"NOISE REDUCTION IN WEB DATA"

A MAJOR PROJECT REPORT

Submitted in partial fulfillment of the
Requirement for the award of Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted by

KASAGONI JAGRUTHI (20JJ1A0522)
BAIRI SRIKRISHNA (20JJ1A0508)
KOLAPUDI KAVITHA (20JJ1A0523)
MULKALA ANUSHA(20JJ1A0533)

Under the Esteemed Guidance of
P.SREENIVASA RAO

Associate Professor, Depatment of CSE.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Accredited by NBA: UG (CSE))

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY
HYDERABAD UNIVERSITY COLLEGE OF ENGINEERING JAGTIAL

(Autonomous and Accredited By NAAC with A+ Grade)

Nachupally (Kondagattu), Jagtial Dist. -505501. T.S
2023-2024

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

HYDERABAD UNIVERSITY COLLEGE OF ENGINEERING JAGTIAL

(Autonomous and Accredited By NAAC with A+ Grade)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Accredited by NBA: UG(CSE))

CERTIFICATE

This is to certify that the project work entitled **"NOISE REDUCTION IN WEBDATA"** is a bonafide work carried out by **KASAGONI JAGRUTHI(20JJ1A0522), BAIRI SRIKRISHNA(20JJ1A0508) ,KOLAPUDI KAVITHA(20JJ1A0523), MULAKALA ANUSHA (20JJ1A0533)** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ENGINEERING** by the Jawaharlal Nehru Technological University Hyderabad during the academic year 2023-2024.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

Internal Project Guide	Head of the Department
P.SREENIVASA RAO ,	**Dr. B. SATEESH KUMAR,**
ASSOCIATE Professor of CSE	**Professor of CSE**

....

External Examiner

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD UNIVERSITY COLLEGE OF ENGINEERING JAGTIAL

(Autonomous and Accredited By NAAC with A+ Grade)

DECLARATION

We hereby declare that the Project work entitled **“NOISE REDUCTION IN WEB DATA”** submitted in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** which was carried out under the supervision of **Dr. P. SREENIVASA RAO, ASSOCIATE PROFESSOR**, Department of CSE, JNTUH UCEJ.

Also, we declare that the matter embedded in the thesis have not been submitted by me in full or partial thereof to any other University or Institute for the award of any degree previously.

KASAGONI JAGRUTHI(20JJ1A0522)

BAIRI SRIKRISHNA(20JJ1A0508)

KOLAPUDI KAVITHA(20JJ1A0523)

MULKALA ANUSHA(20JJ1A0533)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

I wish to express my deep sense of gratitude to **Dr. P.SREENIVASA RAO**, Professor of CSE, Project guide, for his able guidance and useful suggestions which helped me in completing the project work intime.

I am extremely grateful to our HOD- **Dr. B. SATEESH KUMAR**, Professor of CSE& Head of the Department, for his immense support and cooperation that contributed a lot in the completion of the task.

We show gratitude to our beloved Principal, I/c **Dr. BITLA PRABHAKAR** , Professor of ECE for providing necessary Infrastructure and resources for the accomplishment of our project report at JNTUH UCEJ.

I show our immense gratitude towards our college library for providing us with necessary references required for completing this project.

I also thank all the staff members of Computer Science & Engineering department, JNTUH University College of Engineering, Jagtial for their valuable support and generous advice.

Finally thanks to my parents, all my family members and friends for their continuous support and enthusiastic help.

KASAGONI JAGRUTHI(20JJ1A0522)
BAIRI SRIKRISHNA(20JJ1A0508)
KOLAPUDI KAVITHA(20JJ1A0523)
MULKALA ANUSHA(20JJ1A0533)

ABSTRACT

One of the significant issues facing web users is the amount of noise in web data which hinders the process of finding useful information in relation to their dynamic interests. Current research works consider noise as any data that does not form part of the main web page and propose noise web data reduction tools which mainly focus on eliminating noise in relation to the content and layout of web data. This paper argues that not all data that form part of the main web page is of a user interest and not all noise data is actually noise to a given user. Therefore, learning of noise web data allocated to the user requests ensures not only reduction of noisiness level in a web user profile, but also a decrease in the loss of useful information hence improves the quality of a web user profile.

Noise Web Data Learning (NWDL) tool/algorithm capable of learning noise web data in web user profile is proposed. The proposed work considers elimination of noise data in relation to dynamic user interest. In order to validate the performance of the proposed work, an experimental design setup is presented. The results obtained are compared with the current algorithms applied in noise web data reduction process. The experimental results show that the proposed work considers the dynamic change of user interest prior to elimination of noise data. The proposed work contributes towards improving the quality of a web user profile by reducing the amount of useful information eliminated as noise.

This explores how current available tools address problems with noise in web user profile. We establish that current research works eliminate noise from web data mainly based on the structure and layout of web pages i.e. they consider noise as any data that does not form part of the main web page. However, not all data that form part of main web page is of a user interest and not every data considered noise is actually noise to a given user. The ability to determine what is noise and useful to a dynamic web user profile has not been fully addressed by current research works. We aim to justify a claim that it is important to learn noise prior to elimination, to not only decrease levels of noise but also reduce loss of useful information. This is because if noise in web data is not clearly defined and analysed through learning, the purpose and its use will be compromised hence its overall quality.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.INTRODUCTION

1.1PROJECT OVERVIEW

Nowadays the web is widely used in every aspect of day to day life, a daily use of web means that users are searching for useful information. However, ensuring useful information is available to a specific user has become a challenging issue due to the amount of noise data present on the web. Noise in web data is defined as any data that is not part of the main content of a web page. For example, advertisements banners, graphics, web page links from external web sites etc. Noise web data elimination is a concept which involves detection of web data that needs to be eliminated because it either does not form part of the main web page content or is not useful to a given user. It is recognised in the current research work that the noise web data reduction process is site-specific, i.e. it involves removal of external web pages that do not form part of the main web page content. However, this work does not focus on the structure and layout of web data to identify and eliminate noise but instead, a key focus is on extracted web log data that defines a web user profile.

In view of this research, noise is not necessarily advertisements from external web pages, duplicate links and dead URLs or any data that does not form a part of the main content of a web page, but also useful information that does not reflect dynamic changes in user interests. Various machine learning tools/algorithms are used to discover useful information from web data, this process is referred to as web usage/data mining process. It finds user interest patterns from web log data. Web log data contains a list of actions that have occurred on the web based on a user. These log files give an idea about what a user is interested in available web data. Web log data contain basic information such as IP address, user visit duration and visiting path, web page visited by the user, time spent on each web page visit etc. In this work, web log file and web data are used interchangeably because a log file contains web data, therefore elimination of noise web data is based on extracted web user log file. A web user profile is defined as a description of user interests, characteristics, and preferences on a given website. User interests can be implicit or explicit. Explicit interests are where a user tell the system what his/her interests are and what they think about available web data while implicit interest is where the system automatically finds interests of a user through various means such as time and frequency of web page visits. Many users may not be willing to tell the system what their true intentions are on available web data, therefore, this work will focus on implicit user interests. Current research efforts in noise web data reduction have worked with the assumption that the web data is static. For example, proposed a mechanism where noise detected from web pages is matched by stored noise data for classification and subsequent elimination. Therefore, it shows that elimination of noise in web data is based on pre-existing noise data patterns. In evolving web data, existing noise data patterns used to identify and eliminate noise from web data may become out of date. For this reason, the dynamic aspects of user interest have recently become important. Moreover, web access patterns are dynamic not only due to evolving web data but also due to changes in user interests. For example, web users are likely to be interested in data derived from events such as Weddings, Christmas, Birthdays etc. Therefore, it is necessary to discover where such dynamic tendencies impact the process of eliminating noise from web data.

