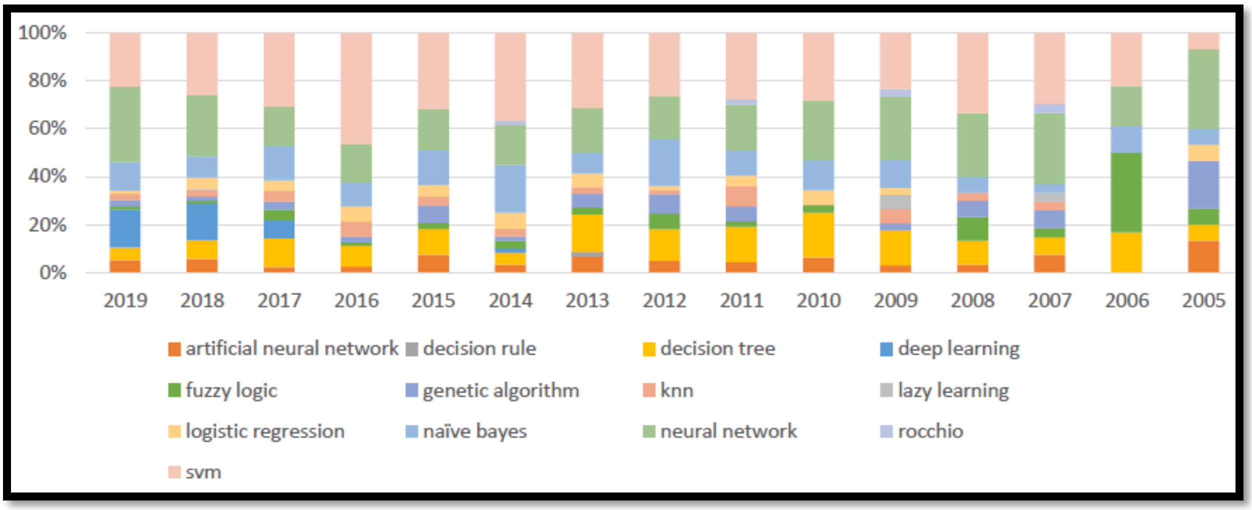


Figure 4 - Year wise distribution of articles for a classifier

Chapter 3: Research Objectives

3.1 Goals

A study of web page classification algorithms, which can efficiently support diversified applications [1], such as:

- Web Mining.
- Information filtering and information retrieval.
- Search engine.
- User profile mining.

3.2 Objectives

- Features extraction of web page.
- Classify a new web page.
- Information retrieval.
- Features of web pages.
- Clustering of web pages and sites.
- Search engine application such as Google and yahoo.

Chapter 4: Research Methodology

4.1 Web Page Classification

Web page classification, also known as a web page categorization, may be defined as the task of determining whether a web page belongs to a category or categories.[12] First, web pages are semi-structured text documents that are usually written in HTML. Secondly, web pages are connected to each other forming direct graphs via *hyperlinks*. Thirdly, web pages are often short and by using only text in those web pages may be insufficient to analyze them. Finally, the sources of web pages are numerous, nonhomogeneous, distributed, and dynamically changing. Choi and Yao [1] gave a formal definition as bellow: "Let $C=\{c_1,\ldots\ldots,c_k\}$ be a set of predefined categories, $D=\{d_1,\ldots\ldots,d_N\}$ be a set of web pages to be classified, and A = D×C be a decision matrix as described in Table(1).

Table 1 - Decision Matrix

Web Pages	Categories				
	C_1	…	Cj	…	C_K
d_1	a_{11}	…	a_{1j}	…	a_{1K}
…	…	…	…	…	…
d_i	a_{i1}	…	a_{ij}	…	a_{iK}
…	…	…	…	…	…
d_N	a_{N1}	…	a_{Nj}	…	a_{NK}

Where, each entry a_{ij} $(1\leq i\leq N,1\leq j\leq K)$ represents whether web page d_i belongs to category c_j or not. Each $a_{ij} \epsilon \{0,1\}$ where 1 indicates web page d_i belongs to category c_j, and 0 for not belonging. A web page can belong to more than one category. The task of web page classification is to approximate the unknown assignment function $f{:}D\times C\rightarrow\{0,1\}$ using a learned function $f'{:}D\times C\rightarrow\{0,1\}$, called a classifier, a model, or a hypothesis, such that f' coincides to f as much as possible [2].

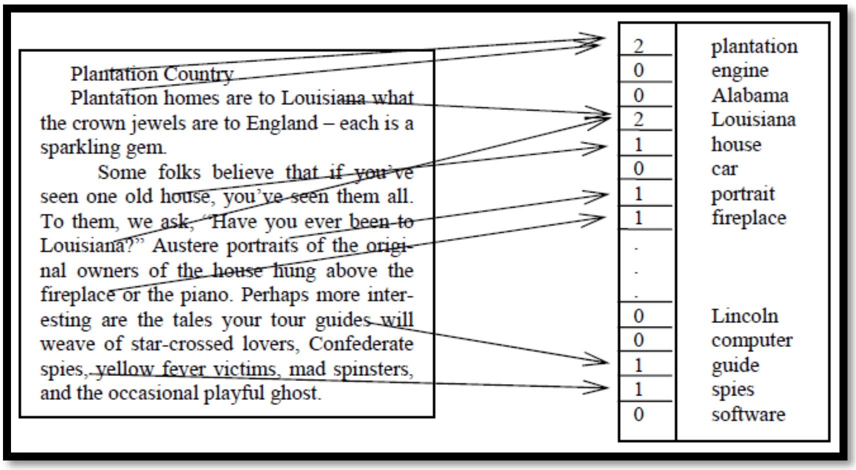
4.2 Web Page Representation

The first step in web page classification is to transform a web page, which typically composes of strings of characters, hyperlinks, images, and HTML tags, into a feature vector [10]. It first defines two types of web page classification, subject based and genre-based classifications.

4.2.1 Representations for Subject Based Classification

Most work for subject-based classifications [10] believes the text source represents the content of a web page. web pages are first preprocessed to discard the less important data. The preprocessing consists of the following steps:

- Removing HTML tags.
- Removing stop words.
- Removing rare words [6].
- Performing word stemming.



Plantation Country

Plantation homes are to Louisiana what the crown jewels are to England – each is a sparkling gem.

Some folks believe that if you've seen one old house, you've seen them all. To them, we ask, "Have you ever been to Louisiana?" Austere portraits of the original owners of the house hung above the fireplace or the piano. Perhaps more interesting are the tales your tour guides will weave of star-crossed lovers, Confederate spies, yellow fever victims, mad spinsters, and the occasional playful ghost.

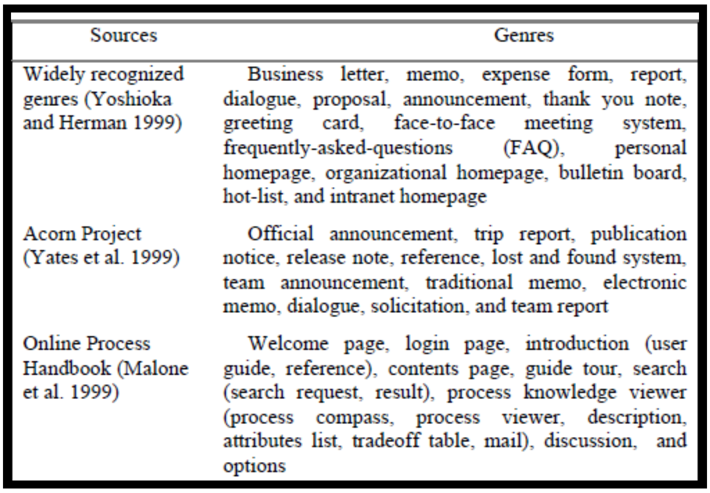
2	plantation
0	engine
0	Alabama
2	Louisiana
1	house
0	car
1	portrait
1	fireplace
.	
.	
.	
0	Lincoln
0	computer
1	guide
1	spies
0	software

Figure 5 - Representing a web page in a vector space model. Each web page is converted into a vector of words in this case.

4.2.2 Representations for Genre-based classification

Web pages are classified depending on functional or genre related factors. In a broad sense, the word "genre" is used here merely as a literary substitute for "a kind of text". This approach can help users find immediate interests. [10] Although text genre has been studied for a long history in linguistic literature, automatically text. As we can see from Table (**2**), genre is a subtle and difficult to define notion.

Table 2 - Current contents in genre

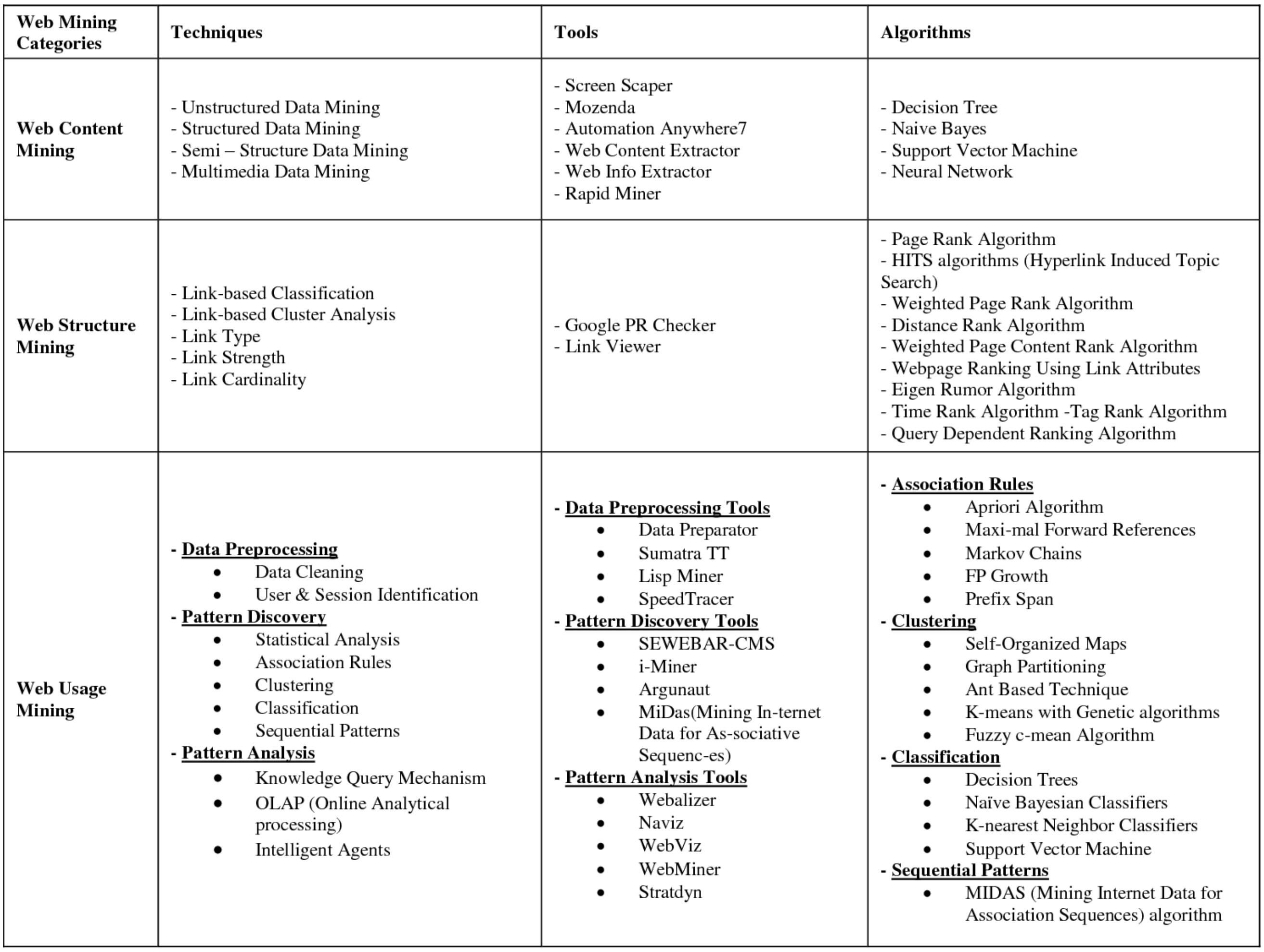


Sources	Genres
Widely recognized genres (Yoshioka and Herman 1999)	Business letter, memo, expense form, report, dialogue, proposal, announcement, thank you note, greeting card, face-to-face meeting system, frequently-asked-questions (FAQ), personal homepage, organizational homepage, bulletin board, hot-list, and intranet homepage
Acorn Project (Yates et al. 1999)	Official announcement, trip report, publication notice, release note, reference, lost and found system, team announcement, traditional memo, electronic memo, dialogue, solicitation, and team report
Online Process Handbook (Malone et al. 1999)	Welcome page, login page, introduction (user guide, reference), contents page, guide tour, search (search request, result), process knowledge viewer (process compass, process viewer, description, attributes list, tradeoff table, mail), discussion, and options

4.3 Web mining techniques, tools, and algorithms

Table (3) list Web mining techniques, tools, and algorithms [8], that are in every type of Web mining and we will explain the algorithms we will use in our research at Web content mining and Structure mining.

Table 3 - Web mining Techniques, tools, and algorithms



Web Mining Categories	**Techniques**	**Tools**	**Algorithms**
Web Content Mining	- Unstructured Data Mining - Structured Data Mining - Semi – Structure Data Mining - Multimedia Data Mining	- Screen Scaper - Mozenda - Automation Anywhere7 - Web Content Extractor - Web Info Extractor - Rapid Miner	- Decision Tree - Naive Bayes - Support Vector Machine - Neural Network
Web Structure Mining	- Link-based Classification - Link-based Cluster Analysis - Link Type - Link Strength - Link Cardinality	- Google PR Checker - Link Viewer	- Page Rank Algorithm - HITS algorithms (Hyperlink Induced Topic Search) - Weighted Page Rank Algorithm - Distance Rank Algorithm - Weighted Page Content Rank Algorithm - Webpage Ranking Using Link Attributes - Eigen Rumor Algorithm - Time Rank Algorithm -Tag Rank Algorithm - Query Dependent Ranking Algorithm
Web Usage Mining	**- Data Preprocessing** • Data Cleaning • User & Session Identification **- Pattern Discovery** • Statistical Analysis • Association Rules • Clustering • Classification • Sequential Patterns **- Pattern Analysis** • Knowledge Query Mechanism • OLAP (Online Analytical processing) • Intelligent Agents	**- Data Preprocessing Tools** • Data Preparator • Sumatra TT • Lisp Miner • SpeedTracer **- Pattern Discovery Tools** • SEWEBAR-CMS • i-Miner • Argunaut • MiDas(Mining In-ternet Data for As-sociative Sequenc-es) **- Pattern Analysis Tools** • Webalizer • Naviz • WebViz • WebMiner • Stratdyn	**- Association Rules** • Apriori Algorithm • Maxi-mal Forward References • Markov Chains • FP Growth • Prefix Span **- Clustering** • Self-Organized Maps • Graph Partitioning • Ant Based Technique • K-means with Genetic algorithms • Fuzzy c-mean Algorithm **- Classification** • Decision Trees • Naïve Bayesian Classifiers • K-nearest Neighbor Classifiers • Support Vector Machine **- Sequential Patterns** • MIDAS (Mining Internet Data for Association Sequences) algorithm

Chapter 5: Management Plans

5.1 Management Plan 1

Table 4 - Management Plan 1

ID	Task	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11
1	Study data mining basics.	→	→	→	→							
2	Search in Web mining and read research papers.		→	→	→	→	→	→	→			
3	Study some classification algorithms & prepare the first presentation and present it.				→	→	→	→				
4	Start learning on MATLAB & apply some HTML text extraction functions.					→	→	→	→	→	→	→
5	Apply on MATLAB and extract keywords for some webpages.						→	→	→	→	→	
6	Design the DB structure for the project and present presentation 2.									→	→	
7	Prepare and submit Final report, Presentation, Poster, and Proposal.										→	→

5.2 Management Plan 2

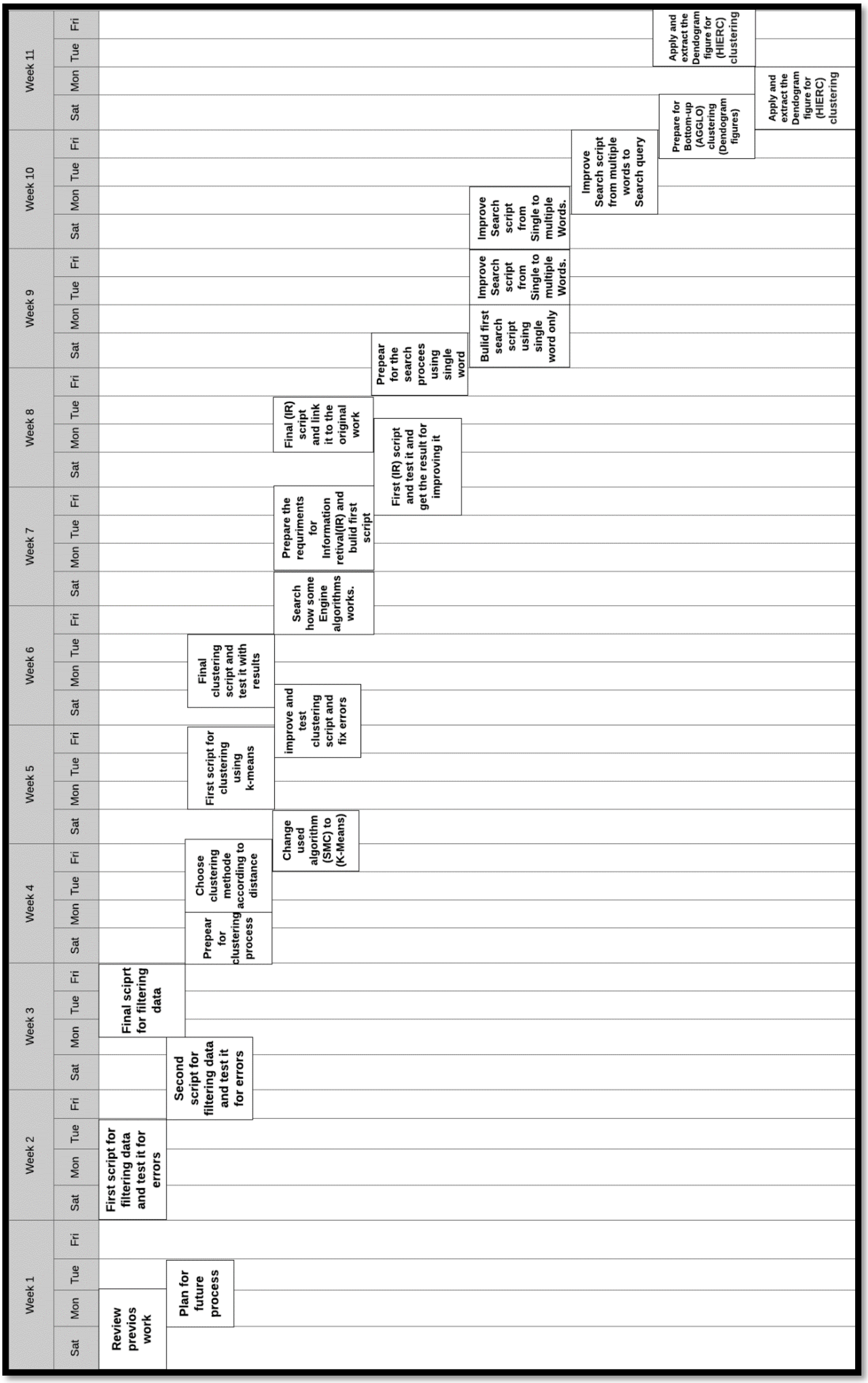


Figure 6 - Gantt chart for GP2

Chapter 6: Expected Results and their Utilization

Classify webpages based on their structure and un-structure contents, and represent webpages as Graph(structure), Context (Data base). And apply a classification algorithm on the data we collect in the database to get the best result and return the similar webpages together to enhance the search for webpages.

6.1 Webpage representation using Graph (Data structure)

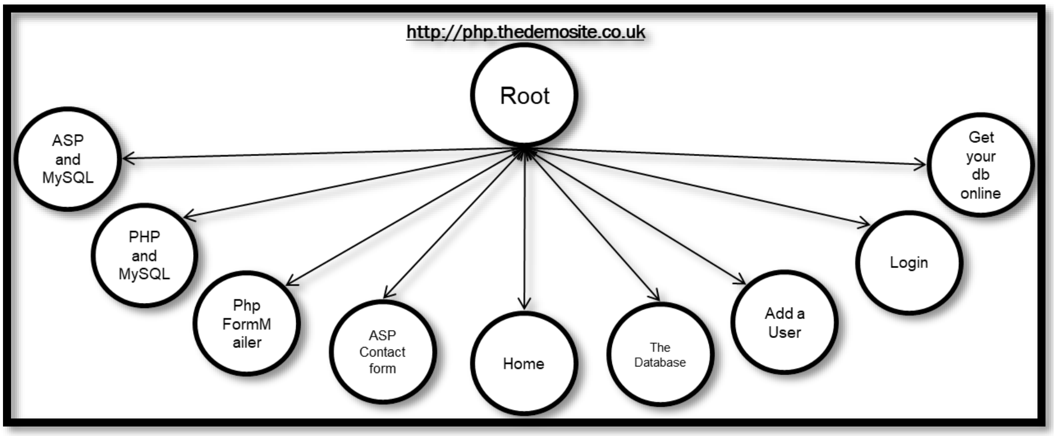


Figure 7 - Webpage representation as Graph

6.2 Webpage representation using Contexts

Extract the most frequently word that are exists, Keywords: - [key-1, key-2, key-3, .., key-k].

Table 5 - Webpage representation as Contexts

Tags	Key-1	Key-2	Key-3	Key-k
Website 1	**Most frequent word 1**	**Most frequent word 2**	**Most frequent word 3**	n
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
Website n	n	n	n	n

6.2.1 Database of Webpages

Here is the structure of database (represent webpages as DB).

Table 6 - Webpage as DB

Id_web (Pk)	Tag name (string)	Link (string)	Keyword-1 (int)	Keyword-2 (int)	Keyword-n (int)
1	<str>	<str>	<str>	<str>	<str>
2	<str>	<str>	<str>	<str>	<str>
3	**FREE example PHP code and online MySQL database**	**thedemosite.co.uk**	**5**	**10**	**20**
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
n	n	n	n	n	n

6.3 Web Pages Categorization

This project will develop the way of categorization of web pages using K-means and other methods through extracting the features automatically. Here eight major categories of web pages will be selected for categorization; these are business & economy, education, government, entertainment, sports, news & media, job search, and science.

Automated categorization of web pages can lead to better web retrieval tools with the added convenience of selecting among properly organized directories. Web page classification is proposed which extract the features automatically through analyzing the html source and categorize the web pages into eight major classes.

Chapter 7: Methodology for Results Implementation

7.1 Decision tree

Is a classification and structured based approach which consist of root node, branches and leaf nodes. It is hierarchical process in which root node is split into sub-branches and leaf node contains class label.

7.2 K-means

K-means algorithm is an iterative algorithm that tries to partition the dataset into clusters, where each data point belongs to only one group. It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic means of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

$$\boxed{D(a,b)=\sqrt{\sum_{i=1}^{n}(b_i-a_i)^2}}$$

Figure 8 - Euclidean distance

The way K-means works is as follows:

1) Specify number of clusters K.
2) Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
3) Keep iterating until there is no change to the centroids.
4) Compute the sum of the squared distance (we used Euclidean distance **Fig.8** as a metric of similarity for our data set) between data points and all centroids.
5) Assign each data point to the closest cluster (centroid).
6) Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

7.3 Hierarchical clustering

In data mining and statistics, hierarchical is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

1) **Agglomerative**(bottom-up): each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
2) **Divisive**(top-down): all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

In general, the merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented in a dendrogram(**Fig.**).