1.2PROBLEM STATEMENT

Ensuring useful information is available toa specific user has become challenging issue due to the amount of noise data present on the web. Noise in web data is defined as any data that is not part of the main content of a web page. For example, advertisements banners, graphics, webpage links from external web sites etc. Noise web data elimination is a concept which involves detection of web data that needs to be eliminated because it either does not form part of the main webpage content or is not useful toa given user. It is recognised in the current research work that the noise web data reduction process is site-specific, i.e. it involves removal of external web pages that do not form part of the main web page content.

1.3 PURPOSE

The main purpose of this project is to identify Noise Reduction in web data A learning approach based on dynamic user interests. However, ensuring useful information is available to a specific user has become a challenging issue due to the amount of noise data present on the web. Noise in web data is defined as any data that is not part of the main content of a web page. For example, advertisements banners, graphics, web page links from external web sites etc. Noise web data elimination is a concept which involves detection of web data that needs to be eliminated because it either does not form part of the main web page content or is not useful to a given user.

2.LITERATURE REVIEW

2.1 SURVEY DETAILS

Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data: Web usage mining is the application of data mining techniques to discover usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Web usage mining consists of three phases, namely preprocessing, pattern discovery, and pattern analysis. This paper describes each of these phases in detail. Given its application potential, Web usage mining has seen a rapid increase in interest, from both the research and practice communities. This paper provides a detailed taxonomy of the work in this area, including research efforts as well as commercial offerings. An up-to-date survey of the existing work is also provided. Finally, a brief overview of the WebSIFT system as an example of a prototypical Web usage mining system is given. Extracting Users'Navigational Behavior from Web Log Data: a Survey: Web Usage Mining (WUM) is a kind of data mining method that can be used to discover user access patterns from Web log data. A lot of research has been done already about this area and the obtained results are used in different applications such as recommending the Web usage patterns, personalization, system improvement and business intelligence. WUM includes three phases that are called preprocessing, pattern discovery and pattern analysis. There are different techniques for WUM that have their own advantages and disadvantages. This paper presents a survey on some of the existing WUM techniques and it is shown that how WUM can be applied to Web server logs.

A Survey On Web Log Mining And Pattern Prediction:

Web sites have abundant web usage log which provides great source of knowledge that can be used for discovery and analysis of user accessibility pattern. The web log mining is the process of identifying browsing patterns by analyzing the user's navigational behaviour. The web log files which store the information about the visitors of web sites is used as input for web log mining and pattern prediction process. First these log files are pre-processed and converted into required formats so web usage mining techniques can apply on these web logs for frequent patternsWeb user interest prediction framework based on user behavior for dynamic websites: We develop a framework to predict the user interest based on the behavior of user to increase the efficiency of dynamic websites. The content management in the dynamic website is difficult because it varies with the user profiles, i.e. different contents have to be placed for different users according to the user profiles. Various ways have been identified earlier to track the user interest but lacks with the accuracy here we propose a new one which composes both implicit and explicit. We track all behaviors like time of visit, navigation URL, web logs, user actions on the web page. Our model uses the web log data of the user and also tracks the implicit behaviors performed by the user.

2.2 EXISTING SYSTEM

Various machine learning tools/algorithms are used to discover useful information from web data, this process is referred to as web usage/data mining process. It finds user interest patterns from web log data. Web log data contains a list of actions that have occurred on the web based on a user. These log files give an idea about what a user is interested in available web data. Web log data contain basic information such as IP address, user visit duration and visiting path, web page visited by the user, time spent on each web page visit etc. In this work, web log file and web data are used interchangeably because a log file contains web data, therefore elimination of noise web data is based on extracted web user log file. Now-a-days almost all users are using web pages to get various information such as news, sports, technology etc but all web pages will use noise data such as images, video clips or advertisement which makes difficult for the users to get interested information. To remove noise data all existing technologies were using static web matching pattern such as the main page look and feel will be match with rest of the screen and if not match then it will remove unmatched data from the web pages to show only interested data to the user. This static technique will not work if web pages look and feel changes dynamically.

2.3 PROPOSED SYSTEM

To overcome from above issue author is proposing Noise Web Data Learning (NWDL) technique, in this technique server will maintain log for each user access page and will be called as web log dataset. This dataset will have information such as User id, access page, date time, URL. By analyzing such log data, we can identify user interested pages in dynamic or static web pages. User interested pages can be found by seeing frequency of web page access by a single user and total time spend on each page. If user spend more time and access this page more than 2 times then we can consider that user is interested in that page. If user spend less time on seeing that page and visiting that page very rarely then it will consider a sum interested page and will becalled as noise page. For example, proposed a mechanism where noise detected from web pages is matched by stored noise data for classification and subsequent elimination. It shows that elimination of noise in web data is based on pre-existing noise data patterns. In evolving web data, existing noise data patterns used to identify and eliminate noise from web data may become out of date. For this reason, the dynamic aspects of user interest have recently become important. Moreover, web access patterns are dynamic not only due to evolving web data but also due to changes in user interests. For example, web users are likely to be interested in data derived from events such as Christmas, Birthdays etc. Therefore, it is necessary to discover where such dynamic tendencies impact the process of eliminating noise from web data. To propose a machine learning algorithm capable of learning noise in a web user profile prior to elimination. Elimination of noise from a web user profile does not only depend on pre-existing noise data patterns, but it learns noise levels based on dynamic changes in user interest as well as evolving web data.

3.SYSTEM ANALYSIS

These system requirements provide a technical foundation for the development, deployment, and maintenance of the system. They ensure that the system operates effectively, securely, and efficiently within its intended environment.

3.1 HARDWARE REQUIREMENTS

Processor	i5 and above
RAM	4GB and above
Hard Disk	20GB and above

Table 2(a) Hardware Requirements

3.2 SOFTWARE REQUIREMENTS

Operating System	Windows
Software	Python IDEL (Python 3.7)

Table 2(b) Software Requirements

4.SYSTEM DESIGN

Programming configuration sits at the specialized portion of the product designing procedure and is applied paying little mind to the advancement worldview and zone of use. Configuration is the initial phase in the advancement stage for any designed item or framework. The fashioner will likely deliver a model or portrayal of a substance that will later be assembled. Starting, when framework necessity has been indicated and dissected, framework configuration is the first of the three specialized exercises - plan, code and test that is required to construct and check programming.

4.1 SYSTEM ARCHITECTURE

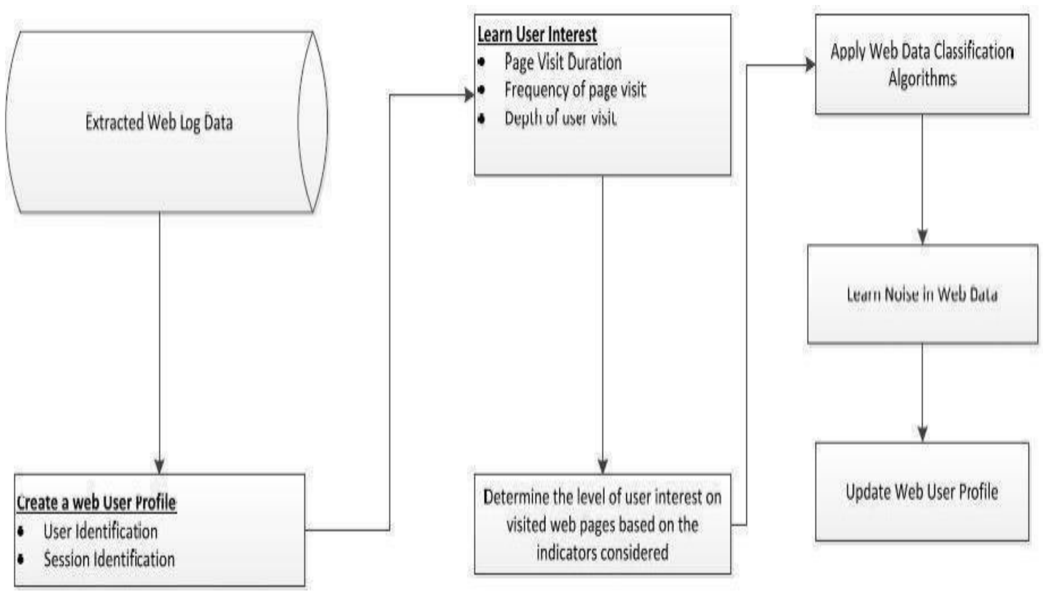


Fig 4.1.1 System architecture

In this section, a machine learning algorithm capable of learning noise in a web user profile prior to elimination is proposed. A key focus is to learn, identify and eliminate noise, taking into account the dynamic interest of a user and the evolving web data. Eliminating noise in extracted web log data is determined based on what a user is interested and not interested in. It is widely discussed in current research work, that the interest of a user on a web page is measured by how often they visit that page, how long they spend on the page, how recently they visited the page and the number of links on the page that they visit. To some extent, current research works measure user interest in extracted web data logs but there is inadequate evidence to demonstrate how noise in a web user profile is determined prior to elimination.

WEB USER PROFILE

A user profile has a set of URLs that represent a user interest. Creating a user profile is based on a set web page accessed by a user taking into account relevance of his/her interest. After creating a user profile, this work learns user interest levels on visited web pages so as to determine useful information from noise data. Various measures are considered, i.e. time, frequency and depth of visit of user visit to a webpage.

LEARNING USER INTEREST

The current research work recognises that it is important to learn user interest level to find useful information. This can be done by collecting user log data, analyzing it and storing the results in a user profile. User interest relies on the basis that the visiting time of a web page is an indicator of a user’s interest level . The amount of time spent in a set of web pages requested by the user within a single session reflects the interest of that user. In addition, states that web pages with higher frequency are of stronger interest to a user. Even though this paper considers page visit duration and frequency of visit to learn user interest on visited. Web pages, it is difficult to measure user interest levels based on page visit duration and frequency of visit alone. For example, high frequency of visit to a web page may either reflect a user struggling to find useful information or based on website layout, he/she is forced to visit some pages before accessing interested ones. Therefore, the proposed work considers additional measures such as depth of visit and frequency of visit to a web page category to learn user interest prior to elimination of noise data.

PAGE VISIT DURATION

Page visit duration is one of the metrics widely used by current research work to measure user interest level on a webpage. Generally, a user spends more time on a more useful page, if a user is not interested in a page, he/she will exit or move to another page duration defines the length of user interest on a web page. argue that calculation of page visit duration is a bit skewed because it is not possible to determine the time a user exits a web page as it is always 0.

DEPTH OF USER VISIT

Page visit depth is defined as the average number of pages viewed by visitors during a single browser session. The depth of the user visit on page is an indicator of a user interest level. The proposed tool considers the depth of the user visit not only in terms of a number of page views but the route a user takes to navigate through a website. The user creates a path of page views when searching for information on a specific website.

For example, a user may enter a website from home page but only interested in finding delivery charges for a specific item under accessories. Even though the user is likely to visit other web pages to get to the information of interest, it is difficult to assume that every page visit is of a user interest unless measures such as time duration and frequency to visit over a number of sessions are considered.

WEB PAGE CATEGORY WEIGHT

In this work, web page category is defined as a set of related web pages. The weight of a web page category is determined based on the frequency of user visits to a particular web page category. The more frequent a user visits the same category the higher the level of interest. Unlike frequency of visit to web page discussed in the frequency of visit toa web page category determines if a user is interested in information from a given category of web data. For example, ahigh number of visits to footwear web pages under men category depict interest on information regarding men shoes. Based on this concept, the weight of a web page category is presented for the purposes of learning user interest level to a particular web page category.

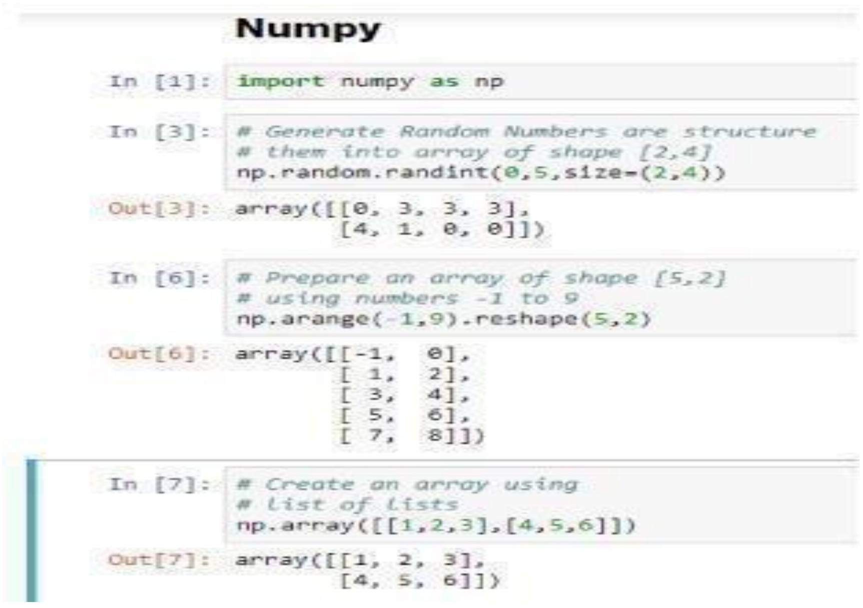
LEARNING NOISE WEB DATA

In this paper, learning of noise data in the user profile involves classification of the weighted web page presented in. Web page classification is the process of assigning a label to a web page. A class isa representation of a data object while an object is an instance of a class. For example, web page in the user profile is assigned to a class based on the level of interest. is a set of predefined classes. For illustrative purposes, let us consider the following classes as Interest class, as Potential noise class and as Noise class.

4.2 SYSTEM COMPONENTS (MODULES)

NUMPY

Python has a strong set of data types and data structures. Yet it wasn't designed for Machine Learning per say. Enter numpy (pronounced as num-pee). Numpy is a data handling library, particularly one which allows us to handle large multi- dimensional arrays along with a huge collection of mathematical operations. The following is a quick snippet of numpy in action.



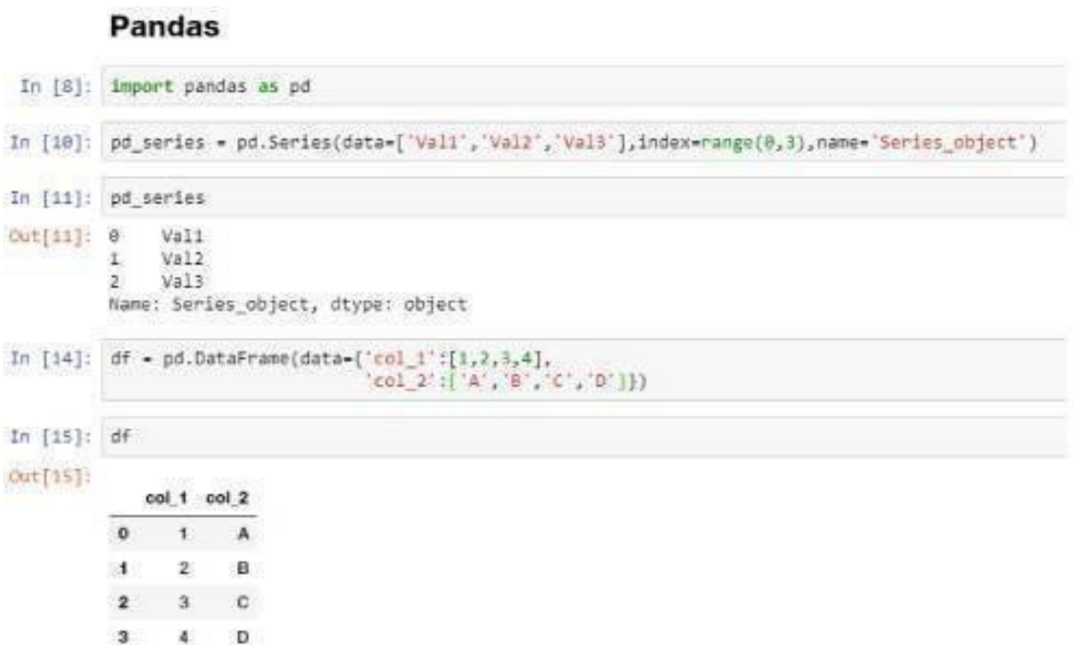
Numpy

```
In [1]:  import numpy as np

In [3]:  # Generate Random Numbers are structure
         # them into array of shape [2,4]
         np.random.randint(0,5,size=(2,4))

Out[3]:  array([[0, 3, 3, 3],
                [4, 1, 0, 0]])

In [6]:  # Prepare an array of shape [5,2]
         # using numbers -1 to 9
         np.arange(-1,9).reshape(5,2)

Out[6]:  array([[-1,  0],
                [ 1,  2],
                [ 3,  4],
                [ 5,  6],
                [ 7,  8]])

In [7]:  # Create an array using
         # list of lists
         np.array([[1,2,3],[4,5,6]])

Out[7]:  array([[1, 2, 3],
                [4, 5, 6]])
```