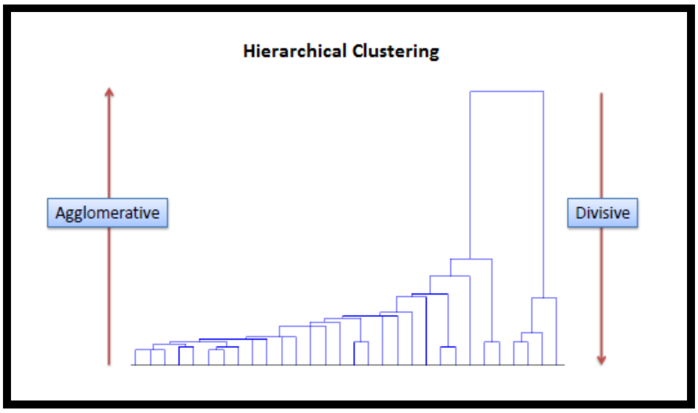
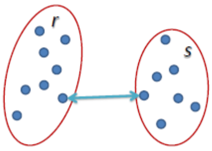
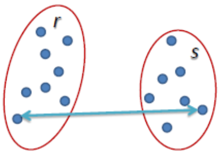
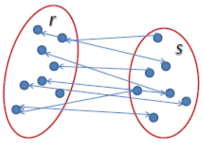


Figure 9 - Hierarchical clustering

Before any clustering is performed, it is required to determine the proximity matrix containing the distance between each point using a distance function. Then, the matrix is updated to display the distance between each cluster. The following three methods differ in how the distance between each cluster is measured.

Table 7 - Proximity matrix type

Single Linkage	**Complete Linkage**	**Average Linkage**
 $L(r,s)=\min(D(x_{ri},x_{sj}))$	 $L(r,s)=\max(D(x_{ri},x_{sj}))$	 $L(r,s)=\frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_s} D(x_{ri},x_{sj})$
The distance between two clusters is defined as the shortest distance between two points in each cluster.	The distance between two clusters is defined as the longest distance between two points in each cluster.	The distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.

7.4 Artificial Neural network (ANN)

Artificial Neural Network (ANN) models were called parallel distributed processing models. An ANN mimics the human brain's biological neural network. The biological neural network is the mechanism through which a living organism's nervous system functions, enabling complex tasks to be performed instinctively. The central processing unit of that nervous system is known as a "**neuron**". The human brain has around 10 to 100 billion neurons, each connected to many others by "**synapses**". The human brain has around 100 trillion synapses. These connections control the human body and its thought processes. In short, they attempt to replicate the learning processes of the human brain. It's another web content mining approach which use artificial neural networks (ANNs) with supervised and unsupervised classification. The back-propagation learning algorithm is famous method for supervised ANNs. A NN consists of multiple layers i.e. **input layer**, some **hidden layers** and then **output layer**, each feeds the next layer till last layer (output) [3].

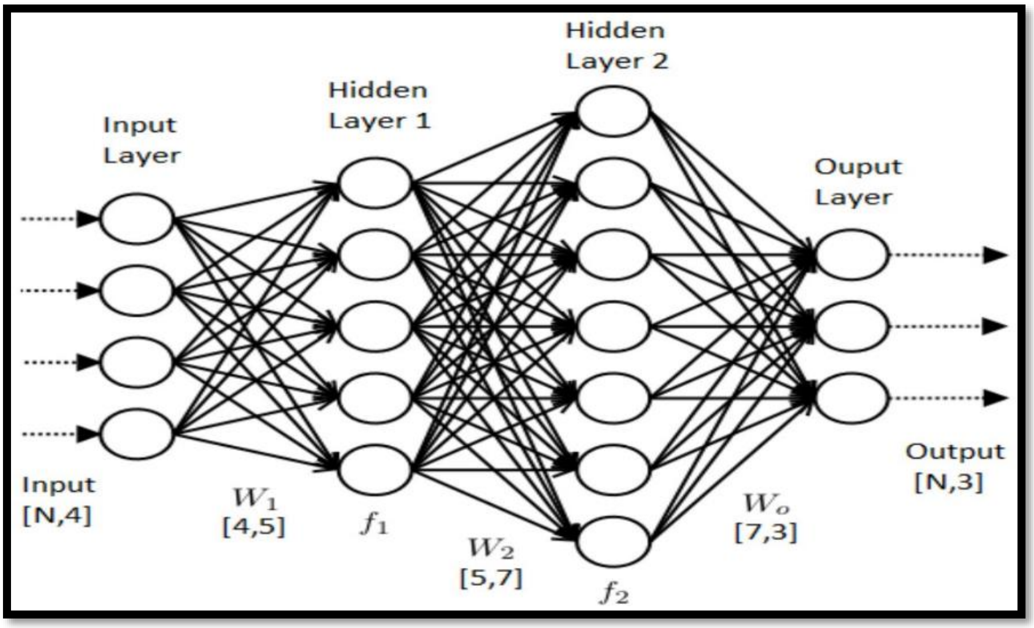


Figure 10 - Artificial Neural network

Chapter 8: Webpage clustering

Web page clustering is one of the major and most important preprocessing steps in web mining. In this chapter web context mining items to be studied are web pages. Web page clustering puts together web pages into groups, based on similarity or other relationship measures. Tightly-couple pages, pages in the same cluster, are considered as singular items for following steps. A complete data mining analysis could be performed by using web pages information as it appears in web logs, but when the number of pages to take into account increases (i.e., in a corporative largescale web server or a server using dynamic web pages) this process could be quite hard or even unbearable. In order to deal with this issue, web page clustering appears as a reasonable solution. These techniques group pages together based on relationship measure.

8.1 Algorithms

In this project we used two types of clustering algorithms to cluster our data, **K-means** (partitional) and **Agglomerative** (hierarchical) clustering.

- A Partitional clustering a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.
- A Hierarchical clustering is a set of nested clusters that are organized as a tree.

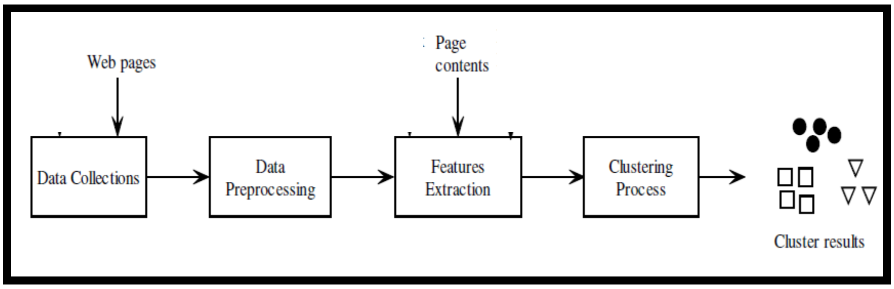


Figure 11 - General Process of Web Page Clustering

1) **Data collection:** Data are being collected in form of original web page addresses.
2) **Preprocessing:** It's done on collected data to transform it into appropriate form so that it can be efficiently used in clustering process.
3) **Feature extraction:** Features are extracted in form of page contents(ex. words or terms are extracted as features.)
4) **Clustering:** Algorithms are applied on extracted features and cluster result is obtained. Based on feature extraction we categorize web page in Text-based approach.

8.1.1 Text-based approach.

Web pages are treated as a text documents that is all HTML and XML tags, special characters and images are removed from the web pages so web pages would be converted into normal documents.

Here web documents are clustered according to their contents. As clustering algorithms cannot work on text directly so documents must be presented in correct format. Most widely used text document representation is based on vector space model (VSM) [14]. Each document is represented by a vector of number of frequent of the extracted features from the document. Consider the vector for i^{th} document $d_i = (f_{1i\times j}, f_{2i\times j}, f_{3i\times j}, . . . ,f_{mi\times j})$. Here ‘m’ features are extracted, ‘i’ is the number of frequencies of ‘j’ in the document, and ‘j’ is the extracted future.

8.1.2 K-means Clustering

In this part we used K-means algorithm to cluster webpages into 3 different clusters, and we apply it on small dataset which we will be shown within the whole processes on how the data being gathered through preprocessing until applying the k-means and get its results. The processes are separated into two main processes for all the project **Importing and Filtering** and one for algorithm process **K-means** or any other algorithm.

8.1.2.1 Importing

With help of MATLAB software, we were able to extract what we want from any website easily and more efficient with excel DB, because of how MATLAB and Excel sharing the same structure of storing data into cells. At this part we developed a script (.m file) that will start by reading the .xlsx DB file and imported inside the workstation as the variables (num txt raw),were ‘num’ contains only the numbers from the xlsx, ‘txt’ only contains the strings, and the ‘raw’ will have the whole xlsx DB. then it will ask for any web address as an input after that it will start to analyze the website through the script and extract the most 3 frequent words in the entered website, and add it with its frequency to ‘raw’, and it will assign the frequency value for any exists keywords with ‘0’. After finishing the websites insertion and ask it to stop getting input. It will start to refresh the whole values for all keywords inside either old or new, and will give check the old keywords that we assign it as ‘0’ if its exists inside each website content or not and if it’s their automatically it will get the frequency value for that word and put it inside ‘raw’. After the refreshing process finish ‘raw’ will be wrote inside the xlsx file and finish the insertion process. At the end of this process the output should be as mentioned before in (**Table.6**).

- Enter websites one by one to extract the 3 most frequent words, ‘1’ to stop and extract the result into excel DB, or ‘2’ to refresh the whole DB with new or old words and its frequencies In order to start the second step for filtering the data from unwanted and noise data. “You must refresh the DB at the end of to make sure everything is correct”.

8.1.2.1.1 Importing Flaw chart

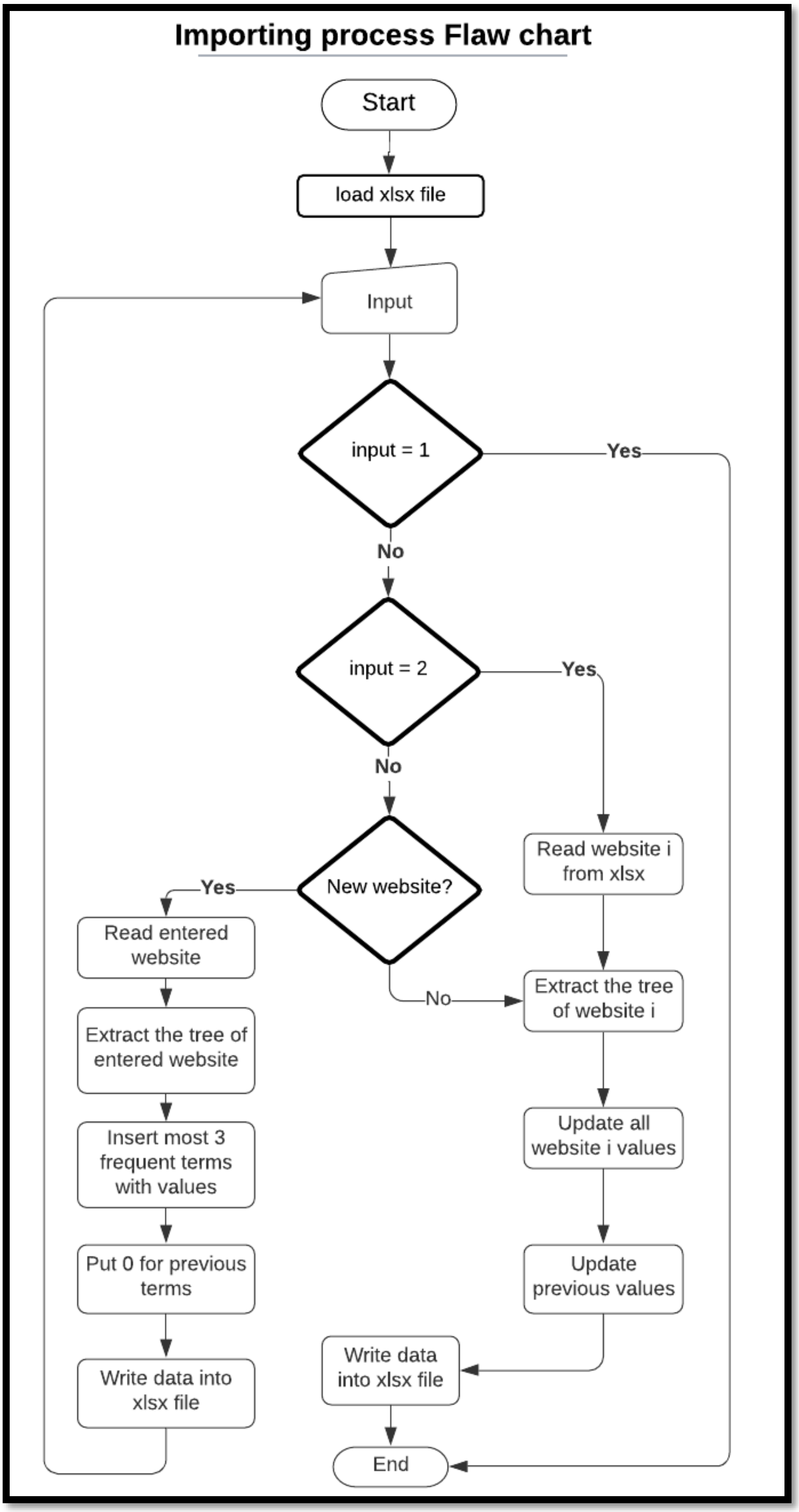
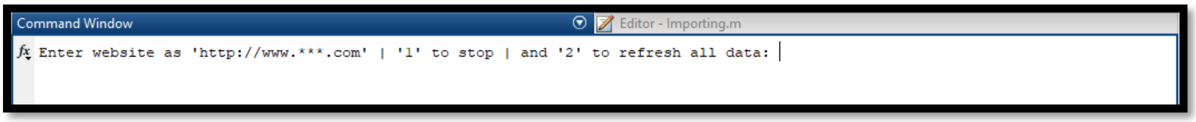

Importing process Flaw chart
Start
load xlsx file
Input
input = 1
Yes
No
input = 2
Yes
No
New website?
Yes
No
Read website i from xlsx
Read entered website
Extract the tree of website i
Extract the tree of entered website
Update all website i values
Insert most 3 frequent terms with values
Put 0 for previous terms
Update previous values
Write data into xlsx file
Write data into xlsx file
End

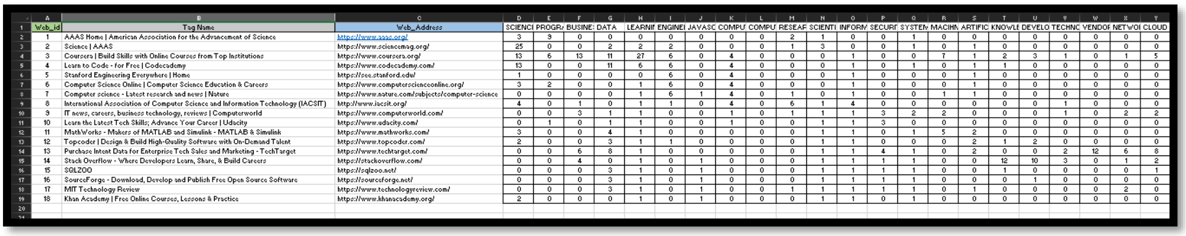
Figure 12 - Importing Flaw chart

8.1.2.1.2 Script result



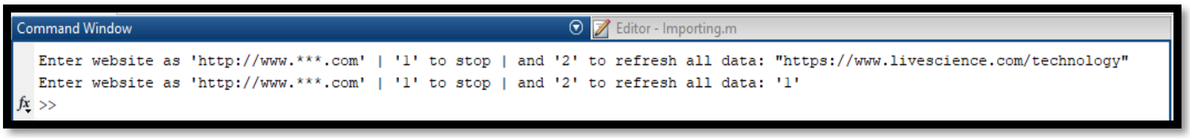
```
Enter website as 'http://www.***.com' | '1' to stop | and '2' to refresh all data:
```

Figure 13 - importing script



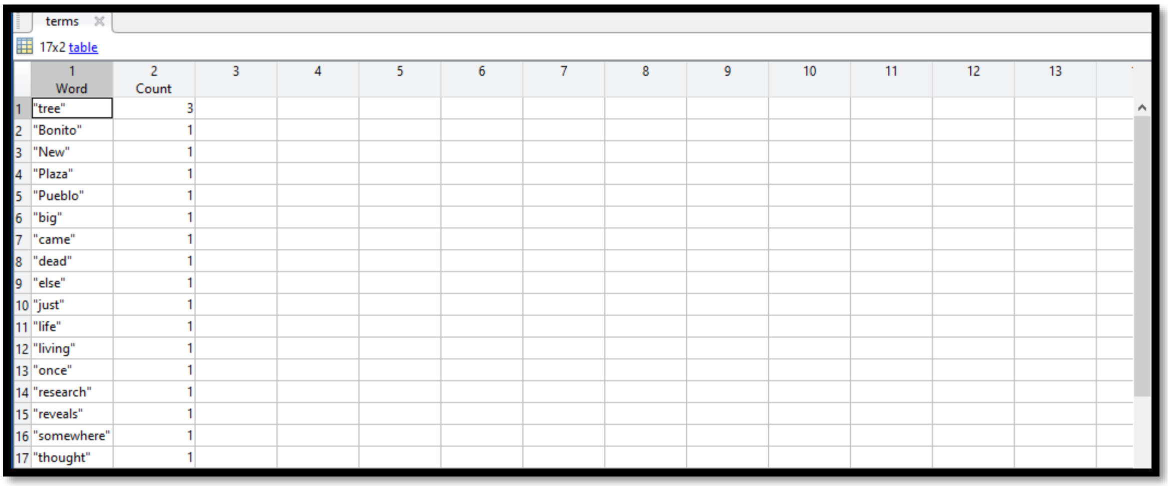
Web_id	Tag Name	Web_Address	SCIENCI	PROGR/	BUSINE:	DATA	LEARNII	ENGINEI	JAVASC	COMPU	COMPU	RESEAF	SCIENTI	INFORM	SECURI	SYSTEM	MACIHN	ARTIFIC	KNOWLE	DEVELO	TECHNC	VENDOF	NETWOI	CLOUD
1	AAAS Home | American Association for the Advancement of Science	https://www.aaas.org/	3	9	0	0	0	0	0	0	0	2	1	0	0	1	0	0	0	0	0	0	0	0
2	Science | AAAS	https://www.sciencemag.org/	25	0	0	2	2	2	0	0	0	1	3	0	0	1	0	1	0	0	0	0	0	0
3	Coursera | Build Skills with Online Courses from Top Institutions	https://www.coursera.org/	13	6	13	11	27	6	0	4	0	0	0	1	0	0	7	1	2	3	1	0	1	5
4	Learn to Code - for Free | Codecademy	https://www.codecademy.com/	13	0	0	11	6	6	0	4	0	0	0	1	0	1	0	1	1	0	0	0	0	0
5	Stanford Engineering Everywhere | Home	https://see.stanford.edu/	1	0	0	0	0	6	0	4	0	0	0	1	0	0	0	1	0	0	0	0	0	0
6	Computer Science Online | Computer Science Education & Careers	https://www.computerscienceonline.org/	3	2	0	0	1	6	0	4	0	0	0	1	0	0	0	1	0	0	0	0	0	0
7	Computer science - Latest research and news | Nature	https://www.nature.com/subjects/computer-science	0	0	0	0	1	6	1	4	0	1	0	1	0	0	1	1	0	0	0	0	0	0
8	International Association of Computer Science and Information Technology (IACSIT)	http://www.iacsit.org/	4	0	1	0	1	1	0	4	0	6	1	4	0	0	0	0	0	0	1	0	0	0
9	IT news, careers, business technology, reviews | Computerworld	https://www.computerworld.com/	0	0	3	1	1	0	0	1	0	1	1	1	3	2	2	0	0	0	1	0	2	2
10	Learn the Latest Tech Skills; Advance Your Career | Udacity	https://www.udacity.com/	0	1	0	1	1	0	0	1	0	0	1	1	3	0	0	0	0	0	0	0	0	0
11	MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink	https://www.mathworks.com/	3	0	0	4	1	0	0	0	0	0	1	1	0	1	5	2	0	0	0	0	0	0
12	Topcoder | Design & Build High-Quality Software with On-Demand Talent	https://www.topcoder.com/	2	0	0	3	1	1	0	0	0	0	1	1	0	0	0	2	1	2	0	0	0	0
13	Purchase Intent Data for Enterprise Tech Sales and Marketing - TechTarget	https://www.techtarget.com/	0	0	6	8	1	0	0	0	0	0	1	1	4	1	0	2	0	0	2	12	6	8
14	Stack Overflow - Where Developers Learn, Share, & Build Careers	https://stackoverflow.com/	0	0	4	0	1	0	1	0	0	0	1	1	1	1	0	0	12	10	3	0	1	2
15	SQLZOO	https://sqlzoo.net/	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0	0	1
16	SourceForge - Download, Develop and Publish Free Open Source Software	https://sourceforge.net/	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0
17	MIT Technology Review	https://www.technologyreview.com/	0	0	0	3	1	0	1	0	0	1	1	1	1	1	0	1	0	0	0	0	2	0
18	Khan Academy | Free Online Courses, Lessons & Practice	https://www.khanacademy.org/	2	0	0	0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0

Figure 14 - DB before adding or refresh



```
Enter website as 'http://www.***.com' | '1' to stop | and '2' to refresh all data: "https://www.livescience.com/technology"
Enter website as 'http://www.***.com' | '1' to stop | and '2' to refresh all data: '1'
>>
```

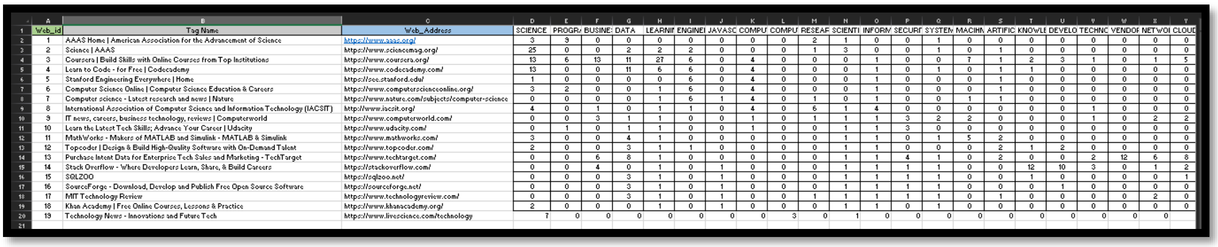
Figure 15 - Inserting "LiveScience.com" then '1' for finish



	Word	Count
1	"tree"	3
2	"Bonito"	1
3	"New"	1
4	"Plaza"	1
5	"Pueblo"	1
6	"big"	1
7	"came"	1
8	"dead"	1
9	"else"	1
10	"just"	1
11	"life"	1
12	"living"	1
13	"once"	1
14	"research"	1
15	"reveals"	1
16	"somewhere"	1
17	"thought"	1

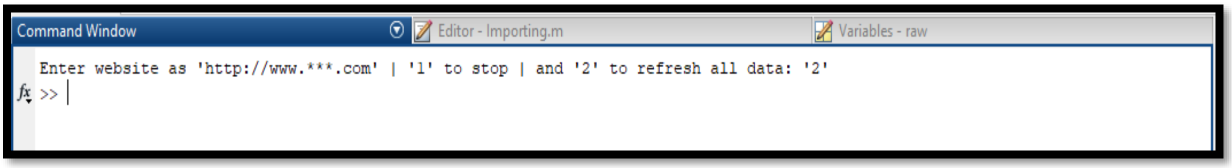
Figure 16 - Words and its frequency in entered website

We can see here that the website has been added successfully, with its keywords, and each pervious keyword assigned with 0s.