Fig 4.2.1 Numpy

PANDAS

Yes, Pandas is a python library that provides flexible and expressive data structures (like data frames and series) for data manipulation. Built on top of numpy, pandas is as fast and yet easier to use.



Pandas

```
In [8]:  import pandas as pd

In [10]: pd_series = pd.Series(data=['Val1','Val2','Val3'],index=range(0,3),name='Series_object')

In [11]: pd_series

Out[11]: 0    Val1
         1    Val2
         2    Val3
         Name: Series_object, dtype: object

In [14]: df = pd.DataFrame(data={'col_1':[1,2,3,4],
                                 'col_2':['A','B','C','D']})

In [15]: df

Out[15]:
```

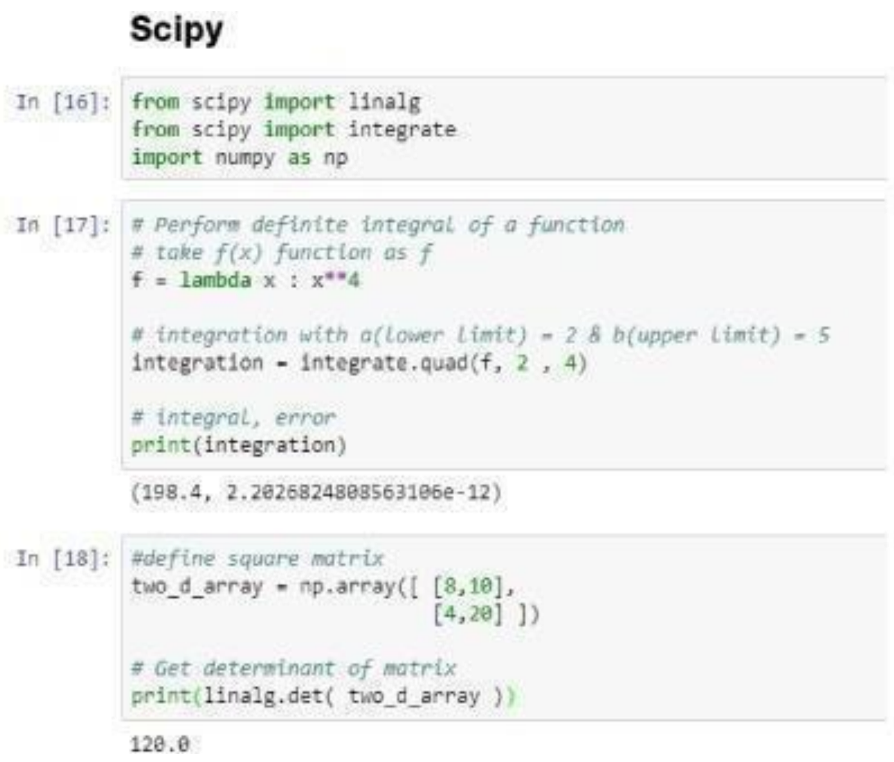
	col_1	col_2
0	1	A
1	2	B
2	3	C
3	4	D

Fig 4.2.2 Pandas

Pandas provides capabilities to read and write data from different sources like CSVs, Excel, SQL Databases, HDFS and many more. It provides functionality to add, update and delete columns, combine or split data frames/series, handle datetime objects, impute null/missing values, handle time series data, conversion numpy to and from numpy objects and soon. If you are working on a real-world Machine Learning use case, chances are, you would need pandas sooner than later. Similar to, pandas is also an important component of the Scipy or Scientific Python Stack.

SCIPY

Pronounced as Sigh-Pie, this is one of the most important python libraries of all time. Scipy is a scientific computing library for python. It is also built on top of numpy and is a part of the Scipy Stack.



```
Scipy

In [16]: from scipy import linalg
         from scipy import integrate
         import numpy as np

In [17]: # Perform definite integral of a function
         # take f(x) function as f
         f = lambda x : x**4

         # integration with a(lower limit) = 2 & b(upper limit) = 5
         integration = integrate.quad(f, 2 , 4)

         # integral, error
         print(integration)

         (198.4, 2.2026824808563106e-12)

In [18]: #define square matrix
         two_d_array = np.array([ [8,10],
                                  [4,20] ])

         # Get determinant of matrix
         print(linalg.det( two_d_array ))

         120.0
```

Fig 4.2.3 Scipy

MATPLOTLIB

Another component of the SciPy stack, matplotlib is essentially a visualization library. It works seamlessly with numpy objects (and its high- level derivatives like pandas). Matplotlib provides a MATLAB like plotting environment to prepare high- quality figures/charts for publications, notebooks, web applications and so on.

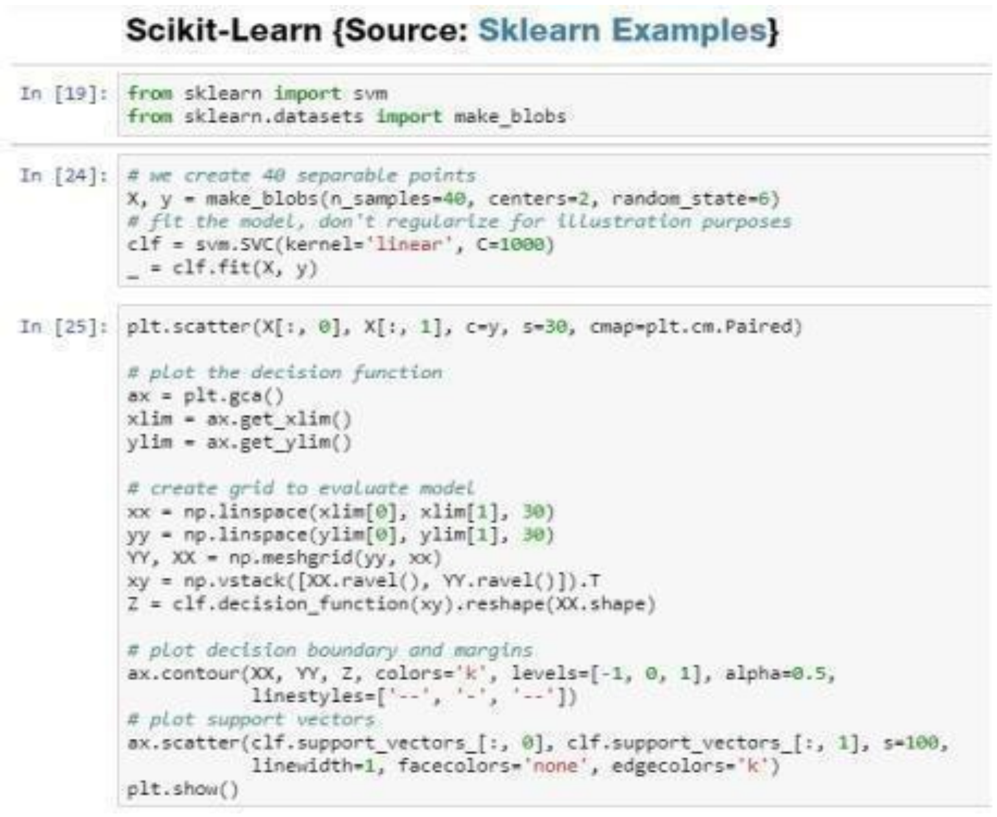
Fig 4.2.4 Matplotlib

Matplotlib is a high customizable low-level library that provides a whole lot of controls and knobs to prepare any type of visualization/figure. Given its low-level nature, it requires a bit of getting used to along with plenty of code to get stuff done.

SCIKIT-LEARN

Designed as an extension to the SciPy library, scikit- learn has become the de-facto standard for many of the machine learning tasks. Developed as part of Google Summer of Code project, it has now become a widely contributed open-source project with over 1000 contributors.

Scikit-learn provides a simple yet powerful fit- transform and predict paradigm to learn from data, transform the data and finally predict. Using this interface, it provides capabilities to prepare classification, regression, clustering and ensemble models.



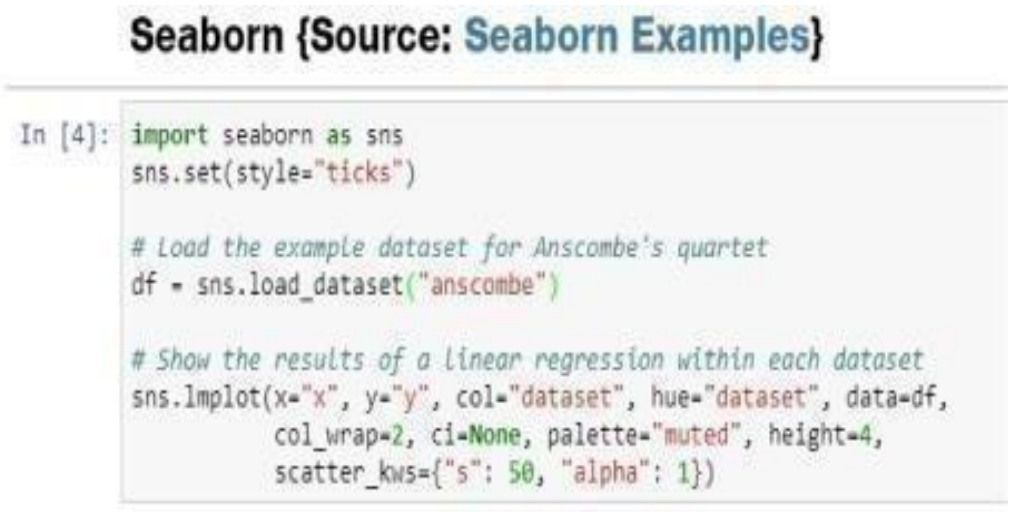
Scikit-Learn {Source: Sklearn Examples}

```
In [19]: from sklearn import svm
         from sklearn.datasets import make_blobs

In [24]: # we create 40 separable points
         X, y = make_blobs(n_samples=40, centers=2, random_state=6)
         # fit the model, don't regularize for illustration purposes
         clf = svm.SVC(kernel='linear', C=1000)
         _ = clf.fit(X, y)

In [25]: plt.scatter(X[:, 0], X[:, 1], c=y, s=30, cmap=plt.cm.Paired)

         # plot the decision function
         ax = plt.gca()
         xlim = ax.get_xlim()
         ylim = ax.get_ylim()

         # create grid to evaluate model
         xx = np.linspace(xlim[0], xlim[1], 30)
         yy = np.linspace(ylim[0], ylim[1], 30)
         YY, XX = np.meshgrid(yy, xx)
         xy = np.vstack([XX.ravel(), YY.ravel()]).T
         Z = clf.decision_function(xy).reshape(XX.shape)

         # plot decision boundary and margins
         ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
                    linestyles=['--', '-', '--'])
         # plot support vectors
         ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=100,
                    linewidth=1, facecolors='none', edgecolors='k')
         plt.show()
```

Fig 4.2.5 Scikit-learn

SEABORN

Built on top of matplotlib, seaborn is a high-level visualization library. It provides sophisticated styles straight out of the box(which would take some good amount of effort if done using matplotlib).



Seaborn {Source: Seaborn Examples}

```
In [4]: import seaborn as sns
        sns.set(style="ticks")

        # Load the example dataset for Anscombe's quartet
        df = sns.load_dataset("anscombe")

        # Show the results of a linear regression within each dataset
        sns.lmplot(x="x", y="y", col="dataset", hue="dataset", data=df,
                   col_wrap=2, ci=None, palette="muted", height=4,
                   scatter_kws={"s": 50, "alpha": 1})
```

Fig 4.2.6 Seaborn example

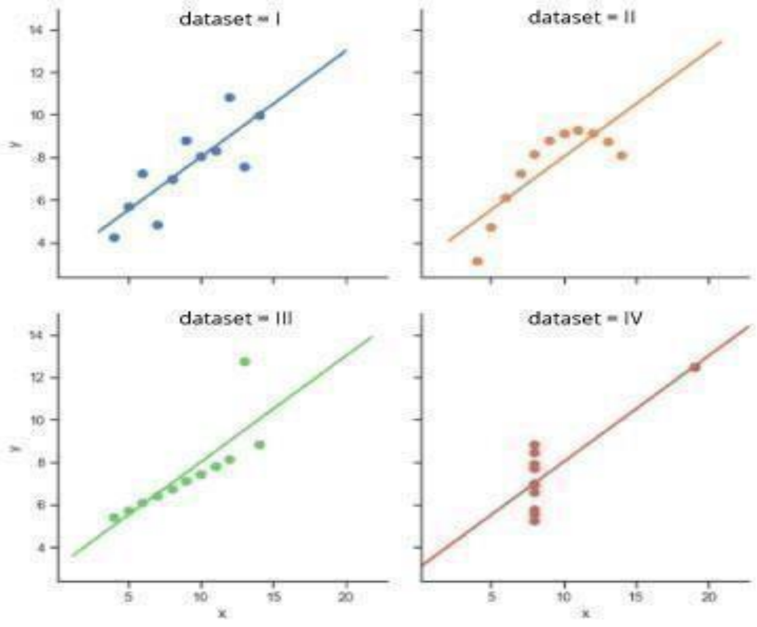


Fig 4.2.7 Seaborn Datasets example

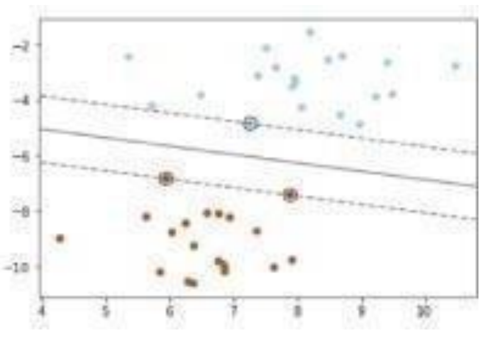


Fig 4.3.8 Seaborn

4.3 DFD DIAGRAM

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as data flow graph or a bubble chart.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create a DFD's are as follows:

Dataflow: Data move in a specific direction from an origin to a destination.

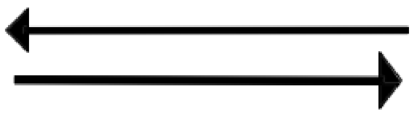


Fig 4.3.1 Data Flow

Process: People, procedures, or devices that use or produce (Transform)Data. The physical component is not identified.