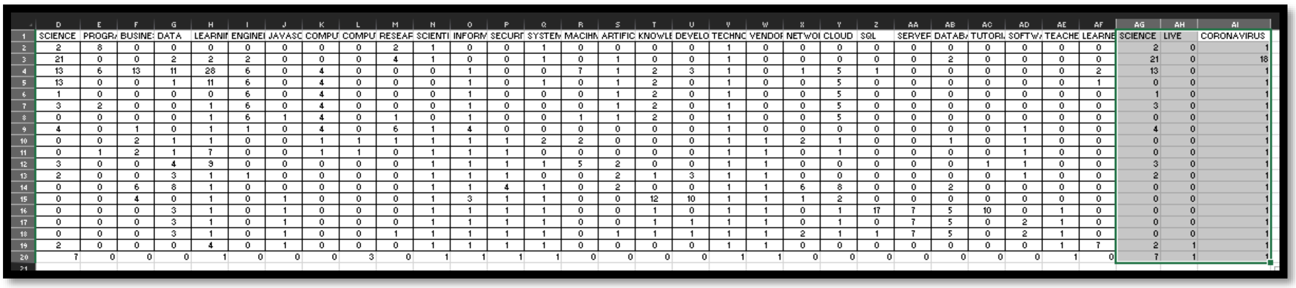
Web_id	Tag Name	Web_Address	SCIENCE	PROGR/	BUSINE:	DATA	LEARNII	ENGINEI	JAVASC	COMPU	COMPU	RESEAF	SCIENTI	INFORM	SECURI	SYSTEM	MACIHN	ARTIFIC	KNOWLE	DEVELO	TECHNC	VENDOF	NETWOI	CLOUD
1	AAAS Home | American Association for the Advancement of Science	https://www.aaas.org/	3	9	0	0	0	0	0	0	0	2	1	0	0	1	0	0	0	0	0	0	0	0
2	Science | AAAS	https://www.sciencemag.org/	25	0	0	2	2	2	0	0	0	1	3	0	0	1	0	1	0	0	0	0	0	0
3	Coursera | Build Skills with Online Courses from Top Institutions	https://www.coursera.org/	13	6	13	11	27	6	0	4	0	0	0	1	0	0	7	1	2	3	1	0	1	5
4	Learn to Code - for Free | Codecademy	https://www.codecademy.com/	13	0	0	11	6	6	0	4	0	0	0	1	0	1	0	1	1	0	0	0	0	0
5	Stanford Engineering Everywhere | Home	https://see.stanford.edu/	1	0	0	0	0	6	0	4	0	0	0	1	0	0	0	1	0	0	0	0	0	0
6	Computer Science Online | Computer Science Education & Careers	https://www.computerscienceonline.org/	3	2	0	0	1	6	0	4	0	0	0	1	0	0	0	1	0	0	0	0	0	0
7	Computer science - Latest research and news | Nature	https://www.nature.com/subjects/computer-science	0	0	0	0	1	6	1	4	0	1	0	1	0	0	1	1	0	0	0	0	0	0
8	International Association of Computer Science and Information Technology (IACSIT)	http://www.iacsit.org/	4	0	1	0	1	1	0	4	0	6	1	4	0	0	0	0	0	0	1	0	0	0
9	IT news, careers, business technology, reviews | Computerworld	https://www.computerworld.com/	0	0	3	1	1	0	0	1	0	1	1	1	3	2	2	0	0	0	1	0	2	2
10	Learn the Latest Tech Skills; Advance Your Career | Udacity	https://www.udacity.com/	0	1	0	1	1	0	0	1	0	0	1	1	3	0	0	0	0	0	0	0	0	0
11	MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink	https://www.mathworks.com/	3	0	0	4	1	0	0	0	0	0	1	1	0	1	5	2	0	0	0	0	0	0
12	Topcoder | Design & Build High-Quality Software with On-Demand Talent	https://www.topcoder.com/	2	0	0	3	1	1	0	0	0	0	1	1	0	0	0	2	1	2	0	0	0	0
13	Purchase Intent Data for Enterprise Tech Sales and Marketing - TechTarget	https://www.techtarget.com/	0	0	6	8	1	0	0	0	0	0	1	1	4	1	0	2	0	0	2	12	6	8
14	Stack Overflow - Where Developers Learn, Share, & Build Careers	https://stackoverflow.com/	0	0	4	0	1	0	1	0	0	0	1	1	1	1	0	0	12	10	3	0	1	2
15	SQLZOO	https://sqlzoo.net/	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0	0	1
16	SourceForge - Download, Develop and Publish Free Open Source Software	https://sourceforge.net/	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0
17	MIT Technology Review	https://www.technologyreview.com/	0	0	0	3	1	0	1	0	0	1	1	1	1	1	0	1	0	0	0	0	2	0
18	Khan Academy | Free Online Courses, Lessons & Practice	https://www.khanacademy.org/	2	0	0	0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
19	Technology News - Innovations and Future Tech	https://www.livescience.com/technology	7	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	

Figure 17 - DB after inserting



```
Enter website as 'http://www.***.com' | '1' to stop | and '2' to refresh all data: '2'
>>
```

Figure 18 - Refreshing DB after finish inserting



	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
1	SCIENCE	PROGRA	BUSINE	DATA	LEARNI	ENGINEI	JAVASC	COMPU	COMPU	RESEAF	SCIENTI	INFORM	SECURI	SYSTEM	MACIHN	ARTIFIC	KNOWLE	DEVELO	TECHNC	VENDOF	NETWOI	CLOUD	SQL	SERVEF	DATABA	TUTORI	SOFTWA	TEACHE	LEARNE	SCIENCE	LIVE	CORONAVIRUS
2	2	8	0	0	0	0	0	0	0	2	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	1
3	21	0	0	2	2	2	0	0	0	4	1	0	0	1	0	1	0	0	1	0	0	0	0	0	2	0	0	0	0	21	0	18
4	13	6	13	11	28	6	0	4	0	0	0	1	0	0	7	1	2	3	1	0	1	5	1	0	0	0	0	0	2	13	0	1
5	13	0	0	1	11	6	0	4	0	0	0	1	0	1	0	1	2	0	1	0	0	5	0	0	0	0	0	0	1	0	0	1
6	1	0	0	0	0	6	0	4	0	0	0	1	0	0	0	1	2	0	1	0	0	5	0	0	0	0	0	0	0	1	0	1
7	3	2	0	0	1	6	0	4	0	0	0	1	0	0	0	1	2	0	1	0	0	5	0	0	0	0	0	0	0	3	0	1
8	0	0	0	0	1	6	1	4	0	1	0	1	0	0	1	1	2	0	1	0	0	5	0	0	0	0	0	0	0	0	0	1
9	4	0	1	0	1	1	0	4	0	6	1	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	4	0	1
10	0	0	2	1	1	0	0	1	1	1	1	1	1	2	2	0	0	0	1	1	2	1	0	0	1	0	1	0	0	0	0	1
11	0	1	2	1	7	0	0	1	1	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	1
12	3	0	0	4	9	0	0	0	0	0	1	1	1	1	5	2	0	0	1	1	0	0	0	0	0	1	1	0	0	3	0	1
13	2	0	0	3	1	1	0	0	0	0	1	1	1	0	0	2	1	3	1	1	0	0	0	0	0	0	1	0	0	2	0	1
14	0	0	6	8	1	0	0	0	0	0	1	1	4	1	0	2	0	0	1	1	6	8	0	0	2	0	0	0	0	0	0	1
15	0	0	4	0	1	0	1	0	0	0	1	3	1	1	0	0	12	10	1	1	1	2	0	0	0	0	0	0	0	0	0	1
16	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	17	7	5	10	0	1	0	0	0	1
17	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	7	5	0	2	1	0	0	0	1
18	0	0	0	3	1	0	1	0	0	1	1	1	1	1	0	1	1	1	1	1	2	1	1	7	5	0	2	1	0	0	0	1
19	2	0	0	0	4	0	1	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	7	2	1	1
20	7	0	0	0	1	0	0	0	3	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	7	1	1

Figure 19 - DB after refreshing all the data with new keywords

8.2.2.1.3 Script Code

```
% Clear everything before start
clear all;
clc;
path1 = 'C:\Users\xxmoh\Desktop\GP2\Presentation\Scripts\1 -
Importing\web_dataset.xlsx';

[num, txt, raw] = xlsread(path1); %Read from xlsx files.
selector = "title";

while(true)
     %_____________________Reading and extracting_____________________%

%Read website from user
    user_in = input("Enter website as 'http://www.***.com' | '1' to stop |
and '2' to refresh all data: ", 's');

    if(user_in == "1")
        return; %Stop entering
    end
          %_____________________Refresh data_____________________%

  if (user_in == "2") %Refresh all data
      for e=1:1:size(raw,1)-1
            web_url = string(raw(e+1,3));%take each website with every itr.

            url = webread(web_url); %Read the website
            tree = htmlTree(url); %Get the tree of website

            term = extractHTMLText(tree);%Get text from website

            tag_name = extractHTMLText(findElement(tree,selector));
            terms = wordCloudCounts(term); %Extract the Frequency
```

```
         %Compare address for each website in db and get the index for it
          [row , col] = size(raw);
          counter = 1;
          in(1,3) = 0;
          find_site = strcmp(raw(:,3), web_url);
          site_indx = find(find_site == 1);

          %Compare old terms with all web terms and get indexes
          for i=1:1:size(terms.Word,1)
              for j=4:1:col
                  if upper(terms.Word(i)) == txt(1,j:j)
                      in(counter,1) = terms.Count(i);
                      in(counter,2) = j;
                      counter = counter + 1;
                  end
              end
          end

          %Check if there's same terms and insert or not then write
          if(in(1,1:3) == 0)
              continue;
          else
              insert_new = size(in,1);
              for i=1:1:insert_new
                  raw(site_indx ,in(i,2)) = {in(i,1)};
              end
          end
    end
    xlswrite(path1, raw);
    return;
end
      %_____________________Entered website_____________________%

  url = webread(user_in);
  tree = htmlTree(url);
  term = extractHTMLText(tree);
  tag_name = extractHTMLText(findElement(tree,selector));
  terms = wordCloudCounts(term);
  [row , col] = size(raw);

%Check input if exsist or not using string logical comparison 1 or 0
  p = strcmp(raw(2:end,3), {user_in});
  s = find(p == 1);
  %New site
  if (isempty(s))
      raw(end+1,1) = {row};
      raw(end,2) = {tag_name(1,1)};
      raw(end,3) = {user_in};

      %Add new terms and it's values
      for i=1:1:3
          [row , col] = size(raw);
          raw(1, col+1)= {upper(terms.Word(i,1))}';
          raw(end,end) = {terms.Count(i,1)}';
      end
      %Give empty cells 0s
      empty_cell = cellfun('isempty',raw);
      raw(empty_cell) = {0};
      counter = 1;
      in(1,3) = 0;
```

```
        %Compare old terms with web terms and get indexes if exists
        for i=1:1:size(terms.Word,1)
            for j=4:1:row
                if upper(terms.Word(i)) == txt(1,j:j)
                    in(counter,1) = terms.Count(i);
                    in(counter,2) = j;
                    counter = counter + 1;
                    in(1,3) = 1;
                else
                    in(1,3) = 0;
                end
            end
        end

        %Check if there's same terms and insert or not then write
        if(in(1,1:3) == 0)
            xlswrite(path1, raw);
        else
            insert_new = size(in,1);
            for i=1:1:insert_new
                raw(end,in(i,2)) = {in(i,1)};
            end
            %xlswrite(path1, raw);
        end
%_______________________________Exists site_____________________________%
    else
        counter = 1;
        in(1,3) = 0;
        find_site = strcmp(raw(:,3), user_in);
        site_indx = find(find_site == 1);
        tag_name = extractHTMLText(findElement(tree,selector));
        raw(site_indx,2) = {tag_name(1,1)};

        %Compare old terms with web terms and get indexes
        for i=1:1:size(terms.Word,1)
            for j=3:1:row
                if upper(terms.Word(i)) == txt(1,j:j)
                    in(counter,1) = terms.Count(i);
                    in(counter,2) = j;
                    counter = counter + 1;
                    in(1,3) = 1;
                else
                    in(1,3) = 0;
                end
            end
        end
        %Check if there's same terms and insert or not then write
        insert_new = size(in,1);
        for i=1:1:insert_new
            raw(site_indx ,in(i,2)) = {in(i,1)};
        end
        xlswrite(path1, raw);
    end
end
```

8.1.2.2 Filtering

The filtering process we used starts by taking the output of the pervious step(Importing process) and start to filter it by delete the duplicated words first from entire DB, then the script starts to filter the data according to the average of the sum of word frequency for all keywords and accept only the frequency value that is greater than the average and reject and delete any keywords that are less than the average, in order to not make the data fill with noise, and unwanted values. Then extract it to the excel DB. file to be ready for the Final step after preprocessing and feature extraction.

8.1.2.2.1 Filtering Flaw chart

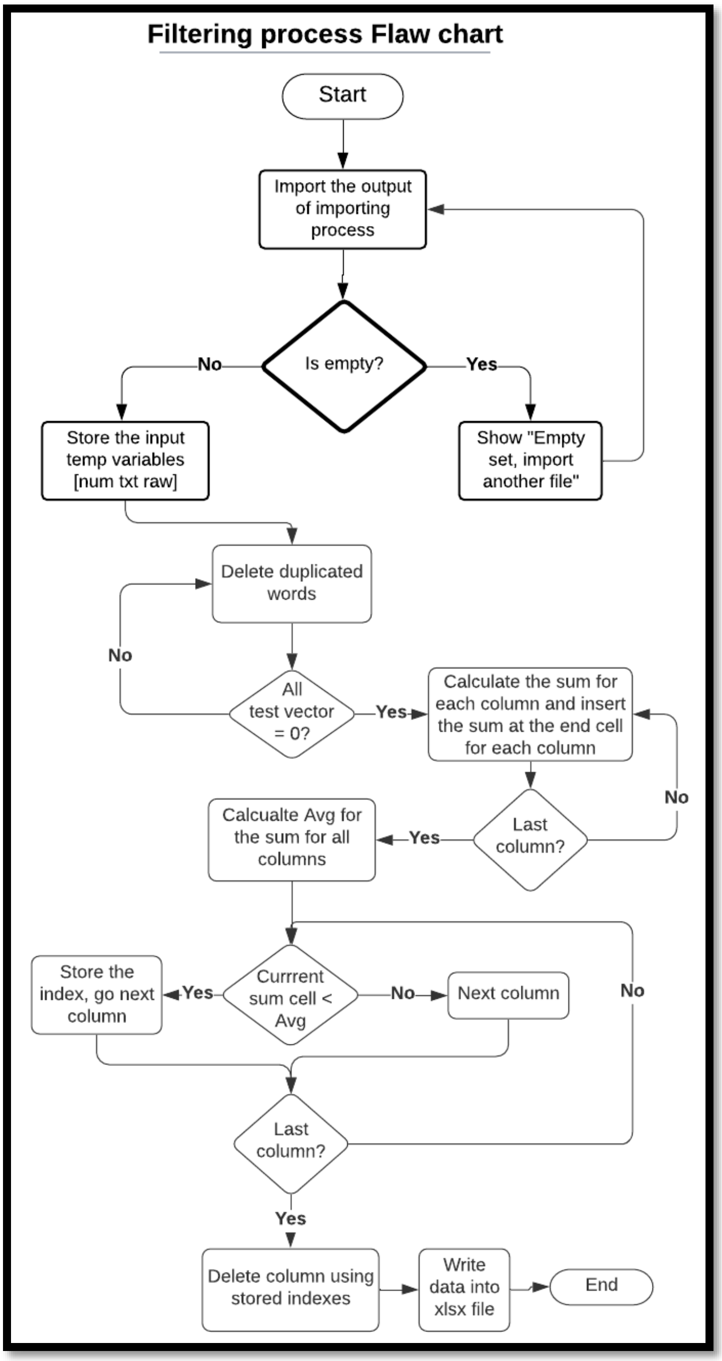
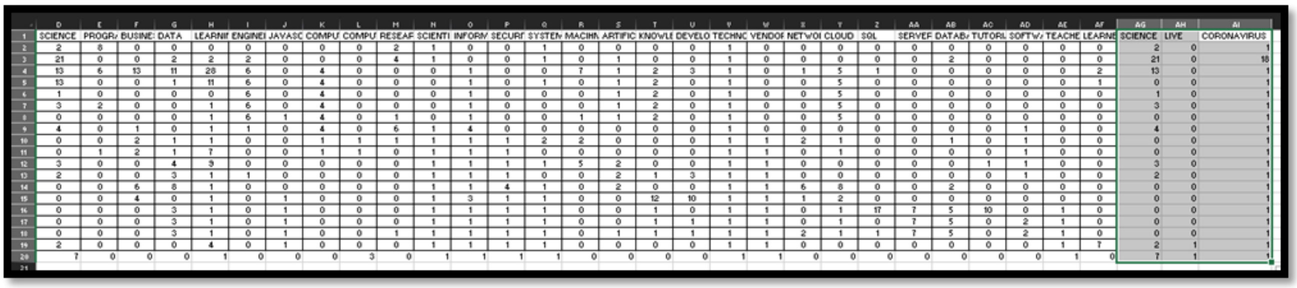

Filtering process Flaw chart
Start
Import the output of importing process
Is empty?
No
Yes
Store the input temp variables [num txt raw]
Show "Empty set, import another file"
Delete duplicated words
No
All test vector = 0?
Yes
Calculate the sum for each column and insert the sum at the end cell for each column
No
Calcualte Avg for the sum for all columns
Yes
Last column?
Store the index, go next column
Yes
Currrent sum cell < Avg
No
Next column
No
Last column?
Yes
Delete column using stored indexes
Write data into xlsx file
End

Figure 20 - Filtering Flaw chart

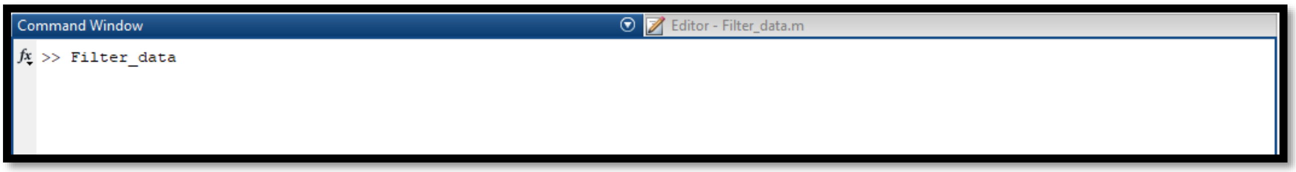
8.1.2.2.2 Script result

The expected result from this step, is the DB will be filtered from unwanted, noise data. So, it becomes more efficient for use in our project and make it better data to give better result.



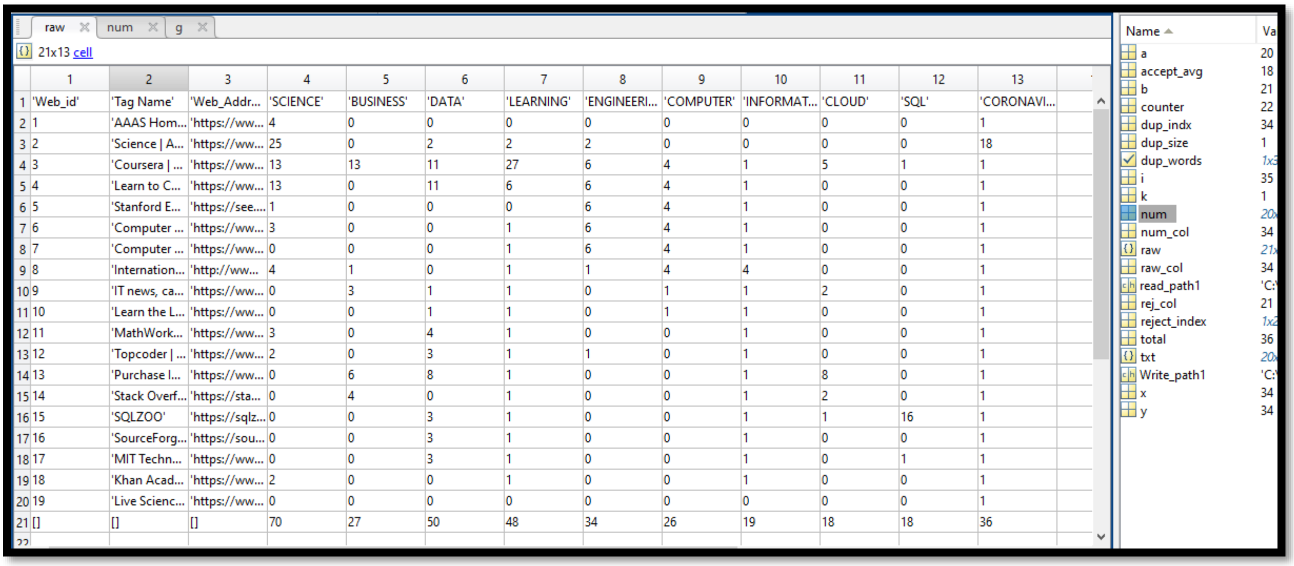
	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
1	SCIENCE	PROGR/	BUSINE:	DATA	LEARNII	ENGINEI	JAVASC	COMPU'	COMPU'	RESEAF	SCIENTI	INFORM	SECURI	SYSTEM	MACHIN	ARTIFIC	KNOWLE	DEVELO	TECHNC	VENDOF	NETWOI	CLOUD	SQL	SERVEF	DATAB/	TUTORI	SOFTW/	TEACHE	LEARNI	SCIENCE	LIVE	CORONAVIRUS
2	2	8	0	0	0	0	0	0	0	2	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	1
3	21	0	0	2	2	2	0	0	0	4	1	0	0	1	0	1	0	0	1	0	0	0	0	0	2	0	0	0	0	21	0	18
4	13	6	13	11	28	6	0	4	0	0	0	1	0	0	7	1	2	3	1	0	1	5	1	0	0	0	0	0	2	13	0	1
5	13	0	0	1	11	6	0	4	0	0	0	1	0	1	0	1	2	0	1	0	0	5	0	0	0	0	0	0	1	0	0	1
6	1	0	0	0	0	6	0	4	0	0	0	1	0	0	0	1	2	0	1	0	0	5	0	0	0	0	0	0	0	1	0	1
7	3	2	0	0	1	6	0	4	0	0	0	1	0	0	0	1	2	0	1	0	0	5	0	0	0	0	0	0	0	3	0	1
8	0	0	0	0	1	6	1	4	0	1	0	1	0	0	1	1	2	0	1	0	0	5	0	0	0	0	0	0	0	0	0	1
9	4	0	1	0	1	1	0	4	0	6	1	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	4	0	1
10	0	0	2	1	1	0	0	1	1	1	1	1	1	2	2	0	0	0	1	1	2	1	0	0	1	0	1	0	0	0	0	1
11	0	1	2	1	7	0	0	1	1	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	1
12	3	0	0	4	9	0	0	0	0	0	1	1	1	1	5	2	0	0	1	1	0	0	0	0	0	1	1	0	0	3	0	1
13	2	0	0	3	1	1	0	0	0	0	1	1	1	0	0	2	1	3	1	1	0	0	0	0	0	0	1	0	0	2	0	1
14	0	0	6	8	1	0	0	0	0	0	1	1	4	1	0	2	0	0	1	1	6	8	0	0	2	0	0	0	0	0	0	1
15	0	0	4	0	1	0	1	0	0	0	1	3	1	1	0	0	12	10	1	1	1	2	0	0	0	0	0	0	0	0	0	1
16	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	17	7	5	10	0	1	0	0	0	1
17	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	7	5	0	2	1	0	0	0	1
18	0	0	0	3	1	0	1	0	0	1	1	1	1	1	0	1	1	1	1	1	2	1	1	7	5	0	2	1	0	0	0	1
19	2	0	0	0	4	0	1	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	7	2	1	1
20	7	0	0	0	1	0	0	0	3	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	7	1	1
21																																