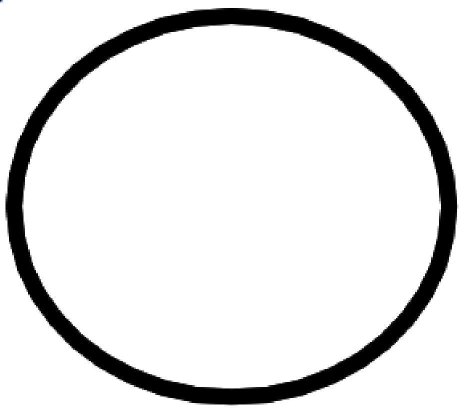


Fig 4.3.2 Process

Source: External sources or destination of data, which may be People, programs, organizations or other entities.

Fig 4.3.3 Source

Data Store: Here data are stored or referenced by a process in the System.

Fig 4.3.4 Data Source

4.4 UML DIAGRAM

UML represents Unified Modelling Language. UML is an institutionalized broadly useful displaying language in the field of article situated programming building. The standard is overseen, and was made by, the Object Management Group.

The objective is for UML to turn into a typical language for making models of article arranged PC programming. In its present structure UML is involved two significant segments: a Meta-model and a documentation. Later on, some type of strategy or procedure may likewise be added to; or related with, UML.

The Unified Modelling Language is a standard language for determining, Visualization, Constructing and reporting the ancient rarities of programming framework, just as for business demonstrating and other non-programming frameworks.

The UML speaks to an assortment of best designing practices that have demonstrated fruitful in the displaying of huge and complex frameworks. The UML is a significant piece of creating objects arranged programming and the product advancement process. The UML utilizes for the most part graphical documentations to communicate the plan of programming ventures.

USE CASE DIAGRAMS

An utilization case chart in the Unified Modelling Language (UML) is a kind of conduct graph characterized by and made from a Use-case examination. Its motivation is to introduce a graphical outline of the usefulness gave by a framework as far as on-screen characters, their objectives (spoke to as use cases), and any conditions between those utilization cases.

The fundamental motivation behind an utilization case chart is to show what framework in the framework can be delineated capacities are performed for which on-screen character. Jobs of the on-screen characters.

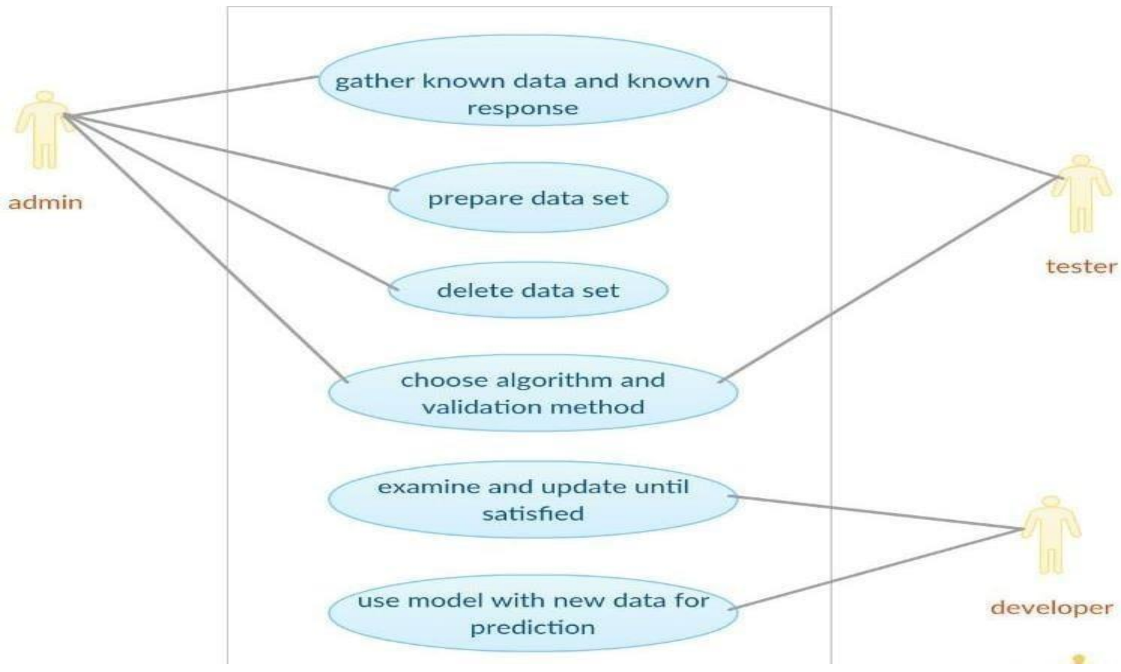

admin
gather known data and known response
prepare data set
delete data set
choose algorithm and validation method
examine and update until satisfied
use model with new data for prediction
tester
developer

Fig 4.4.1 Use case Diagram

CLASS DIAGRAM

In programming building, a class graph in the Unified Modelling Language (UML) is a kind of static structure outline that portrays the structure of a framework by indicating the framework's classes, their properties, tasks (or techniques), and the connections among the classes. It clarifies which class contains data.

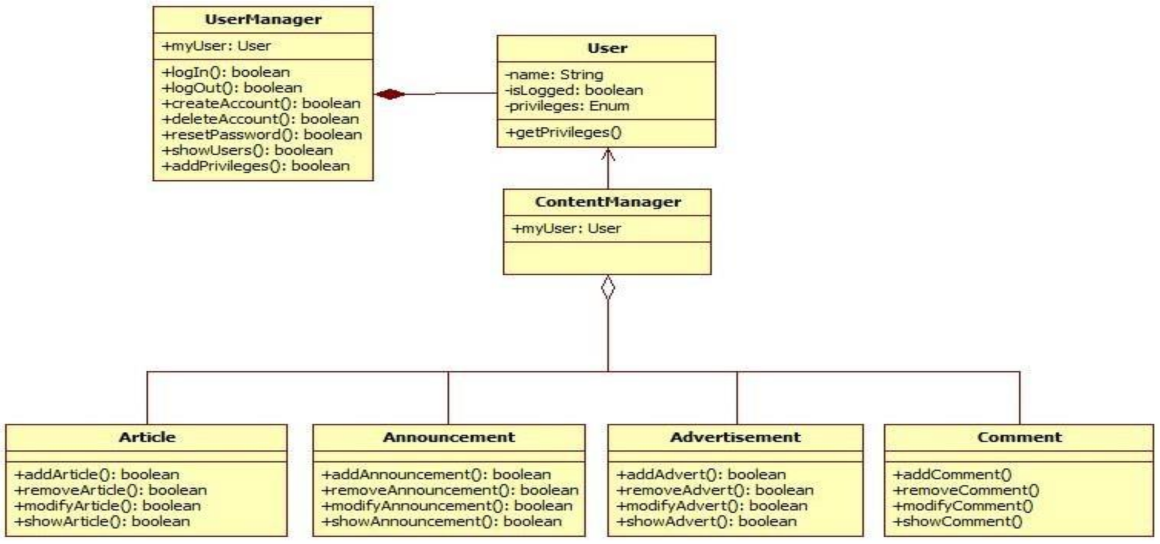


Fig 4.4.2 Class Diagram

SEQUENCE DIAGRAM

An arrangement outline in Unified Modelling Language (UML) is a sort of connection graph that shows how procedures work with each other and in what request. It is a build of a Message Sequence Chart. Arrangement outlines are now and then called occasion graphs, occasion situations, and timing charts.

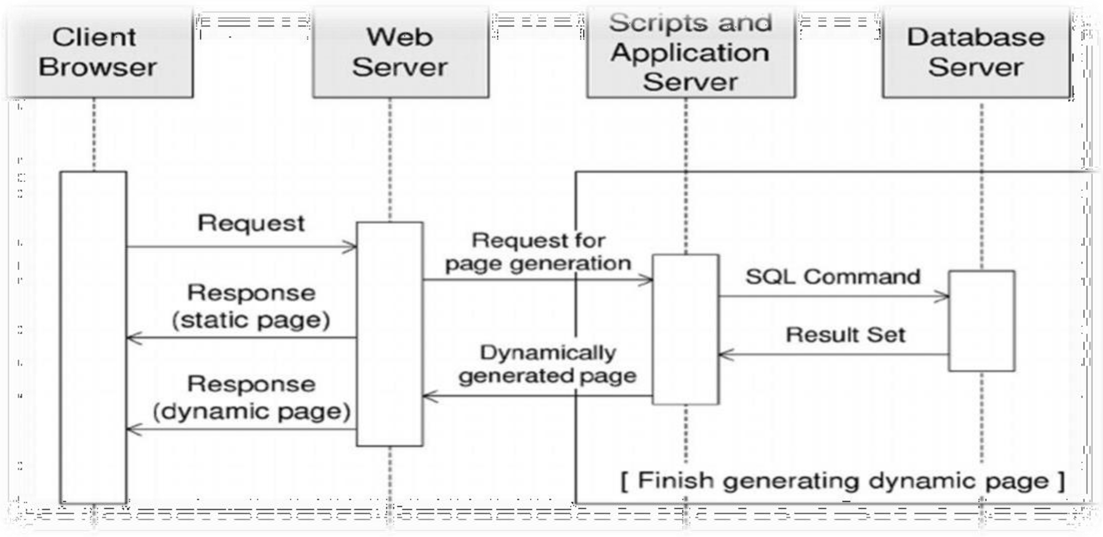
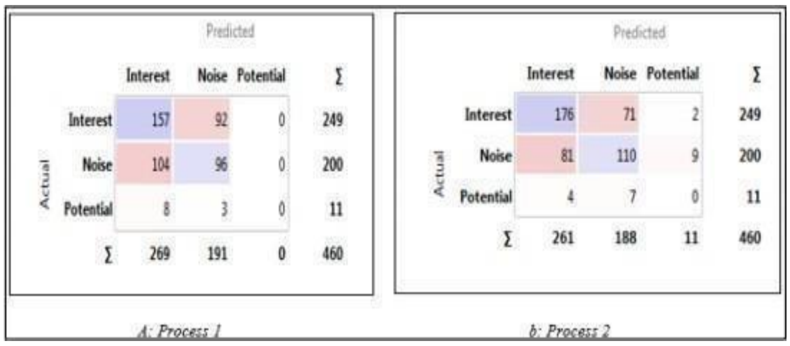

Client Browser
Web Server
Scripts and Application Server
Database Server
Request
Response (static page)
Response (dynamic page)
Request for page generation
Dynamically generated page
SQL Command
Result Set
[Finish generating dynamic page]

Fig 4.4.3 Sequence Diagram

DATA DICTIONARY

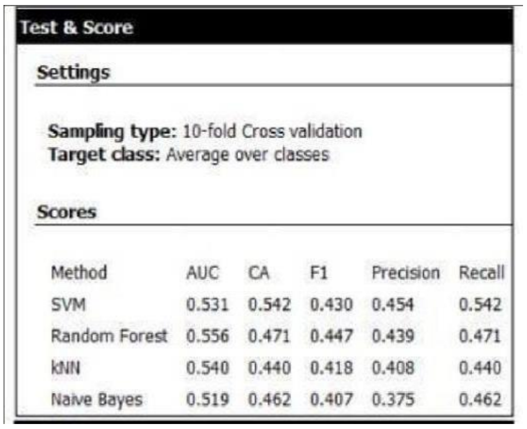


		Predicted			
		Interest	Noise	Potential	∑
Actual	Interest	157	92	0	249
	Noise	104	96	0	200
	Potential	8	3	0	11
	∑	269	191	0	460

A: Process 1

		Predicted			
		Interest	Noise	Potential	∑
Actual	Interest	176	71	2	249
	Noise	81	110	9	200
	Potential	4	7	0	11
	∑	261	188	11	460

b: Process 2



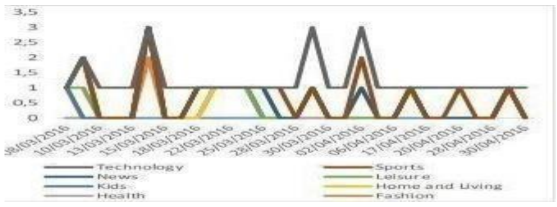
Test & Score

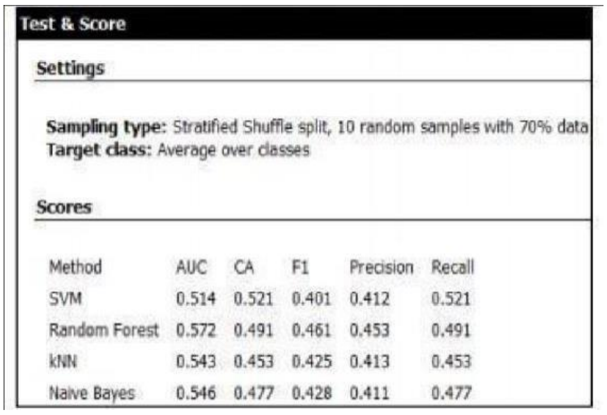
Settings

Sampling type: 10-fold Cross validation
Target class: Average over classes

Scores

Method	AUC	CA	F1	Precision	Recall
SVM	0.531	0.542	0.430	0.454	0.542
Random Forest	0.556	0.471	0.447	0.439	0.471
kNN	0.540	0.440	0.418	0.408	0.440
Naive Bayes	0.519	0.462	0.407	0.375	0.462





Test & Score

Settings

Sampling type: Stratified Shuffle split, 10 random samples with 70% data
Target class: Average over classes

Scores

Method	AUC	CA	F1	Precision	Recall
SVM	0.514	0.521	0.401	0.412	0.521
Random Forest	0.572	0.491	0.461	0.453	0.491
kNN	0.543	0.453	0.425	0.413	0.453
Naive Bayes	0.546	0.477	0.428	0.411	0.477

Fig 4.4.4 Data Dictionary

ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by- step workflows of components in a system. An activity diagram shows the overall flow of control.

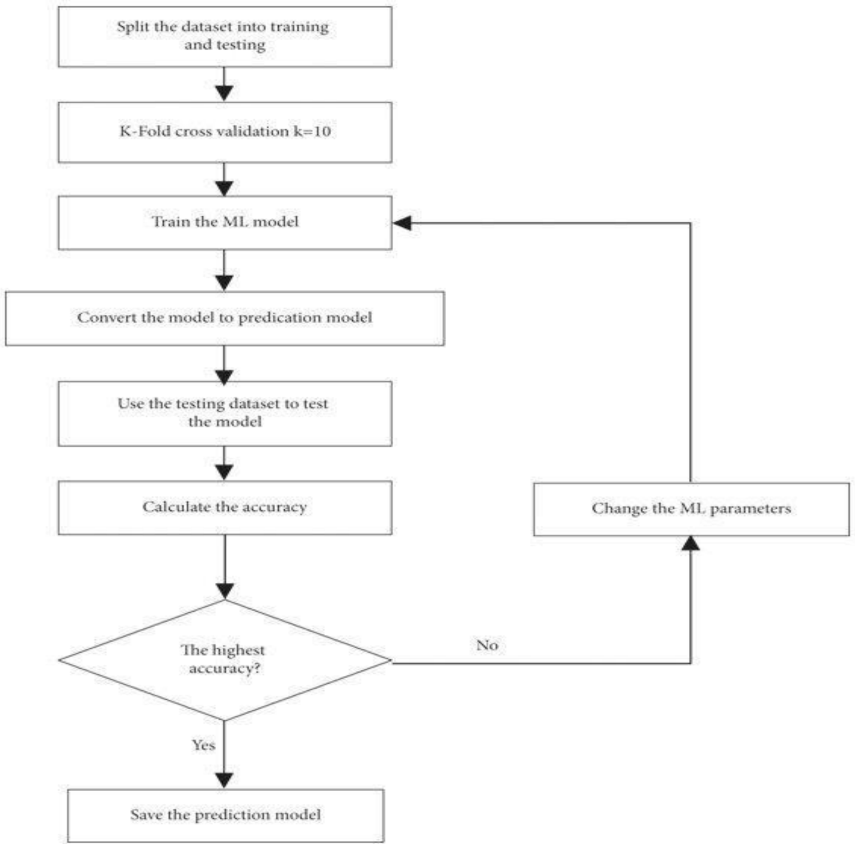

Split the dataset into training and testing
K-Fold cross validation k=10
Train the ML model
Convert the model to predication model
Use the testing dataset to test the model
Calculate the accuracy
Change the ML parameters
The highest accuracy?
No
Yes
Save the prediction model