Figure 21 - Last update for DB



```
>> Filter_data
```

Figure 22 - Call the script



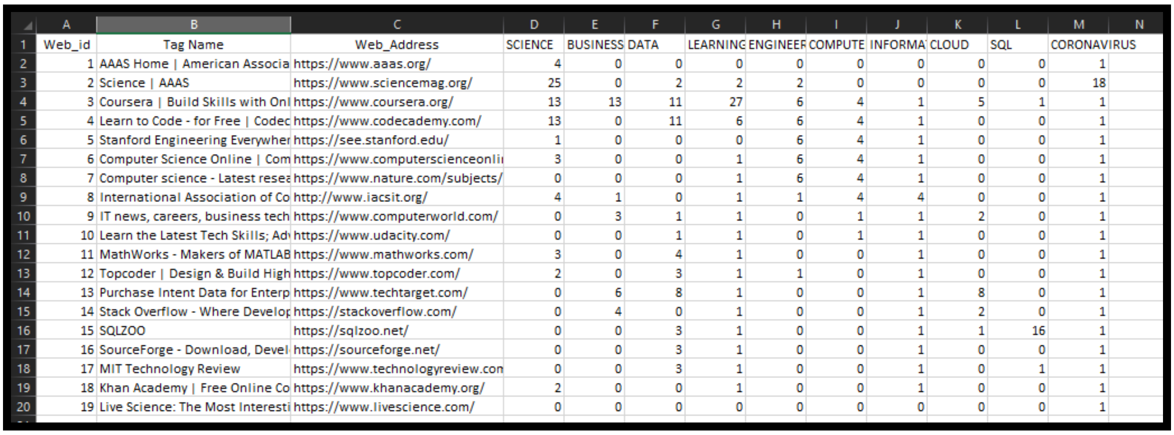
21x13 cell

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	'Web_id'	'Tag Name'	'Web_Addr...	'SCIENCE'	'BUSINESS'	'DATA'	'LEARNING'	'ENGINEERI...	'COMPUTER'	'INFORMAT...	'CLOUD'	'SQL'	'CORONAVI...
2	1	'AAAS Hom...	'https://ww...	4	0	0	0	0	0	0	0	0	1
3	2	'Science | A...	'https://ww...	25	0	2	2	2	0	0	0	0	18
4	3	'Coursera | ...	'https://ww...	13	13	11	27	6	4	1	5	1	1
5	4	'Learn to C...	'https://ww...	13	0	11	6	6	4	1	0	0	1
6	5	'Stanford E...	'https://see....	1	0	0	0	6	4	1	0	0	1
7	6	'Computer ...	'https://ww...	3	0	0	1	6	4	1	0	0	1
8	7	'Computer ...	'https://ww...	0	0	0	1	6	4	1	0	0	1
9	8	'Internation...	'http://ww...	4	1	0	1	1	4	4	0	0	1
10	9	'IT news, ca...	'https://ww...	0	3	1	1	0	1	1	2	0	1
11	10	'Learn the L...	'https://ww...	0	0	1	1	0	1	1	0	0	1
12	11	'MathWork...	'https://ww...	3	0	4	1	0	0	1	0	0	1
13	12	'Topcoder | ...	'https://ww...	2	0	3	1	1	0	1	0	0	1
14	13	'Purchase I...	'https://ww...	0	6	8	1	0	0	1	8	0	1
15	14	'Stack Overf...	'https://sta...	0	4	0	1	0	0	1	2	0	1
16	15	'SQLZOO'	'https://sqlz...	0	0	3	1	0	0	1	1	16	1
17	16	'SourceForg...	'https://sou...	0	0	3	1	0	0	1	0	0	1
18	17	'MIT Techn...	'https://ww...	0	0	3	1	0	0	1	0	1	1
19	18	'Khan Acad...	'https://ww...	2	0	0	1	0	0	1	0	0	1
20	19	'Live Scienc...	'https://ww...	0	0	0	0	0	0	0	0	0	1
21	[]	[]	[]	70	27	50	48	34	26	19	18	18	36
22													

Name	Va
a	20
accept_avg	18
b	21
counter	22
dup_indx	34
dup_size	1
dup_words	1x3
i	35
k	1
num	20x
num_col	34
raw	21x
raw_col	34
read_path1	'C:\
rej_col	21
reject_index	1x2
total	36
txt	20x
Write_path1	'C:\
x	34
y	34

Figure 23 - Data after script finish

We can see that each column got the sum of all of its values at last cell, and accept_avg = 18 from the workspace box on the right, Any column that its sum is les than the accept_avg are going to be deleted after calling the script, and the sum row will be deleted from the data. All result will be inside the new .xlsx file by the given path in the code.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Web_id	Tag Name	Web_Address	SCIENCE	BUSINESS	DATA	LEARNING	ENGINEEF	COMPUTE	INFORMA	CLOUD	SQL	CORONAVIRUS	
2	1	AAAS Home | American Associa	https://www.aaas.org/	4	0	0	0	0	0	0	0	0	1	
3	2	Science | AAAS	https://www.sciencemag.org/	25	0	2	2	2	0	0	0	0	18	
4	3	Coursera | Build Skills with Onl	https://www.coursera.org/	13	13	11	27	6	4	1	5	1	1	
5	4	Learn to Code - for Free | Codec	https://www.codecademy.com/	13	0	11	6	6	4	1	0	0	1	
6	5	Stanford Engineering Everywher	https://see.stanford.edu/	1	0	0	0	6	4	1	0	0	1	
7	6	Computer Science Online | Com	https://www.computerscienceonli	3	0	0	1	6	4	1	0	0	1	
8	7	Computer science - Latest resea	https://www.nature.com/subjects/	0	0	0	1	6	4	1	0	0	1	
9	8	International Association of Co	http://www.iacsit.org/	4	1	0	1	1	4	4	0	0	1	
10	9	IT news, careers, business tech	https://www.computerworld.com/	0	3	1	1	0	1	1	2	0	1	
11	10	Learn the Latest Tech Skills; Adv	https://www.udacity.com/	0	0	1	1	0	1	1	0	0	1	
12	11	MathWorks - Makers of MATLAB	https://www.mathworks.com/	3	0	4	1	0	0	1	0	0	1	
13	12	Topcoder | Design & Build High	https://www.topcoder.com/	2	0	3	1	1	0	1	0	0	1	
14	13	Purchase Intent Data for Enterp	https://www.techtarget.com/	0	6	8	1	0	0	1	8	0	1	
15	14	Stack Overflow - Where Develop	https://stackoverflow.com/	0	4	0	1	0	0	1	2	0	1	
16	15	SQLZOO	https://sqlzoo.net/	0	0	3	1	0	0	1	1	16	1	
17	16	SourceForge - Download, Devel	https://sourceforge.net/	0	0	3	1	0	0	1	0	0	1	
18	17	MIT Technology Review	https://www.technologyreview.con	0	0	3	1	0	0	1	0	1	1	
19	18	Khan Academy | Free Online Co	https://www.khanacademy.org/	2	0	0	1	0	0	1	0	0	1	
20	19	Live Science: The Most Interesti	https://www.livescience.com/	0	0	0	0	0	0	0	0	0	1	

Figure 24 - DB after filtering script finished

8.1.2.2.3 Script Code

```
%Clear everything before start%
clear all;
clc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                            %part 1 - Insertion%


%Paths
read_path1 = 'C:\Users\xxmoh\Desktop\GP2\Presentation\Scripts\1 -
Importing\web_dataset.xlsx';
Write_path1 = 'C:\Users\xxmoh\Desktop\GP2\Presentation\Scripts\2 -
Filtering\web_dataset_filterd.xlsx';


[num, txt, raw] = xlsread(read_path1); %Read xlsx file
raw_col = size(raw,2); %Get the number of column in raw
counter = 1;

%Find duplicated words and deleted
for i=1:1:raw_col
    %Search for the dup_word(i) with logic vector
    dup_words = strcmp(raw(1,counter),raw(1,:));

    dup_indx = find(dup_words == 1);     %Find the index for the dup words
    dup_size = size(dup_indx,2);     %Get the size of the indexes

    %Delete the duplicated words
    for x = dup_size:-1:2
        raw(:,dup_indx(1,x)) = [];
        num(:,dup_indx(1,x)) = [];
        txt(:,dup_indx(1,x)) = [];
    end

    raw_col = size(raw,2);      %Update the size after each deletion

    %Check to not get out of the range for .xlsx
    if (counter < raw_col)
        counter = counter + 1;
    else
        continue;
    end
end
num_col = size(num(),2); %Get the number of columns in .xlsx
```

```
%Pointer gets the last row in raw to insert the sum
a = size(raw,1);
b = size(raw,1)+1;

%Inserting sum for all column
for x = 4:1:num_col
    %sum of all element in column x%
    total = sum(num(:,x));

    %Insert the sum at the last cell%
    num(a,x) = total;
    raw(b,x) = num2cell(total);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        %part 2 - Checking%

accept_avg = floor(mean(num(a,4:end))); %Average to be accepted
counter = 1;

%Store the index of rejected data
for y = 4:1:size(raw,2)
    %Check and save index for items to be deleted%
    if (num(a,y) < accept_avg)
        reject_index(counter) = y;
        counter = counter + 1;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %part 3 - Deletion & Writing%

rej_col = size(reject_index,2); %Get the # of column for rejected items

%Filter the data depends on avg
for k = rej_col:-1:1
    raw(:,reject_index(1,k)) = [];
end
raw(b,:) = [];
xlswrite(Write_path1, raw);
```

8.1.2.3 K-means

k-means clustering is a partitioning method. The function kmeans partitions data into *k* mutually exclusive clusters and returns the index of the cluster to which it assigns each observation. kmeans treats each observation in the data as an object that has a location in space. The function finds a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. Each cluster in a *k*-means partition consists of member objects and a centroid . In each cluster kmeans minimizes the sum of the distances between the centroid and all member objects of the cluster and computes centroid clusters differently for the supported distance metrics. And use *k*-means++ algorithm for centroid initialization and squared Euclidean distance by default. ***k*-means++** is an algorithm for choosing the initial values (or "seeds") for the *k*-means clustering algorithm.

8.1.2.3.1 *K-means* Flaw chart

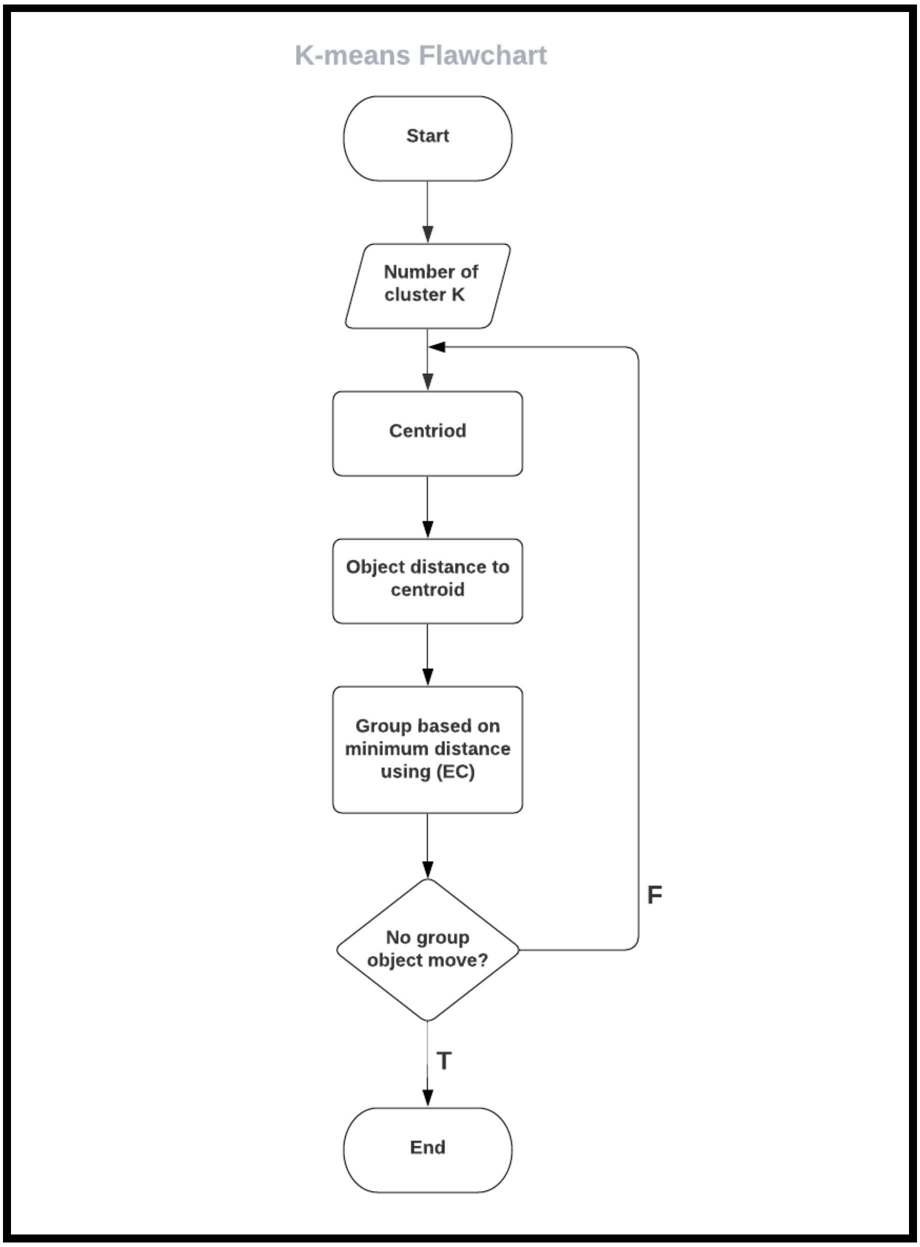
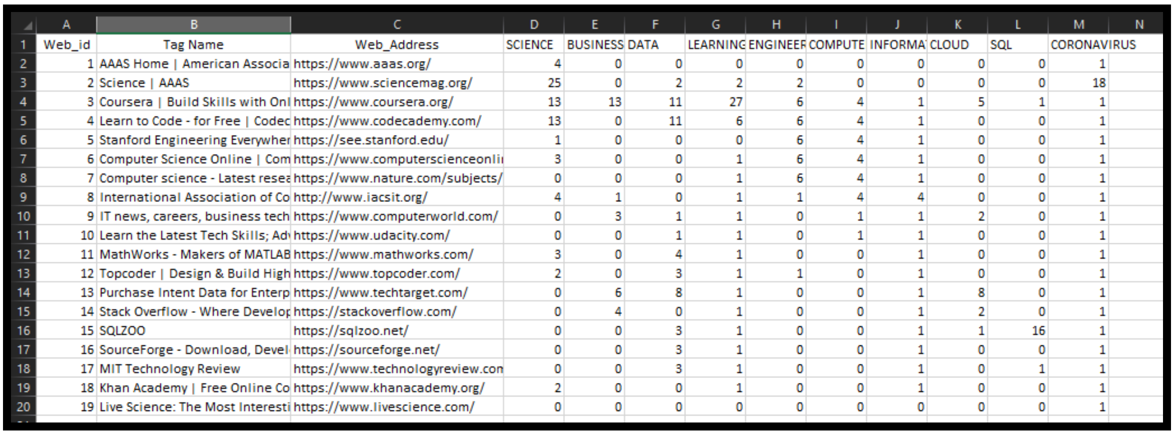


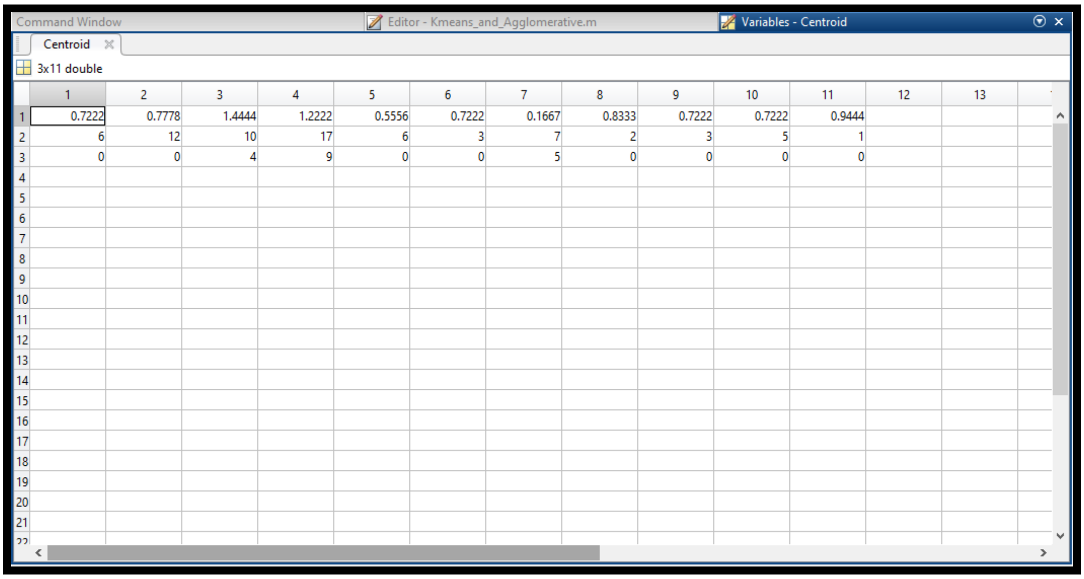
Figure 25 - k means algorithm

8.1.2.3.2 *K-means* result



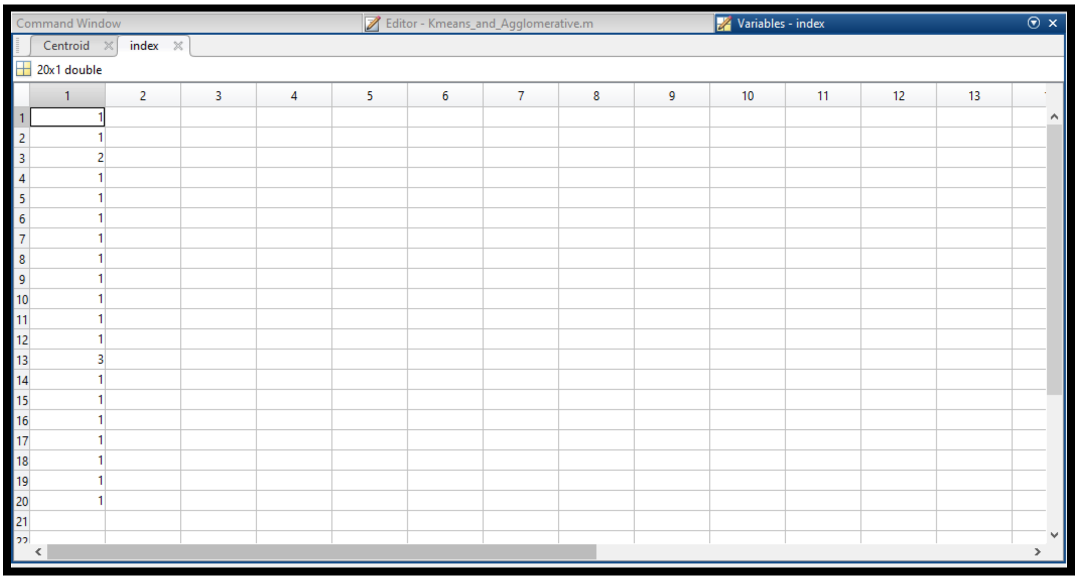
Web_id	Tag Name	Web_Address	SCIENCE	BUSINESS	DATA	LEARNING	ENGINEER	COMPUTE	INFORMA	CLOUD	SQL	CORONAVIRUS
1	AAAS Home | American Associa	https://www.aaas.org/	4	0	0	0	0	0	0	0	0	1
2	Science | AAAS	https://www.sciencemag.org/	25	0	2	2	2	0	0	0	0	18
3	Coursera | Build Skills with Onl	https://www.coursera.org/	13	13	11	27	6	4	1	5	1	1
4	Learn to Code - for Free | Codec	https://www.codecademy.com/	13	0	11	6	6	4	1	0	0	1
5	Stanford Engineering Everywher	https://see.stanford.edu/	1	0	0	0	6	4	1	0	0	1
6	Computer Science Online | Com	https://www.computerscienceonli	3	0	0	1	6	4	1	0	0	1
7	Computer science - Latest resea	https://www.nature.com/subjects/	0	0	0	1	6	4	1	0	0	1
8	International Association of Co	http://www.iacsit.org/	4	1	0	1	1	4	4	0	0	1
9	IT news, careers, business tech	https://www.computerworld.com/	0	3	1	1	0	1	1	2	0	1
10	Learn the Latest Tech Skills; Adv	https://www.udacity.com/	0	0	1	1	0	1	1	0	0	1
11	MathWorks - Makers of MATLAB	https://www.mathworks.com/	3	0	4	1	0	0	1	0	0	1
12	Topcoder | Design & Build High	https://www.topcoder.com/	2	0	3	1	1	0	1	0	0	1
13	Purchase Intent Data for Enterp	https://www.techtarget.com/	0	6	8	1	0	0	1	8	0	1
14	Stack Overflow - Where Develop	https://stackoverflow.com/	0	4	0	1	0	0	1	2	0	1
15	SQLZOO	https://sqlzoo.net/	0	0	3	1	0	0	1	1	16	1
16	SourceForge - Download, Devel	https://sourceforge.net/	0	0	3	1	0	0	1	0	0	1
17	MIT Technology Review	https://www.technologyreview.com	0	0	3	1	0	0	1	0	1	1
18	Khan Academy | Free Online Co	https://www.khanacademy.org/	2	0	0	1	0	0	1	0	0	1
19	Live Science: The Most Interesti	https://www.livescience.com/	0	0	0	0	0	0	0	0	0	1

Figure 26 - The input data to be clustered



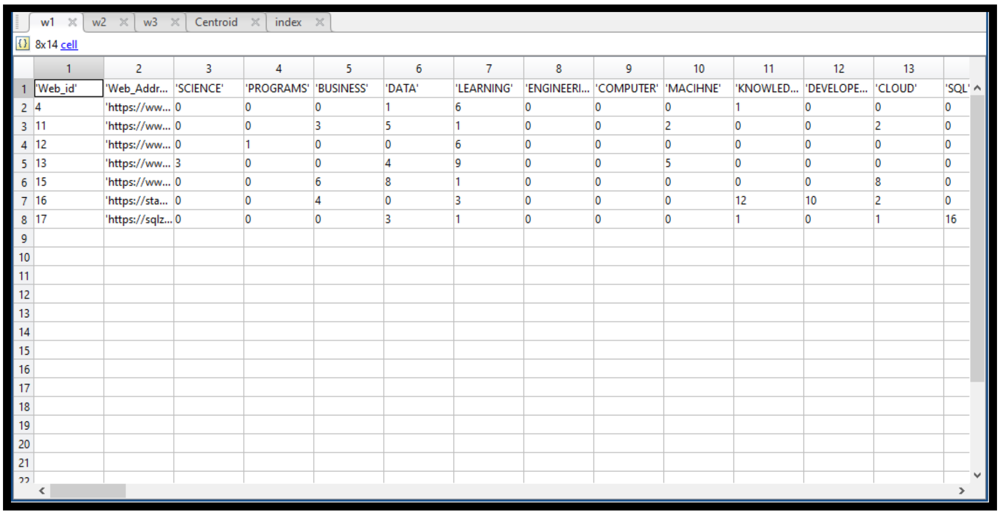
	1	2	3	4	5	6	7	8	9	10	11
1	0.7222	0.7778	1.4444	1.2222	0.5556	0.7222	0.1667	0.8333	0.7222	0.7222	0.9444
2	6	12	10	17	6	3	7	2	3	5	1
3	0	0	4	9	0	0	5	0	0	0	0

Figure 27 – Centroid



	1
1	1
2	1
3	2
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	3
14	1
15	1
16	1
17	1
18	1
19	1
20	1

Figure 28 - Indexes

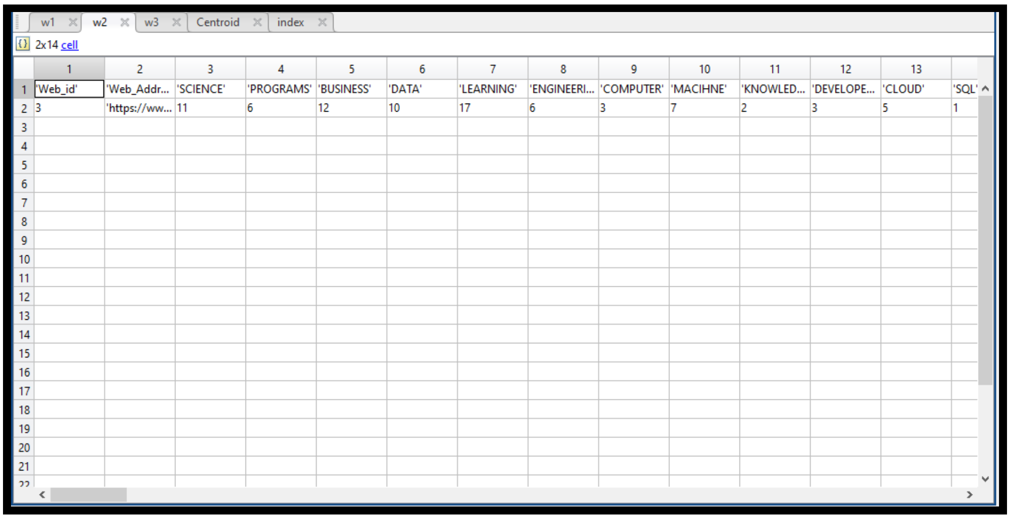


w1 w2 w3 Centroid index

8x14 cell

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	'Web_id'	'Web_Addr...	'SCIENCE'	'PROGRAMS'	'BUSINESS'	'DATA'	'LEARNING'	'ENGINEERI...	'COMPUTER'	'MACIHNE'	'KNOWLED...	'DEVELOPE...	'CLOUD'	'SQL'
2	4	'https://ww...	0	0	0	1	6	0	0	0	1	0	0	0
3	11	'https://ww...	0	0	3	5	1	0	0	2	0	0	2	0
4	12	'https://ww...	0	1	0	0	6	0	0	0	0	0	0	0
5	13	'https://ww...	3	0	0	4	9	0	0	5	0	0	0	0
6	15	'https://ww...	0	0	6	8	1	0	0	0	0	0	8	0
7	16	'https://sta...	0	0	4	0	3	0	0	0	12	10	2	0
8	17	'https://sqlz...	0	0	0	3	1	0	0	0	1	0	1	16

Figure 29 - Cluster 1

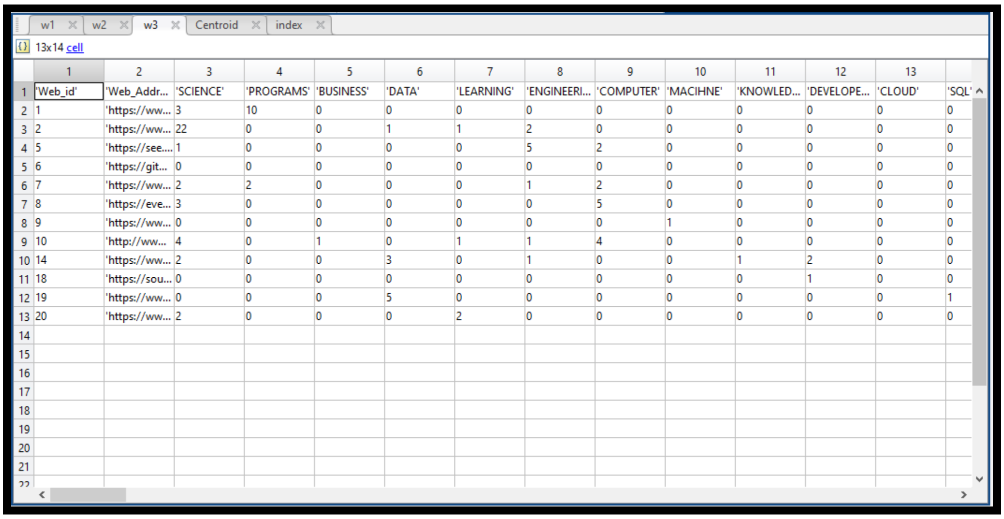


w1 w2 w3 Centroid index

2x14 cell

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	'Web_id'	'Web_Addr...	'SCIENCE'	'PROGRAMS'	'BUSINESS'	'DATA'	'LEARNING'	'ENGINEERI...	'COMPUTER'	'MACIHNE'	'KNOWLED...	'DEVELOPE...	'CLOUD'	'SQL'
2	3	'https://ww...	11	6	12	10	17	6	3	7	2	3	5	1

Figure 30 - Cluster 2



w1 w2 w3 Centroid index

13x14 cell

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	'Web_id'	'Web_Addr...	'SCIENCE'	'PROGRAMS'	'BUSINESS'	'DATA'	'LEARNING'	'ENGINEERI...	'COMPUTER'	'MACIHNE'	'KNOWLED...	'DEVELOPE...	'CLOUD'	'SQL'
2	1	'https://ww...	3	10	0	0	0	0	0	0	0	0	0	0
3	2	'https://ww...	22	0	0	1	1	2	0	0	0	0	0	0
4	5	'https://see....	1	0	0	0	0	5	2	0	0	0	0	0
5	6	'https://git...	0	0	0	0	0	0	0	0	0	0	0	0
6	7	'https://ww...	2	2	0	0	0	1	2	0	0	0	0	0
7	8	'https://eve...	3	0	0	0	0	0	5	0	0	0	0	0
8	9	'https://ww...	0	0	0	0	0	0	0	1	0	0	0	0
9	10	'http://ww...	4	0	1	0	1	1	4	0	0	0	0	0
10	14	'https://ww...	2	0	0	3	0	1	0	0	1	2	0	0
11	18	'https://sou...	0	0	0	0	0	0	0	0	0	1	0	0
12	19	'https://ww...	0	0	0	5	0	0	0	0	0	0	0	1
13	20	'https://ww...	2	0	0	0	2	0	0	0	0	0	0	0

Figure 31 - Cluster 3

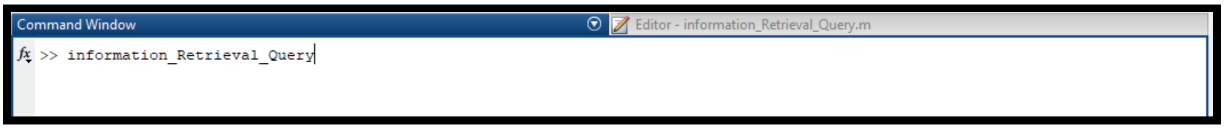
All the websites used here most like each other(‘science’, ‘university’, ‘technology’) sites.

8.1.2.3.3 *K-means* Code

```
                          %Part 1 - Paths and Reading%
 read_path = 'C:\Users\xxmoh\Desktop\GP2\Presentation\Scripts\3 - 
Clustering using K-means & Agglomerative\web_dataset_filterd.xlsx';

 Cluster1_path = 'C:\Users\xxmoh\Desktop\GP2\Presentation\Scripts\3 - 
Clustering using K-means & Agglomerative\C1.xlsx';

 Cluster2_path = 'C:\Users\xxmoh\Desktop\GP2\Presentation\Scripts\3 - 
Clustering using K-means & Agglomerative\C2.xlsx';

 Cluster3_path = 'C:\Users\xxmoh\Desktop\GP2\Presentation\Scripts\3 - 
Clustering using K-means & Agglomerative\C3.xlsx';

 [num, txt, raw] = xlsread(read_path); %Read Xlsx db.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                              %Part 2 - K-means%

%Get websites keyword vector
 web_vec = num(:,4:end);
 web = raw(2:end,:);

%Kmeans and clustering
rng(1);
[index, Centroid] = kmeans(web_vec,3);

%Find indexes for each site\cluster
 c1 = find(index == 1);
 c2 = find(index == 2);
 c3 = find(index == 3);

%Insert data vector in each cluster
 w1 = [txt(1,:); web(c1,:)]
 w2 = [txt(1,:); web(c2,:)]
 w3 = [txt(1,:); web(c3,:)]

 %Print K-means clusters
 w1
 w2
 w3

 xlswrite(Cluster1_path, w1);
 xlswrite(Cluster2_path, w2);
 xlswrite(Cluster3_path, w3);
```

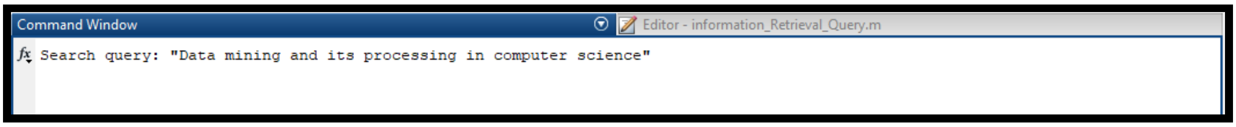
8.1.2.4 Information Retrieval using k-means result

- In this part the script starts by searching in the clusters by entering single, multiple word, or a search query then it will delete the stop words from the search word in order to make the searching better and faster, then it will start to find the words that are meet the keywords inside xlsx. And get the indexes for each one of them then get the highest centroid value from all the k-means clusters, (ex. If we have 3 clusters and 3 search words “Data, image, SQL” after clearing the stop words from the query the script will compare the appearance for each of 3 words in each cluster and get the highest centroid value for each word “each word will have 3 value because we have 3 clusters”, then it will compare all the 3 highest centroid values from each word and display the cluster that has the highest centroid value among them).



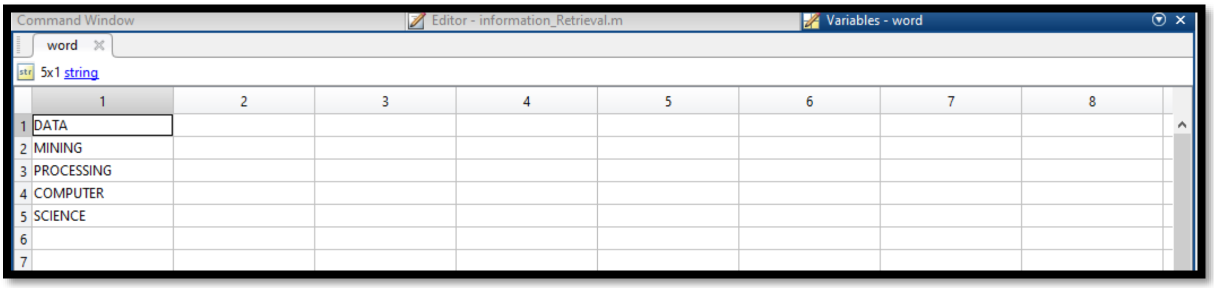
```
>> information_Retrieval_Query
```

Figure 32 - Run Search script



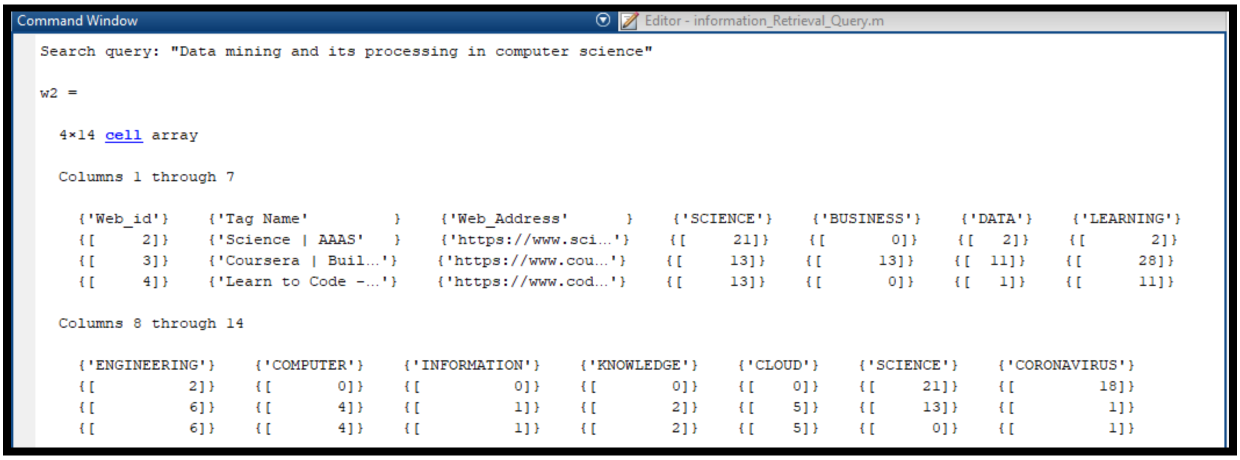
```
Search query: "Data mining and its processing in computer science"
```

Figure 33 - Enter the search keyword or Query



	1	2	3	4	5	6	7	8
1	DATA							
2	MINING							
3	PROCESSING							
4	COMPUTER							
5	SCIENCE							
6								
7								

Figure 34 - After clearing stop words



```
Search query: "Data mining and its processing in computer science"

w2 =

  4×14 cell array

  Columns 1 through 7

    {'Web_id'}    {'Tag Name'           }    {'Web_Address'       }    {'SCIENCE'}    {'BUSINESS'}    {'DATA'}    {'LEARNING'}
    {[     2]}    {'Science | AAAS'     }    {'https://www.sci…'}    {[      21]}    {[        0]}    {[   2]}    {[        2]}
    {[     3]}    {'Coursera | Buil…'}    {'https://www.cou…'}    {[      13]}    {[       13]}    {[  11]}    {[       28]}
    {[     4]}    {'Learn to Code -…'}    {'https://www.cod…'}    {[      13]}    {[        0]}    {[   1]}    {[       11]}

  Columns 8 through 14

    {'ENGINEERING'}    {'COMPUTER'}    {'INFORMATION'}    {'KNOWLEDGE'}    {'CLOUD'}    {'SCIENCE'}    {'CORONAVIRUS'}
    {[           2]}    {[        0]}    {[            0]}    {[          0]}    {[    0]}    {[      21]}    {[            18]}
    {[           6]}    {[        4]}    {[            1]}    {[          2]}    {[    5]}    {[      13]}    {[             1]}
    {[           6]}    {[        4]}    {[            1]}    {[          2]}    {[    5]}    {[       0]}    {[             1]}
```

Figure 35 - Information retrieval result

8.1.2.4.1 Information Retrieval code

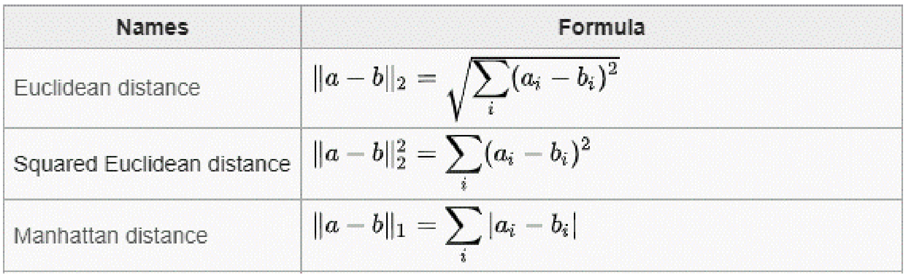
```
                                    %Continue with kmeans code%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        %Part 3 - Searching%

%Search for a word and get the index
word = input("Search query: ");
word = upper(word);

%Stop words
stop_words =
["A";"ABOUT";"ABOVE";"AFTER";"AGAIN";"AGAINST";"ALL";"AM";"AN";"AND";"ANY";
"ARE";"AREN'T";"AS";"AT";"BE";"BECAUSE";"BEEN";"BEFORE";"BEING";"BELOW";"BE
TWEEN";"BOTH";"BUT";"BY";"CAN'T";"CANNOT";"COULD";"COULDN'T";"DID";"DIDN'T"
;"DO";"DOES";"DOESN'T";"DOING";"DON'T";"DOWN";"DURING";"EACH";"FEW";"FOR";"
FROM";"FURTHER";"HAD";"HADN'T";"HAS";"HASN'T";"HAVE";"HAVEN'T";"HAVING";"HE
";"HE'D";"HE'LL";"HE'S";"HER";"HERE";"HERE'S";"HERS";"HERSELF";"HIM";"HIMSE
LF";"HIS";"HOW";"HOW'S";"I";"I'D";"I'LL";"I'M";"I'VE";"IF";"IN";"INTO";"IS"
;"ISN'T";"IT";"IT'S";"ITS";"ITSELF";"LET'S";"ME";"MORE";"MOST";"MUSTN'T";"M
Y";"MYSELF";"NO";"NOR";"NOT";"OF";"OFF";"ON";"ONCE";"ONLY";"OR";"OTHER";"OU
GHT";"OUR";"OURS";"OURSELVES";"OUT";"OVER";"OWN";"SAME";"SHAN'T";"SHE";"SHE
'D";"SHE'LL";"SHE'S";"SHOULD";"SHOULDN'T";"SO";"SOME";"SUCH";"THAN";"THAT";
"THAT'S";"THE";"THEIR";"THEIRS";"THEM";"THEMSELVES";"THEN";"THERE";"THERE'S
";"THESE";"THEY";"THEY'D";"THEY'LL";"THEY'RE";"THEY'VE";"THIS";"THOSE";"THR
OUGH";"TO";"TOO";"UNDER";"UNTIL";"UP";"VERY";"WAS";"WASN'T";"WE";"WE'D";"WE
'LL";"WE'RE";"WE'VE";"WERE";"WEREN'T";"WHAT";"WHAT'S";"WHEN";"WHEN'S";"WHER
E";"WHERE'S";"WHICH";"WHILE";"WHO";"WHO'S";"WHOM";"WHY";"WHY'S";"WITH";"WON
'T";"WOULD";"WOULDN'T";"YOU";"YOU'D";"YOU'LL";"YOU'RE";"YOU'VE";"YOUR";"YOU
RS";"YOURSELF";"YOURSELVES";",";"";"~";"!";"@";"#";"$";"%";"^";"&";"*";"(";
")";"{";"}";"[";"]";"_";"";"+";"=";"/";"*";"";"+";"|";"\";"]";"'";'"';"<";"
>";"?";".";";";];
stop_words =  stop_words'; %Transpose the words

%Split the search text
word = split(word); %Split each word in search query
[sp_col , ~] = find(word == stop_words);

word(sp_col) = [];%Delete the stop words
NSW = size(word,1);

%Compare search words with k-words
  for x=1:1:NSW
     s_isAvaliable = find(txt(1,4:end) == word(x,1));
     if(isempty(s_isAvaliable))
          Unavilable_word_index(x) = word(x,1);
     else
          SWV(1,s_isAvaliable) = 1;
     end
  end
  SWV_Cent = length(find(SWV == 1));

  if (SWV_Cent == 0)
      error = "Try Again!"
      information_retriver
  end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        %Part 4 - Information Retrieval%

  for i=1:1:SWV_Cent
      sid = find(SWV == 1);
      [Centroid_row, Centroid_col] = find( Centroid == max(Centroid(:,sid(1,i))));
      cent_max(i) = max(Centroid(:,sid(1,i)));
      [m_row, m_col ]= find(cent_max == max(cent_max));
      t1 = sid(:,m_col);
      t2 = Centroid(:,t1);
      [s_row, ~] = find(Centroid == max(t2(:)));
  end

  %Show the cluster
  if (s_row(1,1) == 1)
      w1
  elseif (s_row(1,1) == 2)
      w2
  elseif (s_row(1,1) == 3)
      w3
  end
```

8.1.3 Agglomerative Clustering

In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, this is achieved by use of an appropriate metric (a measure of distance between pairs of observations), and a linkage criterion which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

The choice of an appropriate metric will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. For example, in a 2-dimensional space, the distance between the point (1,0) and the origin (0,0) is always 1 according to the usual norms, but the distance between the point (1,1) and the origin (0,0) can be 2 under Manhattan distance, $\sqrt{2}$ under Euclidean distance, or 1 under maximum distance. **Figure.35** shows some commonly metrics used for hierarchical clustering:



Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|a-b\|_1 = \sum_i \lvert a_i - b_i \rvert$

Figure 36 - some commonly metrics used for hierarchical clustering

8.1.3.1 Agglomerative Flaw chart

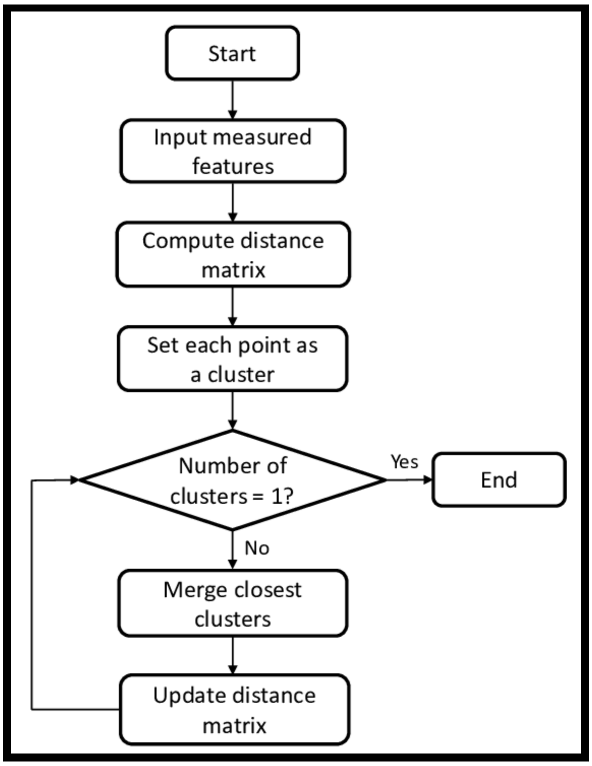
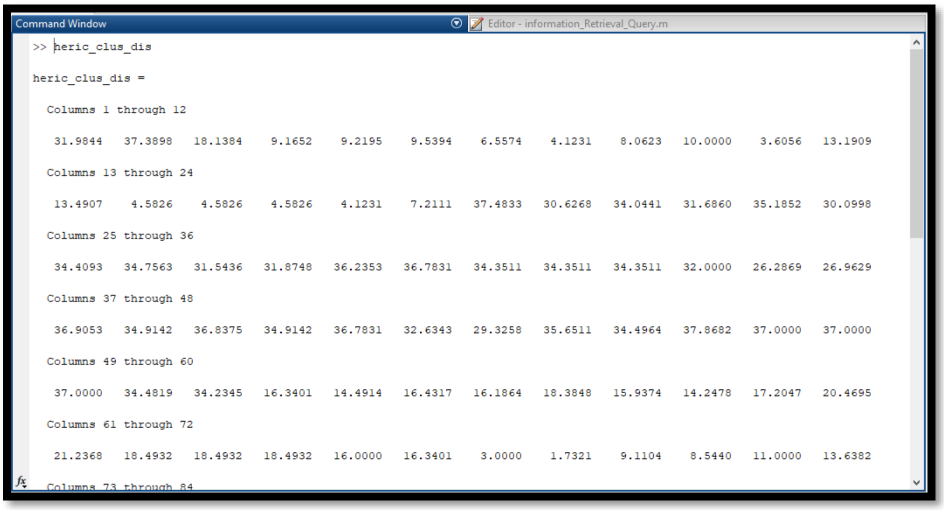


Figure 37 - Agglomerative Flaw chart

8.1.3.2 Agglomerative result

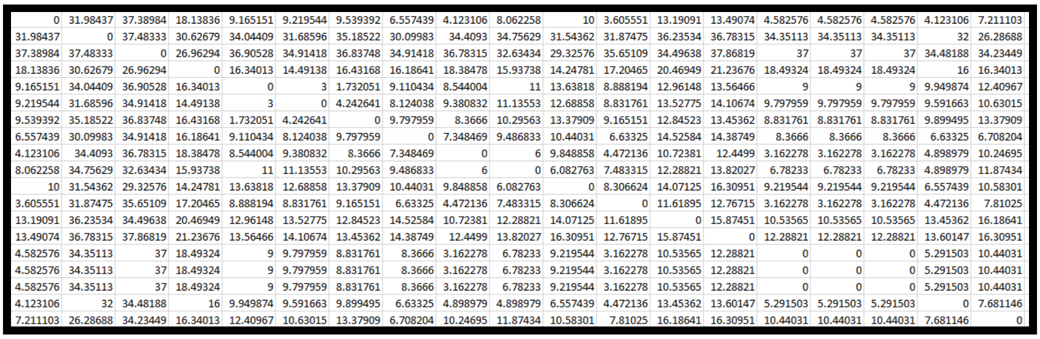
- At this part all filtered data will go through 3 functions inside the MATLAB:

1) **Pdist:** Get the distance between each data point.



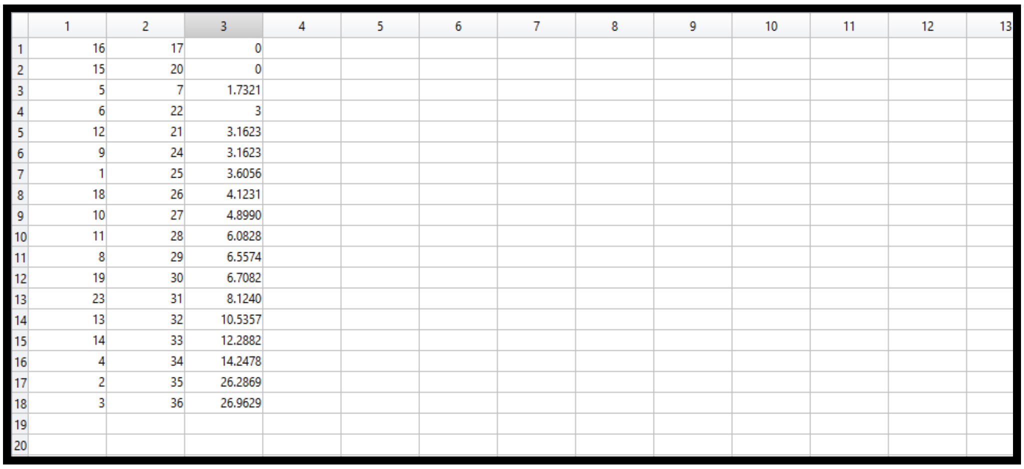
```
>> heric_clus_dis

heric_clus_dis =

  Columns 1 through 12

   31.9844   37.3898   18.1384    9.1652    9.2195    9.5394    6.5574    4.1231    8.0623   10.0000    3.6056   13.1909

  Columns 13 through 24

   13.4907    4.5826    4.5826    4.5826    4.1231    7.2111   37.4833   30.6268   34.0441   31.6860   35.1852   30.0998

  Columns 25 through 36

   34.4093   34.7563   31.5436   31.8748   36.2353   36.7831   34.3511   34.3511   34.3511   32.0000   26.2869   26.9629

  Columns 37 through 48

   36.9053   34.9142   36.8375   34.9142   36.7831   32.6343   29.3258   35.6511   34.4964   37.8682   37.0000   37.0000

  Columns 49 through 60

   37.0000   34.4819   34.2345   16.3401   14.4914   16.4317   16.1864   18.3848   15.9374   14.2478   17.2047   20.4695

  Columns 61 through 72

   21.2368   18.4932   18.4932   18.4932   16.0000   16.3401    3.0000    1.7321    9.1104    8.5440   11.0000   13.6382

  Columns 73 through 84
```

- These are the distances between every data point, we can transport it into proper form like a table using `squareform function.`



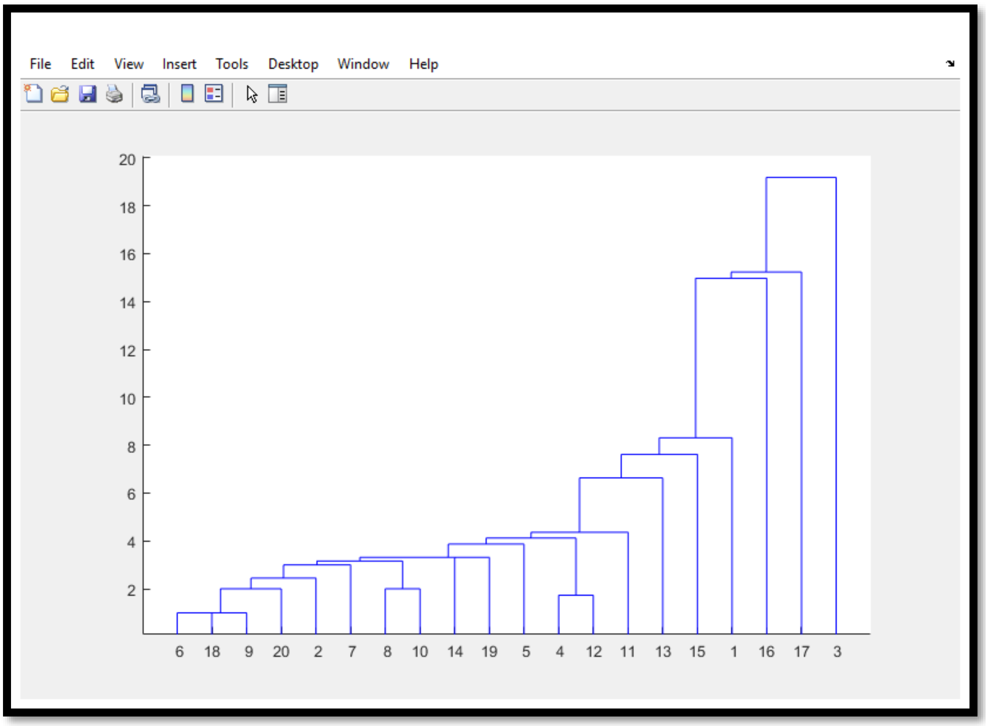
0	31.98437	37.38984	18.13836	9.165151	9.219544	9.539392	6.557439	4.123106	8.062258	10	3.605551	13.19091	13.49074	4.582576	4.582576	4.582576	4.123106	7.211103
31.98437	0	37.48333	30.62679	34.04409	31.68596	35.18522	30.09983	34.4093	34.75629	31.54362	31.87475	36.23534	36.78315	34.35113	34.35113	34.35113	32	26.28688
37.38984	37.48333	0	26.96294	36.90528	34.91418	36.83748	34.91418	36.78315	32.63434	29.32576	35.65109	34.49638	37.86819	37	37	37	34.48188	34.23449
18.13836	30.62679	26.96294	0	16.34013	14.49138	16.43168	16.18641	18.38478	15.93738	14.24781	17.20465	20.46949	21.23676	18.49324	18.49324	18.49324	16	16.34013
9.165151	34.04409	36.90528	16.34013	0	3	1.732051	9.110434	8.544004	11	13.63818	8.888194	12.96148	13.56466	9	9	9	9.949874	12.40967
9.219544	31.68596	34.91418	14.49138	3	0	4.242641	8.124038	9.380832	11.13553	12.68858	8.831761	13.52775	14.10674	9.797959	9.797959	9.797959	9.591663	10.63015
9.539392	35.18522	36.83748	16.43168	1.732051	4.242641	0	9.797959	8.3666	10.29563	13.37909	9.165151	12.84523	13.45362	8.831761	8.831761	8.831761	9.899495	13.37909
6.557439	30.09983	34.91418	16.18641	9.110434	8.124038	9.797959	0	7.348469	9.486833	10.44031	6.63325	14.52584	14.38749	8.3666	8.3666	8.3666	6.63325	6.708204
4.123106	34.4093	36.78315	18.38478	8.544004	9.380832	8.3666	7.348469	0	6	9.848858	4.472136	10.72381	12.4499	3.162278	3.162278	3.162278	4.898979	10.24695
8.062258	34.75629	32.63434	15.93738	11	11.13553	10.29563	9.486833	6	0	6.082763	7.483315	12.28821	13.82027	6.78233	6.78233	6.78233	4.898979	11.87434
10	31.54362	29.32576	14.24781	13.63818	12.68858	13.37909	10.44031	9.848858	6.082763	0	8.306624	14.07125	16.30951	9.219544	9.219544	9.219544	6.557439	10.58301
3.605551	31.87475	35.65109	17.20465	8.888194	8.831761	9.165151	6.63325	4.472136	7.483315	8.306624	0	11.61895	12.76715	3.162278	3.162278	3.162278	4.472136	7.81025
13.19091	36.23534	34.49638	20.46949	12.96148	13.52775	12.84523	14.52584	10.72381	12.28821	14.07125	11.61895	0	15.87451	10.53565	10.53565	10.53565	13.45362	16.18641
13.49074	36.78315	37.86819	21.23676	13.56466	14.10674	13.45362	14.38749	12.4499	13.82027	16.30951	12.76715	15.87451	0	12.28821	12.28821	12.28821	13.60147	16.30951
4.582576	34.35113	37	18.49324	9	9.797959	8.831761	8.3666	3.162278	6.78233	9.219544	3.162278	10.53565	12.28821	0	0	0	5.291503	10.44031
4.582576	34.35113	37	18.49324	9	9.797959	8.831761	8.3666	3.162278	6.78233	9.219544	3.162278	10.53565	12.28821	0	0	0	5.291503	10.44031
4.582576	34.35113	37	18.49324	9	9.797959	8.831761	8.3666	3.162278	6.78233	9.219544	3.162278	10.53565	12.28821	0	0	0	5.291503	10.44031
4.123106	32	34.48188	16	9.949874	9.591663	9.899495	6.63325	4.898979	4.898979	6.557439	4.472136	13.45362	13.60147	5.291503	5.291503	5.291503	0	7.681146
7.211103	26.28688	34.23449	16.34013	12.40967	10.63015	13.37909	6.708204	10.24695	11.87434	10.58301	7.81025	16.18641	16.30951	10.44031	10.44031	10.44031	7.681146	0

2) **Linkage:** Take distances from "Pdist" and links pairs of objects that are close together into binary clusters. Also, by default it's a Single linkage



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	16	17	0										
2	15	20	0										
3	5	7	1.7321										
4	6	22	3										
5	12	21	3.1623										
6	9	24	3.1623										
7	1	25	3.6056										
8	18	26	4.1231										
9	10	27	4.8990										
10	11	28	6.0828										
11	8	29	6.5574										
12	19	30	6.7082										
13	23	31	8.1240										
14	13	32	10.5357										
15	14	33	12.2882										
16	4	34	14.2478										
17	2	35	26.2869										
18	3	36	26.9629										
19													
20													

3) **Dendrogram:** Show the dendrogram for the Binary clusters from Linkage process.



8.1.3.3 Agglomerative Code

```
read_path = 'C:\Users\xxmoh\Desktop\GP2\Presentation\Scripts\3 - Clustering
using K-means & Agglomerative\web_dataset_filterd.xlsx';
[num, txt, raw] = xlsread(read_path);


%Convert cell to matrix
 heric_clus = cell2mat(raw(2:end, 4:end));

%Get the distance between each object
 heric_clus_dis = pdist(heric_clus,'euclidean');

%Transpose the distances into table to make it easier to read *Optional*
 heric_clus_dis_tb = squareform(heric_clus_dis);

%Take distance from pdist and links pairs of objects that are close
together into binary clusters
 heric_clus_dis_tree = linkage(heric_clus_dis, 'centroid');

%Show the dendogram
 dendrogram(heric_clus_dis_tree);
```

Chapter 9: Webpage classification

Classification of Web pages is one of the challenging and important tasks as there is an increase in web pages in day to day life provided by internet. There are many ways of classifying web pages based on different approach and features. Web pages are allocated to pre-determined categories which is done mainly according to their content in Web page classification. The important technique for web mining is web page classification because classifying the web pages of interesting class is the initial step of data mining. In this project we used two different classification algorithms to classify webpages Decision tree and Artificial neural network.

9.1 Classification Algorithms vs Clustering Algorithms

In clustering algorithms, the idea is not to predict the target class as in classification, it's more ever trying to group the similar kind of things by considering the most satisfied condition**,** all the items in the same group should be similar and no two different group items should not be similar.

9.2 Algorithms

In this project we used two classification algorithms to get our result and they are Decision tree and Artificial neural network. The idea of Classification Algorithms is to predict the target class by analyzing the training dataset. We use the training dataset to get better boundary conditions which could be used to determine each target class. Once the boundary conditions are determined, the next task is to predict the target class[17]. The whole process is known as classification.

9.2.1 Basic Terms in Classification Algorithms

- **Classifier:** An algorithm that maps the input data to a specific category.
- **Classification model:** A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- **Feature:** A feature is an individual measurable property of a phenomenon being observed.
- **Binary Classification:** Classification task with two possible outcomes. Ex: Gender classification (Male / Female).

9.2.2 Types of Classification Algorithms

Classification Algorithms can be classified as

- Linear Classifiers (Naïve Bayes)
- Support vector machines
- Quadratic classifiers
- Kernel estimation(k-nearest neighbor)
- Decision trees
- Neural networks
- Learning vector quantization

9.2.3 Decision tree classification

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

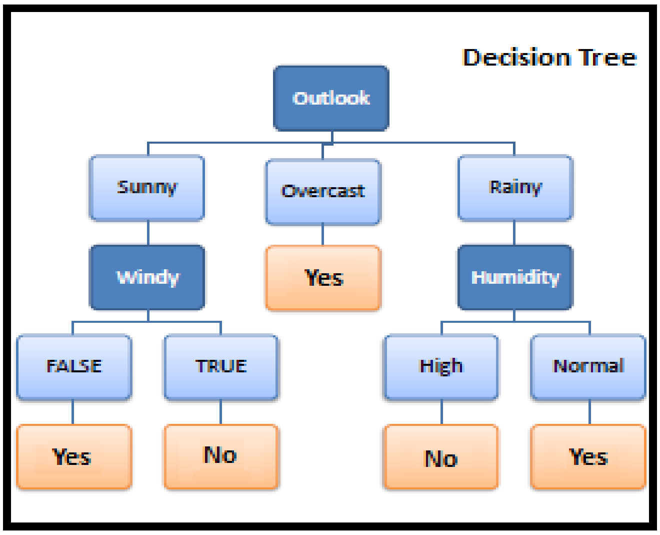


Figure 38 - Decision tree

9.2.3.1 Decision tree result

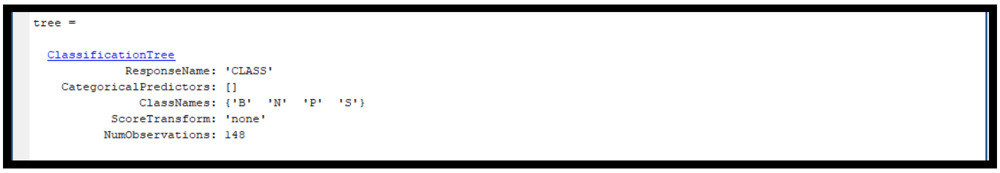
In this project we developed 'DS' script for constructing, training. And predicting our 2 datasets to get result. First dataset has 4 classes (**B**usiness, **N**ews, **S**cience, **S**port), And 200 records. Second dataset has 8 classes (**B**usiness, **N**ews, **S**cience, **S**port, education, history, medical, food), And 400 records. But decision tree is not for training big datasets, so we will train 75% of both datasets and test 25%, since decision tree can't handle big data and get the result for each one before and after applying feature selection(filtering) on both.

Table 8 - Datasets details

Dataset	**Sets**		**Before feature selection**	**After feature selection**
	Training	**Test**	**Features**	**Features**
1	148 record	52 record	152	47
2	296 record	104 record	211	52

9.2.3.1.1 Decision tree result before feature selection

1) Decision tree for first dataset



```
tree =

  ClassificationTree
           ResponseName: 'CLASS'
  CategoricalPredictors: []
             ClassNames: {'B'  'N'  'P'  'S'}
         ScoreTransform: 'none'
        NumObservations: 148
```

Figure 39 - Construction of DS for first dataset

We can see our DS for the first dataset, it has 4 class {'B', 'N', 'P', 'S'} → (**B**usiness, **N**ews, **S**port, **S**cience). And 148 records.

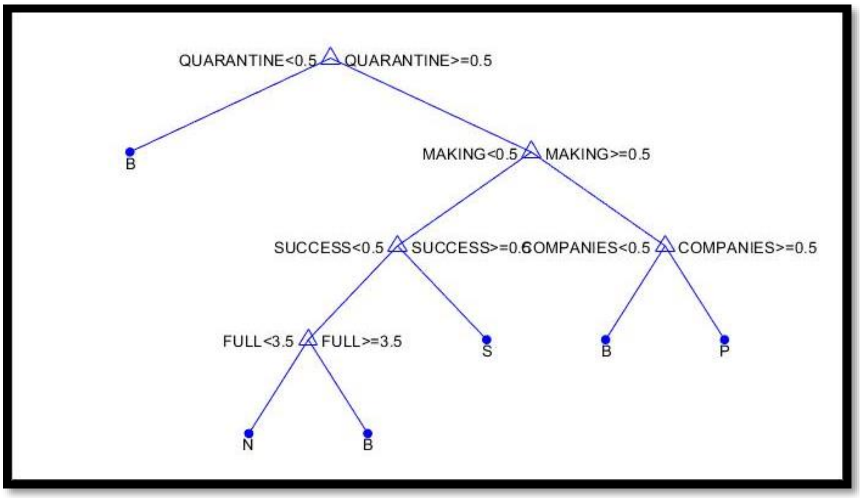


Figure 40 - Decision tree result for first dataset

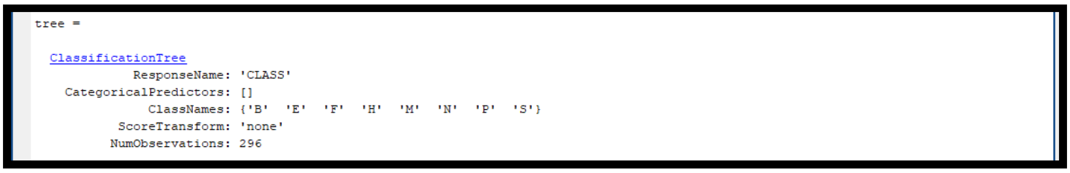
We will now use this tree to predict the 25% of dataset websites for each class and see result in table has two columns, 'Class' for correct class and 'P' for predicted class.

Table 9 – Prediction result for dataset1 before feature selection

Class	P	Class	P	Class	P	Class	P
'B'	**'B'**	'S'	**'S'**	'N'	**'N'**	'P'	**'P'**
'B'	**'B'**	'S'	**'S'**	'N'	**'N'**	'P'	**'P'**
'B'	**'B'**	'S'	**'S'**	'N'	**'B'**	'P'	**'P'**
'B'	**'B'**	'S'	**'S'**	'N'	**'N'**	'P'	**'P'**
'B'	**'B'**	'S'	**'S'**	'N'	**'N'**	'P'	**'P'**
'B'	**'B'**	'S'	**'S'**	'N'	**'B'**	'P'	**'P'**
'B'	**'B'**	'S'	**'P'**	'N'	**'B'**	'P'	**'P'**
'B'	**'B'**	'S'	**'P'**	'N'	**'B'**	'P'	**'P'**
'B'	**'B'**	'S'	**'P'**	'N'	**'B'**	'P'	**'P'**
'B'	**'B'**	'S'	**'P'**	'N'	**'N'**	'P'	**'P'**
'B'	**'B'**	'S'	**'P'**	'N'	**'N'**	'P'	**'P'**
'B'	**'B'**	'S'	**'P'**	'N'	**'N'**	'P'	**'P'**
'B'	**'B'**	'S'	**'P'**	'N'	**'N'**	'P'	**'P'**

From the results above we can see that we have 12 error in predicting and 40 correct one. So, the tree for dataset1 before feature selection has a 23% **ERR** and 77% **ACC.**

2) Decision tree for second dataset



```
tree =

  ClassificationTree
           ResponseName: 'CLASS'
  CategoricalPredictors: []
             ClassNames: {'B'  'E'  'F'  'H'  'M'  'N'  'P'  'S'}
         ScoreTransform: 'none'
        NumObservations: 296
```

Figure 41 - Construction of DT for second dataset

We can see our decision tree for the second dataset, it has 8 class {'B', 'E', 'F', 'H', 'M','N','P','S'} → (**B**usiness, **E**ducation, **F**ood, **H**istory, **M**edical, **N**ews, S**p**ort, **S**cience). And 296 records.

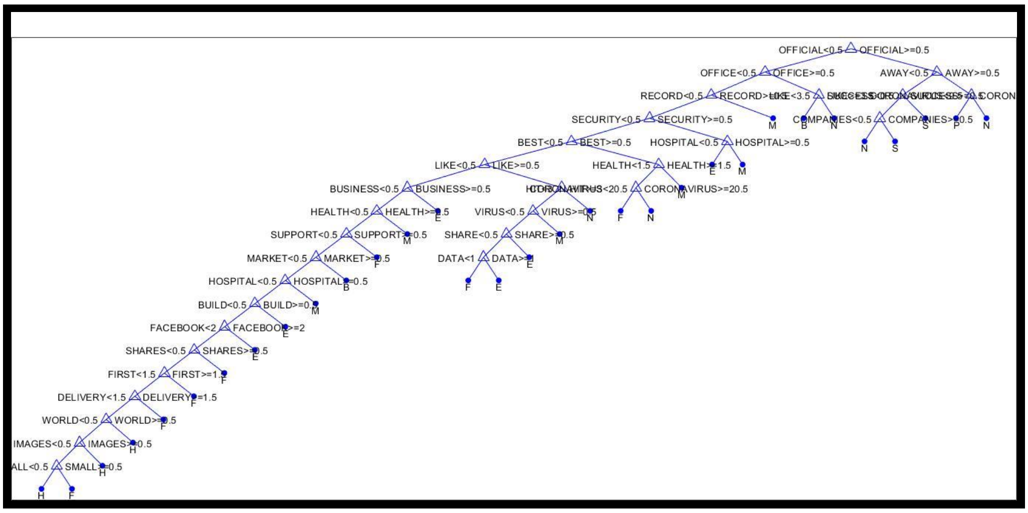


Figure 42 - Decision tree for second dataset

As we can see the figure above shows the decision tree for the second dataset, and we can see how big it is and that's why decision trees are not compatibleand not favorable for big datasets.

Table 10 - Prediction result for dataset2 before feature selection

C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**	'E'	'E'	'H'	'H'	'M'	'H'	'F'	'F'
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**	'E'	'E'	'H'	'H'	'M'	'M'	'F'	'F'
'B'	**'B'**	'N'	**'P'**	'S'	**'S'**	'P'	**'P'**	'E'	'H'	'H'	'F'	'M'	'M'	'F'	'F'
'B'	**'B'**	'N'	**'P'**	'S'	**'S'**	'P'	**'P'**	'E'	'H'	'H'	'F'	'M'	'F'	'F'	'B'
'B'	**'B'**	'N'	**'P'**	'S'	**'P'**	'P'	**'P'**	'E'	'H'	'H'	'H'	'M'	'F'	'F'	'B'
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**	'E'	'H'	'H'	'H'	'M'	'M'	'F'	'B'
'B'	**'B'**	'N'	**'N'**	'S'	**'N'**	'P'	**'P'**	'E'	'H'	'H'	'F'	'M'	'M'	'F'	'B'
'B'	**'B'**	'N'	**'N'**	'S'	**'N'**	'P'	**'P'**	'E'	'E'	'H'	'F'	'M'	'M'	'F'	'F'
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**	'E'	'H'	'H'	'H'	'M'	'E'	'F'	'F'
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**	'E'	'E'	'H'	'H'	'M'	'H'	'F'	'F'
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**	'E'	'E'	'H'	'H'	'M'	'M'	'F'	'F'
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**	'E'	'H'	'H'	'F'	'M'	'H'	'F'	'F'
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'E'**	'E'	'E'	'H'	'H'	'M'	'M'	'F'	'F'

From the results above we can see that we have 34 errors in predicting and 70 correct one. So, the tree for dataset2 before feature selection has a 33% **ERR** and 67% **ACC.**

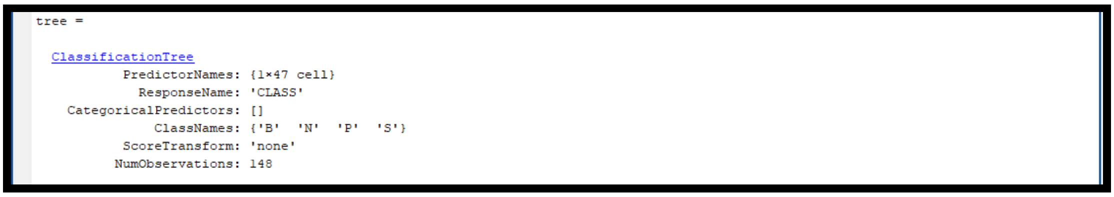
As we can see the ERR increased and ACC decreased for decision tree after using a big dataset and this is one of decision trees problem. We will make a feature selection to improve the performance and reduce the web vector.

Table 11 - ERR and ACC before feature extraction

Datasets before feature extraction	ERR	ACC
1	23%	77%
2	33%	67%

9.2.3.1.2 Decision tree result after feature selection

1) Decision tree for first dataset



```
tree =

  ClassificationTree
           PredictorNames: {1×47 cell}
             ResponseName: 'CLASS'
    CategoricalPredictors: []
               ClassNames: {'B'  'N'  'P'  'S'}
           ScoreTransform: 'none'
          NumObservations: 148
```

Figure 43 - Construction of DT for updated first dataset

We can see our decision tree for the first dataset after feature extraction, it has 47 PredictorNames(feature).

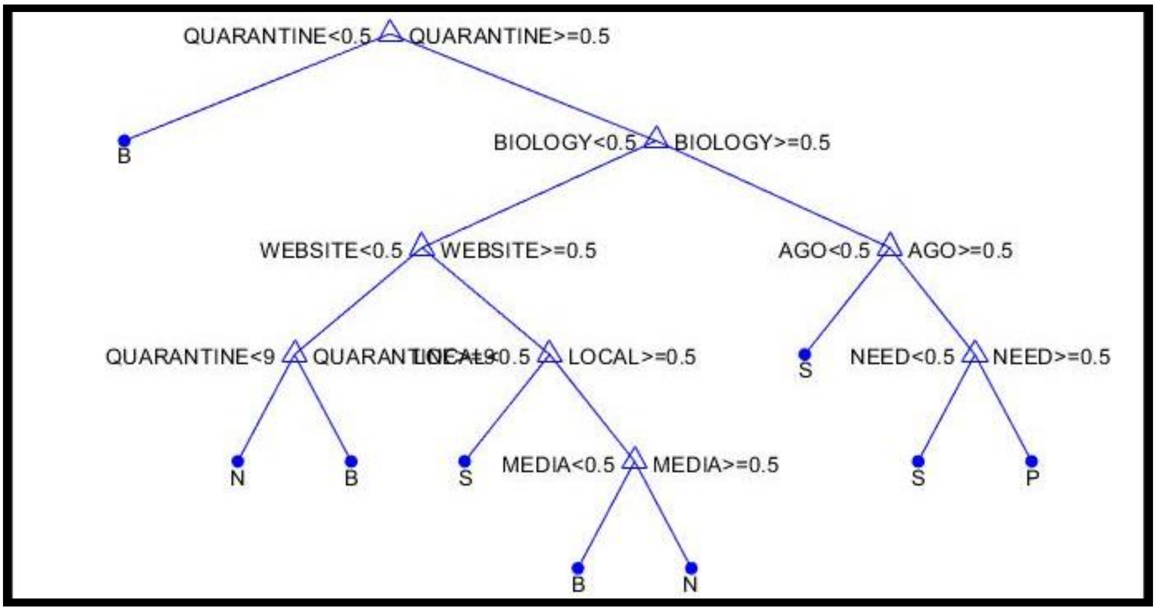


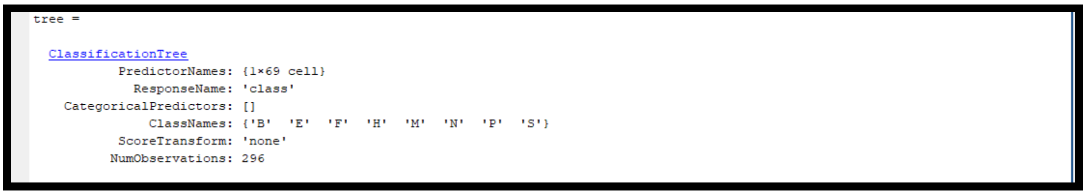
Figure 44 - Decision tree for updated first dataset

Table 12 - Prediction result for dataset1 after feature selection

Class	P	Class	P	Class	P	Class	P
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**
'B'	**'B'**	'N'	**'S'**	'S'	**'S'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**
'B'	**'B'**	'N'	**'S'**	'S'	**'P'**	'P'	**'P'**
'B'	**'B'**	'N'	**'S'**	'S'	**'P'**	'P'	**'P'**

From the results above we can see that we have 10 error in predicting and 42 correct one. So, the tree for dataset1 after feature selection has a 20% **ERR** and 80% **ACC.**

2) Decision tree for second dataset



```
tree =

  ClassificationTree
           PredictorNames: {1×69 cell}
             ResponseName: 'class'
    CategoricalPredictors: []
               ClassNames: {'B'  'E'  'F'  'H'  'M'  'N'  'P'  'S'}
           ScoreTransform: 'none'
          NumObservations: 296
```

Figure 45 - Construction of DT for updated second dataset

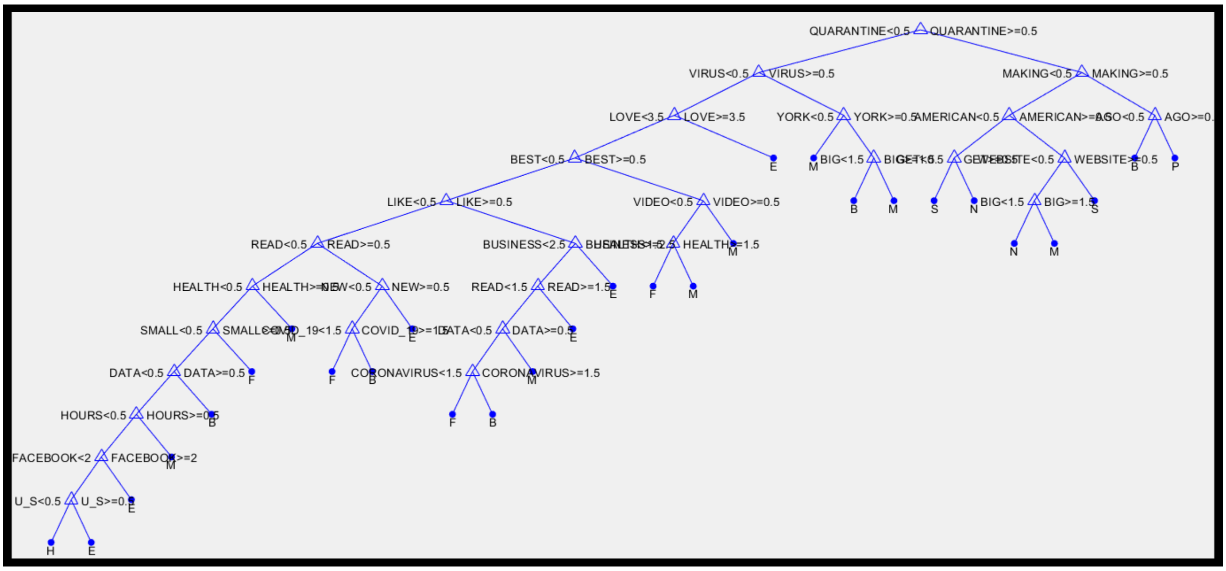


Figure 46 - Decision tree for second updated dataset

We can see from the tree how the feature extraction make the tree better than before. But we will test it to see the result if it really went good or not.

Table 13 - - Prediction result for dataset2 after feature selection

C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**	'E'	**'E'**	'H'	**'H'**	'M'	**'M'**	'F'	**'N'**
'B'	**'B'**	'N'	**'N'**	'S'	**'S'**	'P'	**'P'**	'E'	**'F'**	'H'	**'H'**	'M'	**'M'**	'F'	**'N'**
'B'	**'B'**	'N'	**'P'**	'S'	**'N'**	'P'	**'P'**	'E'	**'F'**	'H'	**'F'**	'M'	**'M'**	'F'	**'N'**
'B'	**'B'**	'N'	**'N'**	'S'	**'N'**	'P'	**'P'**	'E'	**'H'**	'H'	**'E'**	'M'	**'F'**	'F'	**'S'**
'B'	**'B'**	'N'	**'N'**	'S'	**'N'**	'P'	**'P'**	'E'	**'H'**	'H'	**'E'**	'M'	**'F'**	'F'	**'S'**
'B'	**'B'**	'N'	**'P'**	'S'	**'S'**	'P'	**'P'**	'E'	**'H'**	'H'	**'H'**	'M'	**'M'**	'F'	**'S'**
'B'	**'B'**	'N'	**'P'**	'S'	**'P'**	'P'	**'P'**	'E'	**'H'**	'H'	**'E'**	'M'	**'M'**	'F'	**'S'**
'B'	**'B'**	'N'	**'B'**	'S'	**'P'**	'P'	**'P'**	'E'	**'H'**	'H'	**'E'**	'M'	**'F'**	'F'	**'S'**
'B'	**'B'**	'N'	**'B'**	'S'	**'P'**	'P'	**'P'**	'E'	**'M'**	'H'	**'E'**	'M'	**'M'**	'F'	**'N'**
'B'	**'B'**	'N'	**'S'**	'S'	**'P'**	'P'	**'P'**	'E'	**'B'**	'H'	**'H'**	'M'	**'F'**	'F'	**'N'**
'B'	**'B'**	'N'	**'S'**	'S'	**'P'**	'P'	**'P'**	'E'	**'F'**	'H'	**'H'**	'M'	**'M'**	'F'	**'S'**
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**	'E'	**'F'**	'H'	**'H'**	'M'	**'H'**	'F'	**'S'**
'B'	**'B'**	'N'	**'N'**	'S'	**'P'**	'P'	**'P'**	'E'	**'H'**	'H'	**'H'**	'M'	**'M'**	'F'	**'F'**

From the results above we can see that we have 52 error in predicting and 52 correct one. So, the tree for dataset2 after feature selection has a 50% **ERR** and 50% **ACC.**

9.2.3.1.3 Final result

We applied future selection to reduce our web victor to get better results, and prediction. For dataset1 and 2 web vectors. And we can see (**Table.14**), shows all result for training, predicting, ERR and ACC for two different datasets using decision tree induction.

Table 14 - Final decision tree result before feature extraction

DS	Before feature selection				
	Training	**Testing**	**Feature**	**ERR**	**ACC**
1	148 record	52 record	152	23%	77%
	After feature selection				
	148 record	52 record	47	20%	80%
2	**Before feature selection**				
	Training	**Testing**	**Feature**	**ERR**	**ACC**
	296 record	104 record	211	33%	67%
	After feature selection				
	296 record	104 record	69	50%	50%

We can see how feature selection make dataset1 works better with less errors and more accuracy, but not the same with dataset2, because with increasing of training set the tree need to handle more predictors and decision tree is not made to classify big datasets, so it's mot preferable for classifying webpage, because WWW has billions of webpages and DT won't able handle this amount of data.

9.2.4 Artificial Neural network classification

Neural networks have emerged as an important tool for classification. The recent research activities in neural classification have established that Artificial neural networks(**ANN**) are a promising alternative to various conventional classification methods [18].

9.2.4.1 Construction of a Neural Networks

The structure and operation of a neural network can be described as follows: First, the abstract model of a neural network consists of neurons, also called units or nodes. They can pick up information from outside or from other neurons and pass it on to other neurons or output it as a final result. Basically, a distinction can be made between input neurons, hidden neurons and output neurons. The input neurons receive information in the form of patterns or signals from the outside world. The hidden neurons are located between the input and output neurons, and map internal information patterns. The output neurons relay information and signals to the outside world as a result. The different neurons are connected to each other via edges. Thus, the output of one neuron can become the input of the next neuron. Depending on the strength and meaning of the connection, the edge has a certain weighting. The stronger the weighting, the greater the influence a neuron can exert on the connection to another neuron.

9.2.4.2 Feed Forward - Back-Propagation

Feed Forward Neural Networks In these types of networks information flows in only one direction i.e. from input layer to output layer. When the weights are once decided, they are not usually changed. One either explicitly decides weights or uses functions decide weights. The nodes here do their job without being aware whether results produced are accurate or not, there is no communication back from the layers ahead.

Back-Propagation In these types of networks Information passes from input layer to output layer to produce result. Error in result is communicated back to previous layers. Now nodes get to know how much they contributed in the answer being wrong. Weights are re-adjusted. ANN is improved. It learns. There is bi-directional flow of information. This basically has both algorithms implemented, feed-forward and back-propagation.

9.2.4.3 ANN Result

With the rapid growth of the www, there is an increasing need to provide automated assistance to web users for web page categorization [19] [20]. As we have observed that a user's fails to find out their intended web site if the pages are classified only on limited features and similarities. This work covering 4 major classes of web pages including Business, News, Science, Sport. The websites in dataset are classified based on a vector of frequency of keywords from 200 website. The dataset has 3 frequent words in each one of websites then we make a filtering process for our website feature vector to be decreased from 600 to 152 per website. In dataset we collect there's 200 websites, 25% for each of the 4 categories Business, News, Sport, Science. We trained 75% of the dataset and tested 25% of it.

9.2.4.3.1 Proposed Network Architecture

We have created an ANN network that takes 152 inputs as one vector, 2 hidden layers with 12 neurons and 2 output nodes.

The following tables shows the used class labels corresponding the output used in ANN network.

Table 15 - ANN Class labels and outputs for network 1

Class label	ANN output
Business	**00**
News	**01**
Science	**10**
Sport	**11**

9.2.4.3.2 ANN Network structure

We have created a network that takes 152 of features vector as input, with 2 hidden layers each has 12 neurons and 2 output nodes. The following figure shows the structure of the network.

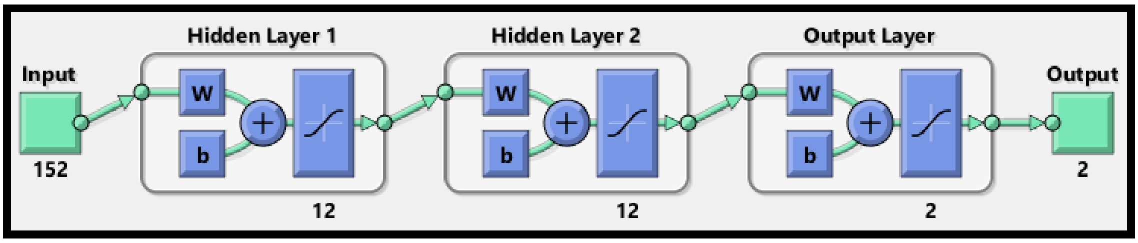


Figure 47 - ANN structure

9.2.4.3.3 Training and Testing Data

Firstly, we used 100 website vectors for training our network, through 8 Epochs. Then test it using 13 websites from each of 4 categories with equal of 52 websites for testing set.
Here are the plot figures after training the network, Performance and Training state.

9.2.4.3.3Network Performance and Training state plots

The following two figures shows the performance and the state while training our network. In performance figure we have three-lines, MATLAB takes dataset and split it into three parts, 70% for training, 15% for validation, and 15% for testing while training. The **Train** shows the training performance based on Mean squared error(MSE), **MSE** is the average squared difference between the estimated values and the actual value. Training set used for learning the parameters of the model [i.e., weights]. **Validation** is the data is used during training to assess how well the network is currently performing and used to learning the hyper-parameters [i.e. Learning Rate]. It is aimed to avoid over-fitting problem. Test used to assess the performance of a trained network.

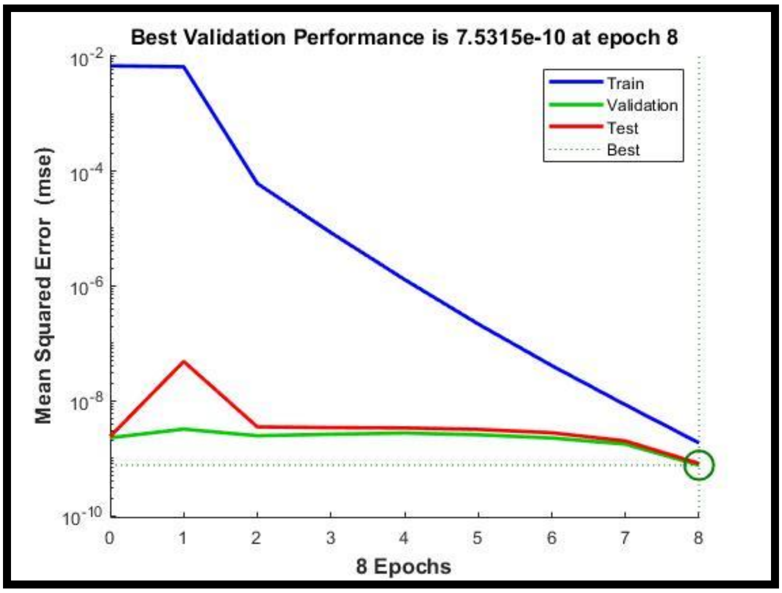


Figure 48 - Performance of training network1

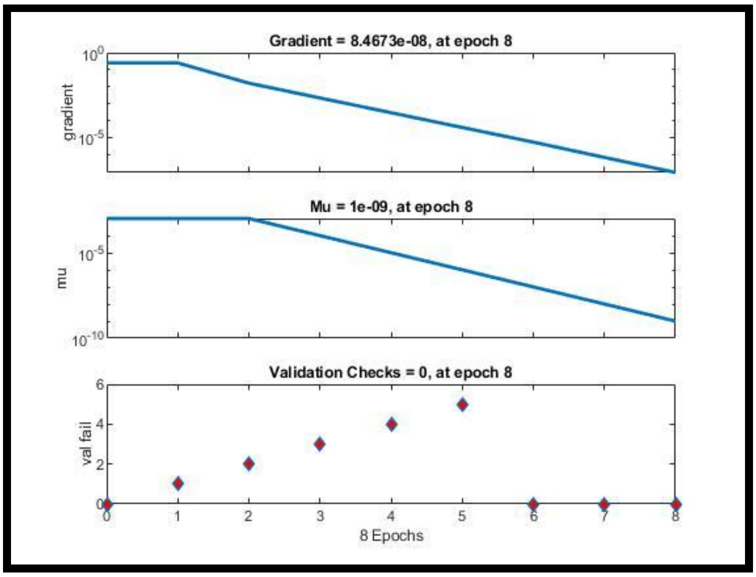
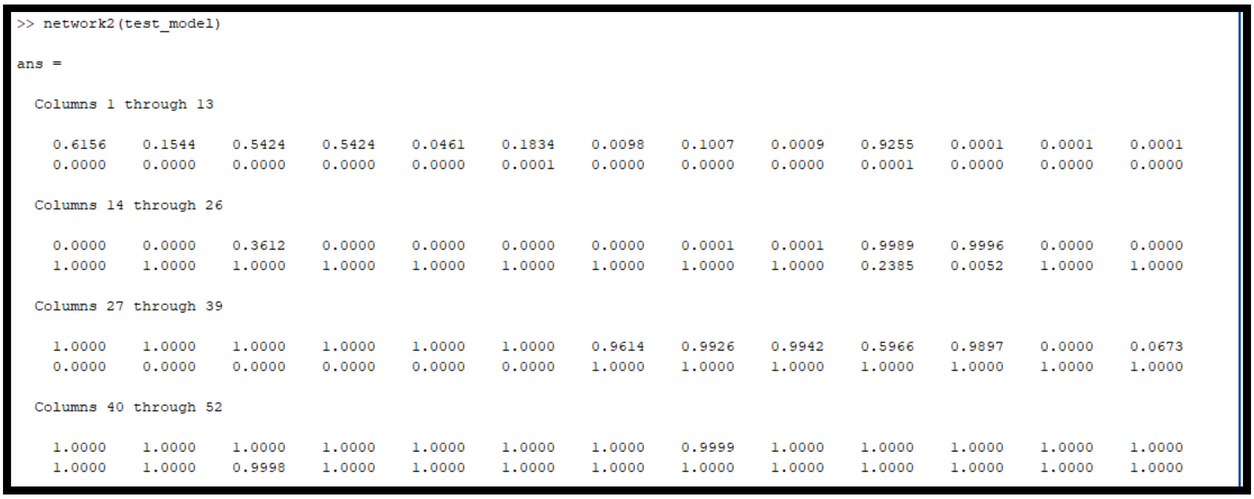


Figure 49 - Training state of training network1

9.2.4.4ANN Testing and results

We have used 25% of our dataset to test our trained network. The test model contains 52 website vectors, corresponding 152 features. First 13 sites are business sites, the second 13 are news, third 13 are science, and last 13 sites are sport.

The following figure shows the result of predicting the test model, each two rows on one column represents the output class for a website in text model, ex. In column1 (0.6156 and 0.0000) is the output for the first website in test model, and so on.



```
>> network2(test_model)

ans =

  Columns 1 through 13

    0.6156    0.1544    0.5424    0.5424    0.0461    0.1834    0.0098    0.1007    0.0009    0.9255    0.0001    0.0001    0.0001
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0001    0.0000    0.0000    0.0000    0.0001    0.0000    0.0000    0.0000

  Columns 14 through 26

    0.0000    0.0000    0.3612    0.0000    0.0000    0.0000    0.0000    0.0001    0.0001    0.9989    0.9996    0.0000    0.0000
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    0.2385    0.0052    1.0000    1.0000

  Columns 27 through 39

    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    0.9614    0.9926    0.9942    0.5966    0.9897    0.0000    0.0673
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000

  Columns 40 through 52

    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    0.9999    1.0000    1.0000    1.0000    1.0000    1.0000
    1.0000    1.0000    0.9998    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
```

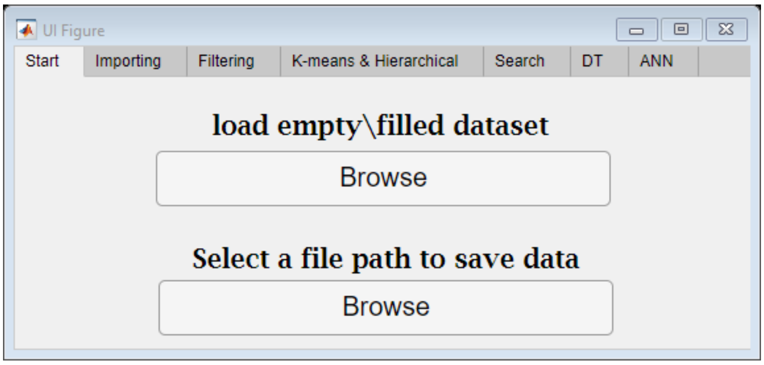
Figure 50 – ANN network1 result

As we can see that our network predicts all class for its inputs from test model. From the output we have 11 errors, the business sites have been predicting it by 100%, News has 2 predicting errors out of 13, Science has 7 errors out of 13, finally Sport has 2 predicting errors out of 13.

So on, we can calculate the error rate and the accuracy of our network in predicting from the tested model. ERR is 21% and ACC is 79% for this network with 4 classes.

Chapter 10: WPC interface

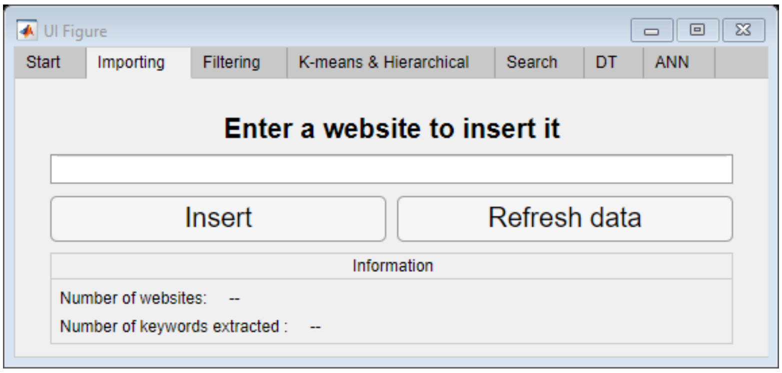
- In this chapter, we will list some interfaces for our GUI for out project. Everything explained and listed in user's manual.



load empty\filled dataset

Select a file path to save data

Figure 51 - Main screen

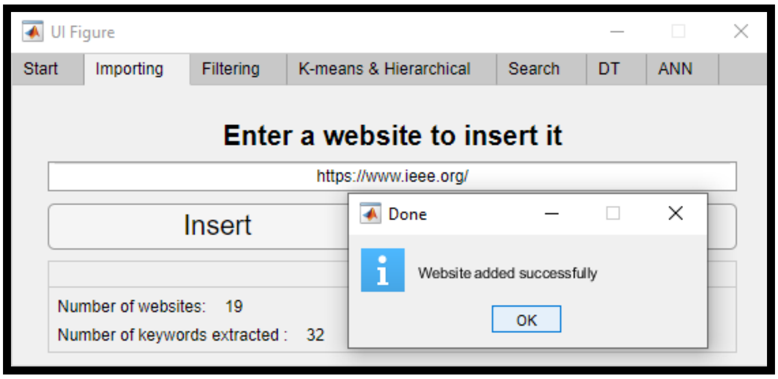


Enter a website to insert it

Number of websites: --

Number of keywords extracted : --

Figure 52 - Importing screen



Enter a website to insert it

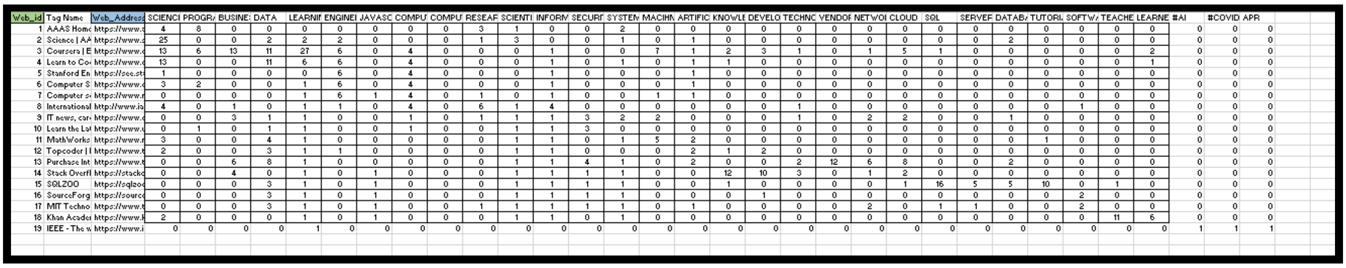
https://www.ieee.org/

Website added successfully

Number of websites: 19

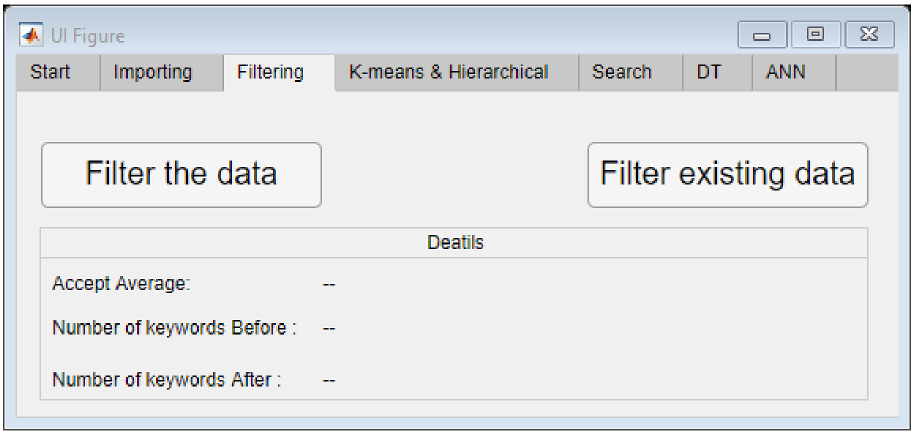
Number of keywords extracted : 32

Figure 53 - After importing a website into dataset



Web_id	Tag Name	Web_Address	SCIENCI	PROGR/	BUSINE:	DATA	LEARNII	ENGINEI	JAVASC	COMPU	COMPU	RESEAF	SCIENTI	INFORM	SECURI	SYSTEM	MACIHN	ARTIFIC	KNOWLE	DEVELO	TECHNC	VENDOF	NETWOI	CLOUD	SQL	SERVEF	DATAB/	TUTORI	SOFTW/	TEACHE	LEARNE	#AI	#COVID	APR
1	AAAS Home	https://www.s	4	8	0	0	0	0	0	0	0	3	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Science | AA	https://www.s	25	0	0	2	2	2	0	0	0	1	3	0	0	1	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
3	Coursera | E	https://www.c	13	6	13	11	27	6	0	4	0	0	0	1	0	0	7	1	2	3	1	0	1	5	1	0	0	0	0	0	2	0	0	0
4	Learn to Co	https://www.c	13	0	0	11	6	6	0	4	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
5	Stanford En	https://see.st	1	0	0	0	0	6	0	4	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	Computer S	https://www.c	3	2	0	0	1	6	0	4	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Computer s	https://www.n	0	0	0	0	1	6	1	4	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	International	http://www.ia	4	0	1	0	1	1	0	4	0	6	1	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
9	IT news, car	https://www.c	0	0	3	1	1	0	0	1	0	1	1	1	3	2	2	0	0	0	1	0	2	2	0	0	1	0	0	0	0	0	0	0
10	Learn the La	https://www.u	0	1	0	1	1	0	0	1	0	0	1	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	MathWorks	https://www.m	3	0	0	4	1	0	0	0	0	0	1	1	0	1	5	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
12	Topcoder | I	https://www.t	2	0	0	3	1	1	0	0	0	0	1	1	0	0	0	2	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	Purchase Int	https://www.t	0	0	6	8	1	0	0	0	0	0	1	1	4	1	0	2	0	0	2	12	6	8	0	0	2	0	0	0	0	0	0	0
14	Stack Overfl	https://stackc	0	0	4	0	1	0	1	0	0	0	1	1	1	1	0	0	12	10	3	0	1	2	0	0	0	0	0	0	0	0	0	0
15	SQLZOO	https://sqlzo	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0	0	1	16	5	5	10	0	1	0	0	0	0
16	SourceForg	https://source	0	0	0	3	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0
17	MIT Techno	https://www.t	0	0	0	3	1	0	1	0	0	1	1	1	1	1	0	1	0	0	0	0	2	0	1	1	0	0	2	0	0	0	0	0
18	Khan Acade	https://www.k	2	0	0	0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	11	6	0	0	0
19	IEEE - The w	https://www.i	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Figure 54 - Dataset after importing

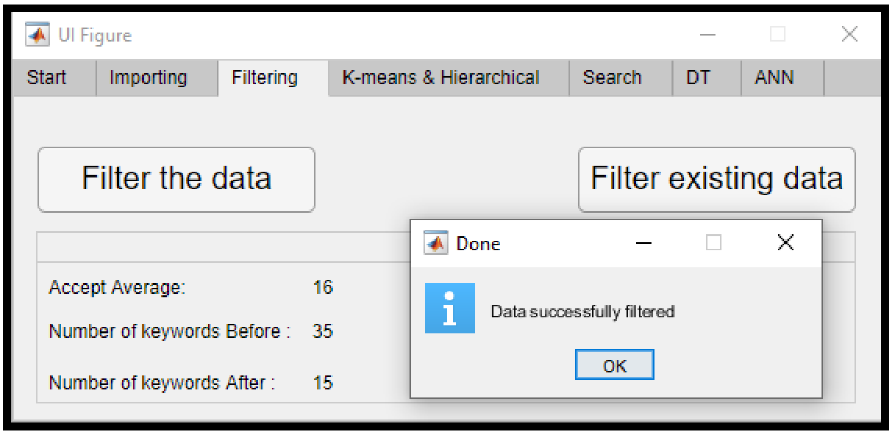


Accept Average: --

Number of keywords Before : --

Number of keywords After : --

Figure 55 - Filtering screen



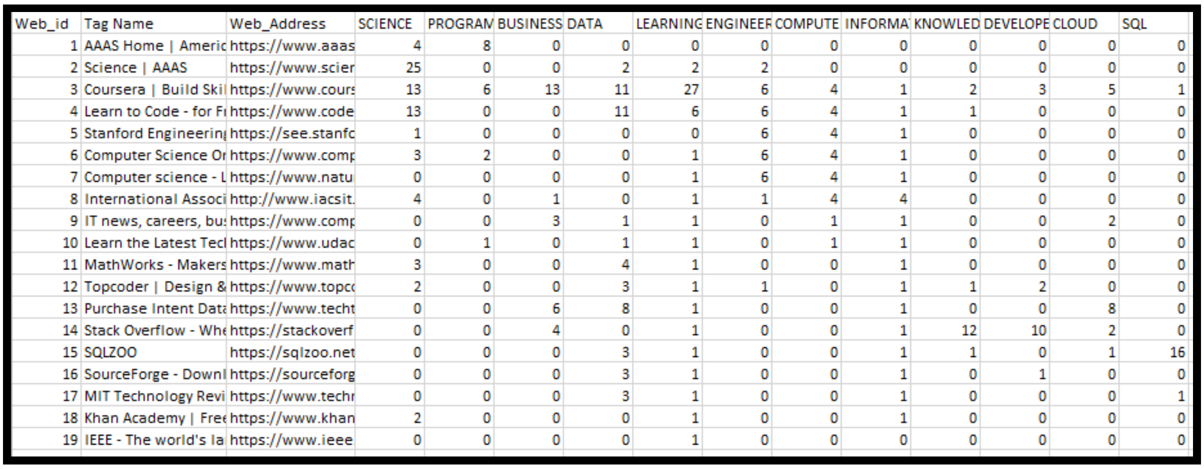
Accept Average: 16

Number of keywords Before : 35

Number of keywords After : 15

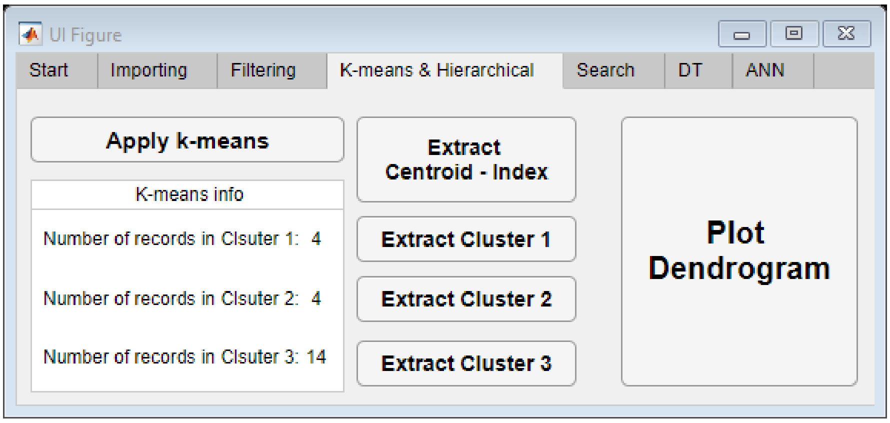
Data successfully filtered

Figure 56 - After filtering



Web_id	Tag Name	Web_Address	SCIENCE	PROGRAM	BUSINESS	DATA	LEARNING	ENGINEER	COMPUTE	INFORMA	KNOWLED	DEVELOPE	CLOUD	SQL
1	AAAS Home | Americ	https://www.aaas	4	8	0	0	0	0	0	0	0	0	0	0
2	Science | AAAS	https://www.scier	25	0	0	2	2	2	0	0	0	0	0	0
3	Coursera | Build Skil	https://www.cours	13	6	13	11	27	6	4	1	2	3	5	1
4	Learn to Code - for Fr	https://www.code	13	0	0	11	6	6	4	1	1	0	0	0
5	Stanford Engineering	https://see.stanfc	1	0	0	0	0	6	4	1	0	0	0	0
6	Computer Science Or	https://www.comp	3	2	0	0	1	6	4	1	0	0	0	0
7	Computer science - L	https://www.natu	0	0	0	0	1	6	4	1	0	0	0	0
8	International Associ	http://www.iacsit.	4	0	1	0	1	1	4	4	0	0	0	0
9	IT news, careers, bus	https://www.comp	0	0	3	1	1	0	1	1	0	0	2	0
10	Learn the Latest Tecl	https://www.udac	0	1	0	1	1	0	1	1	0	0	0	0
11	MathWorks - Makers	https://www.math	3	0	0	4	1	0	0	1	0	0	0	0
12	Topcoder | Design &	https://www.topcc	2	0	0	3	1	1	0	1	1	2	0	0
13	Purchase Intent Data	https://www.techt	0	0	6	8	1	0	0	1	0	0	8	0
14	Stack Overflow - Whe	https://stackoverf	0	0	4	0	1	0	0	1	12	10	2	0
15	SQLZOO	https://sqlzoo.net	0	0	0	3	1	0	0	1	1	0	1	16
16	SourceForge - Downl	https://sourceforg	0	0	0	3	1	0	0	1	0	1	0	0
17	MIT Technology Revi	https://www.techr	0	0	0	3	1	0	0	1	0	0	0	1
18	Khan Academy | Free	https://www.khan	2	0	0	0	1	0	0	1	0	0	0	0
19	IEEE - The world's la	https://www.ieee	0	0	0	0	1	0	0	0	0	0	0	0

Figure 57 - Dataset after filtering



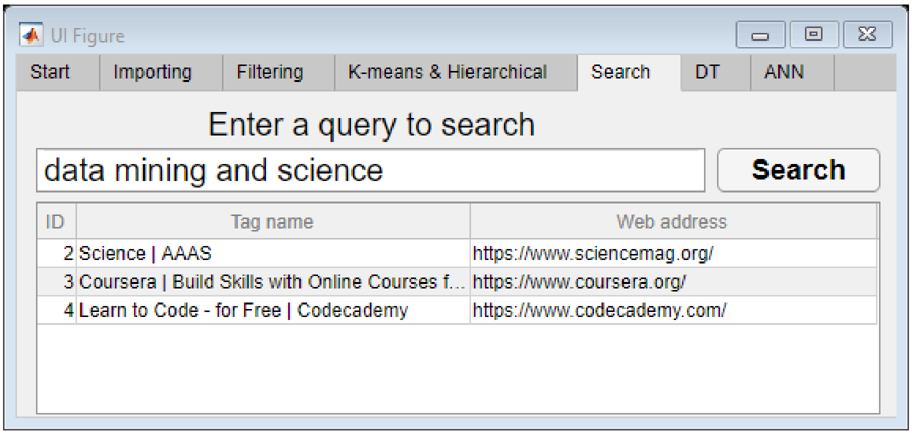
K-means info

Number of records in Clsuter 1: 4

Number of records in Clsuter 2: 4

Number of records in Clsuter 3: 14

Figure 58 - kmeans & Hierarchical screen



data mining and science

ID	Tag name	Web address
2	Science | AAAS	https://www.sciencemag.org/
3	Coursera | Build Skills with Online Courses f...	https://www.coursera.org/
4	Learn to Code - for Free | Codecademy	https://www.codecademy.com/

Figure 59 - Search screen (IR)

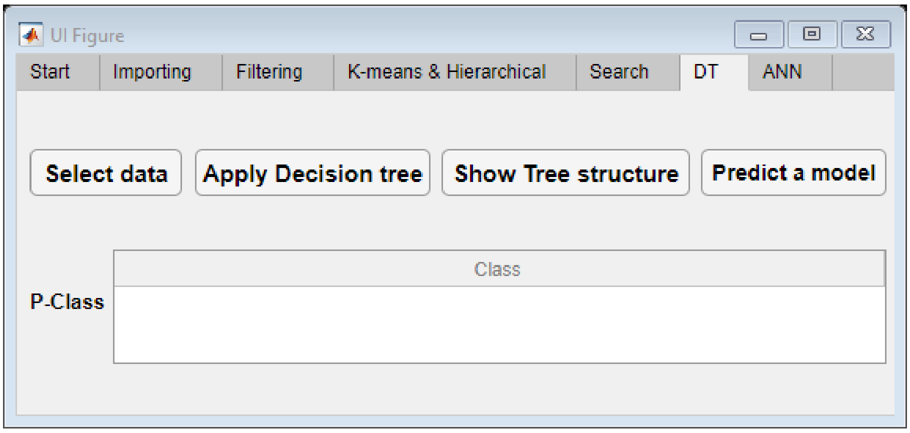


Figure 60 - DT screen

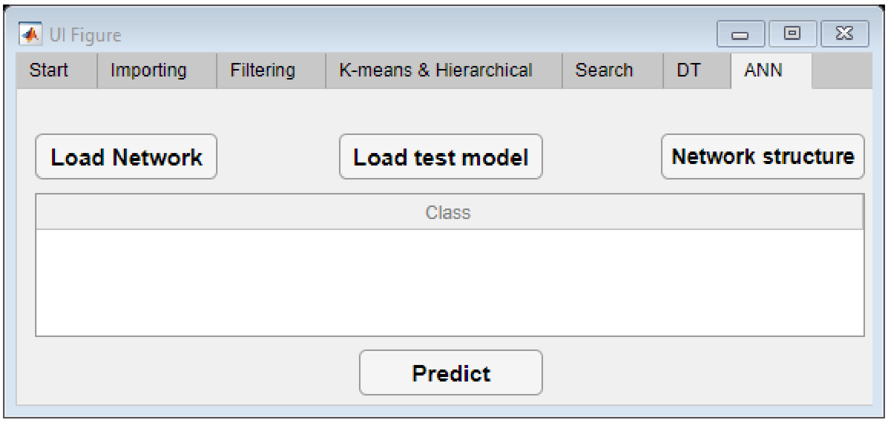


Figure 61 - ANN screen

Chapter 11: Summary

We can say that ANN is more effective than decision tree even at classifying big data like webpages, and we can see the differences in error rate and how accurate for both in the following table.

Table 16 - ANN vs DT

Algorithm	Before feature selection				After feature selection			
	ERR		ACC		ERR		ACC	
	DS1	DS2	DS1	DS2	DS1	DS2	DS1	DS2
Decision tree	23%	33%	77%	67%	20%	50%	80%	50%
ANN	21%	---	79%	---	---	---	---	---

We can see that decision tree did good with dataset1 after feature selection, the error decreased by 3%, and the accuracy increased by 3% but not as good as ANN. And decision tree with dataset2 wasn't good at all, it got 33% error rate and 67% Accuracy, even after feature selection it becomes worse. The error rate increased by 17% and the accuracy decreased by 17%.

At the end of this work, web page categorization/classification is one of the challenging tasks in the world of ever-increasing web technologies. There are many ways of categorization of web pages based on different approach and features. This project will develop the way of categorization of web pages using Decision tree and ANN algorithms through extracting the features automatically. Here eight major categories of web pages selected for categorization; these are Business, Sports, News, Science, Education, History, Medical, and Food.

Automated categorization of web pages can lead to better web retrieval tools with the added convenience of selecting among properly organized directories. Web page classification is proposed which extract the features automatically through analyzing the html source and categorize the web pages into four major classes.

References

[1]. Choi B (2001) Making sense of search results by automatic web-page classification. In: WebNet 2001. Orlando, Florida, USA, pp 184-186.

[2]. Machine Learning for Web Page Classification: A Survey (S.Lassri, H.Benlahmar, A.Tragha) at https://innove.org/ijist/index.php/ijist/article/download/137/74

[3] https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[4]. «36 798 Search Results - Keywords (web page classification) - ScienceDirect » url:https://www.sciencedirect.com/search?qs=web%20page%20classification&show=25 &sortBy=relevance .

[5]. (Search Results – Springer) https://link.springer.com/search?query=web+page+classification

[6]. Luhn HP (1958) The automatic creation of literature abstracts. IBM Journal of Research and Development 2: 159-165.

[7]. T. Xia, Y. Chai, "Improving SVM On Web Content Classification By Document Formulation", In Proceedings Of The 7th International Conference On Computer Science & Education (ICCSE '12), Australia, 14-17 July 2012, Pp. 110-113.

[8]. https://www.semanticscholar.org/paper/Data-Mining%3A-Web-Data-Mining-Techniques%2C-Tools-and-Mughal/f6058601bf4cc73be92d01d0e442cd611d41d403

[9]. Web Page Classification in Web Mining Research – (A SurveyE.Suganya1, Dr.S.Vijayarani2)

[10]. https://pdfs.semanticscholar.org/e1df/beb415a15e82c18c55a689f53cf56b64d8dc.pdf

[11]. Data Mining: Web Data Mining Techniques, Tools and Algorithms: An Overview (Muhammd Jawad Hamid)

[12]. https://pdfs.semanticscholar.org/a4be/30ed3010294e9ba5ce7006ef4157bd78610e.pdf

[13]. 5

[14]. Buckley, C., Salton, G.: Term Weighting Approaches in Automatic Text Retrieval. Inf. Process. Manage 24(5), 513–523 (1988)

[15]. Keselj, V., Milios, E., Miao, Y.: Document clustering using character N-grams: a comparative evaluation with term-based and word-based clustering. In: 14th ACM International Conference on Information and Knowledge Management, New York, USA, pp. 357–358 (2005)

[16]. 9. Grouper, E.O., Zamir, O.: A Dynamic Clustering Interface to Web Search Results. In: Eighth International World Wide Web Conference, pp. 283–296 (1999)

[17]. https://www.edureka.co/blog/classification-algorithms/

[18]. Parsons1- Guoqiang Peter Zhang, Neural Networks for Classifications: A Survey, IEEE Transactions on Systems, Man, And Cybernetics—Part C: Applications and Reviews, Vol. 30, No. 4, (November 2000).

[19]. H.Yu, J.Han, and K.C.C.Chang.Pebl, "Positiveexample based learning for web page categorization using SVM", In KDD, Edmonton, Alberta, Canada, 2002.

[20]. Hui Yang & Tat-Seng Chua " Effectiveness of web page categorization on Finding List Answer ", National University of Singapore.