Fig 4.4.6 Activity Diagram

5. METHODOLOGY

Various machine learning tools/algorithms are used to discover useful information from web data, this process is referred to as web usage/data mining process. It finds user interest patterns from web log data. Web log data contains a list of actions that have occurred on the web based on a user. These log files give an idea about what a user is interested in available web data. Web log data contain basic information such as IP address, user visit duration and visiting path, web page visited by the user, time spent on each web page visit etc. In this work, web log file and web data are used interchangeably because a log file contains web data, therefore elimination of noise web data is based on extracted web user log file.

Current research efforts in noise web data reduction have worked with the assumption that the web data is static. For example, proposed a mechanism where noise detected from web pages is matched by stored noise data for classification and subsequent elimination. Therefore, it shows that elimination of noise in web data is based on pre-existing noise data patterns. In evolving web data, existing noise data patterns used to identify and eliminate noise from web data may become out of date. For this reason, the dynamic aspects of user interest have recently become important. Moreover, web access patterns are dynamic not only due to evolving web data but also due to changes in user interests. For example, web users are likely to be interested in data derived from events such as Weddings, Christmas, Birthdays etc. Therefore, it is necessary to discover where such dynamic tendencies impact the process of eliminating noise from web data.

Now-a-days almost all users are using web pages to get various information such as news, sports, technology etc but all web pages will use noise data such as images, video clips or advertisement which makes difficult for the users to get interested information. To remove noise data all existing technologies were using static web matching pattern such as the main page look and feel will be match with rest of the screen and if not match then it will remove unmatched data from the web pages to show only interested data to the user. This static technique will not work if web pages look and feel changes dynamically.

To overcome from above issue author is proposing Noise Web Data Learning (NWDL) technique, in this technique server will maintain log for each user access page and will be called as web log dataset. This dataset will have information such as User_id, access_page, date time, URL. By analysing such log data we can identify user interested pages in dynamic or static web pages. User interested pages can be found by seeing frequency of web page access by a single user and total time spend on each page.

If user spend more time and access this page more than 2 times then we can consider that user is interested in that page. If user spend less time on seeing that page and visiting that page very rarely then it will consider as uninterested page and will be called as noise page.

Dataset Example:

User_id	**access_page**	**date_time**
1	abcd.html	2019-01-22 11:00:12
2	xyz.html	2019-01-22 12:18:23
1	abcd.html	2019-01-22 11:05:18
1	abcd.html	2019-01-22 11:06:12
1	xyz.html	2019-01-22 12:22:23

Table 5(a) Dataset Example

From above web log dataset, we can easily say that user 1 accessing abcd.html more no of time and its frequency is 3 and he spend almost 6 minutes on that page (spend time will be calculated from first same page visit to till last same page visit) and this abcd.html will be consider as interested page from user 1 and xyz.html is rarely access by that user and will be consider as noise page and will not recommend to user.

To perform experiment author has used WEBLOG dataset and I am also using same.

5.1 ALGORITHM

```
Algorithm: Learn noise web data
Input: Weighted url(k) for the jth user profile
Output: A set of web pages assigned to a class (cln)
1. Define the j(th) user profile
2. for each k(th) web page in j(th) user profile do
3. Determine the weight of k(th) web page using (3)
```

```
4. if url(k) weight > threshold set then
5. assign to cl1
6. else
7. assign to cl2
8. for cl2 do
9. Create a simple page link of the j(th) user profile
10. Determine frequency to web page category using
11. if Freq(m) <threshold set then
12. assign to cl3
13. else
14. update $cl_1$
15. end if
16. end if
17. end
```

6. IMPLEMENTATION

6.1 PYTHON

Python was created in early 1990s by Guido van Rossum at Stich ting Mathematisch Centrum in the Netherlands as a successor of a language called ABC. Guido remains python's principle author, although it includes many contributions from others. Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use python and see almost immediate gains in productivity and lower maintenance costs.

About python:

Python is a remarkably powerful dynamic programming language that is used in a wide variety of application domains. Some of its key distinguish features include:

• Very clear, readable syntax

• Strong introspection capabilities

• Intuitive object orientation

• Natural expression of procedural code

• Full modularity, supporting hierarchical packages

• Exception-based error handling

• Very high-level dynamic data types

• Extensive standard libraries and third-party modules for virtually every task

• Extensions and modules easily written in C, C++(or java for python, or .NET languages for python)

• Embeddable within applications as a scripting interface

Python is Interpreted:

Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive:

You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented:

Python supports object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner’s Language:

Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python runs everywhere:

Python is available for all major operating systems: Windows, Linux/Unix, OS2, Mac,Amiga, among others. There are even versions that run on .NET and the java virtual machine. You’ll be pleased to know that the same source code will run unchanged across all implementations.

Python is friendly and easy to learn:

The python newsgroups known as one of the friendliest around. Python also comes with complete documentation, both integrated into the language and as separate webpages**.**

Python is Open:

The python implementation is under an open source licence that makes it freely usable and distributable, even for commercial use. The python licence is administered by the Python Software Foundation**.**

6.2 DATASET DESCRIPTION

This dataset contains records of user interactions with web servers, specifically accessing pages and resources. Each record includes the following fields:

server_ip: The IP address or hostname of the server accessed.

User_id: An anonymized user identifier.

Access_page: The URL or path of the accessed page or resource.

Date_Time: The date and time of the access event.

URL: The full URL of the accessed page or resource.

This dataset serves various analytical purposes, including understanding user engagement patterns, identifying popular pages or resources, detecting abnormal or suspicious activity, and improving website navigation and user experience. Analysts can leverage this data to derive actionable insights for optimizing website content, enhancing user satisfaction, and mitigating security risks. It's essential to note that user IDs are anonymized to protect user privacy, and some access_page entries may represent non-page resources such as images or audio files.

6.3 MACHINE LEARNING

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models unable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data.**o**nce these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model- based "learning"is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges.The challenges that ML is facing currently are –

• Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

• Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

• Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

• No clear objective for formulating business problems − Having no clear objective and well- defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

• Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.
• Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.
• Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python.

a) Learn Linear Algebra and Multi variate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection,analysis, and presentation of data. So it is no surprise that you need to learn it!!! Some of the keyconcepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc

c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine

Learning such as Keras, TensorFlow,Scikit-learn, etc.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML It’s best to start with the basics and then move on to the more complicated stuff.Some of the basic concepts in ML are:

a) Terminologies of Machine Learning

• Model – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.

• Feature – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector.

• Training – The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

• Prediction – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

b) Types of Machine Learning

• **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

• **Unsupervised Learning –** This involves using un labelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.

• **Semi-supervised Learning –** This involves using un labelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.

• **Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize there ward in the future.

6.4 NWDL (Noise Web Data Learning)

NWDL, short for Noise Web Data Learning, is a specialized framework or methodology developed for analysing web data with the primary goal of distinguishing between signal (meaningful user interactions) and noise (unwanted or irrelevant interactions). Unlike traditional machine learning approaches that may focus solely on pattern recognition, NWDL appears tailored specifically for addressing noise reduction challenges in web server access logs.

NWDL serves as the backbone for several critical tasks:

Noise Reduction: NWDL provides algorithms and techniques for filtering out noise from web server access logs. This includes identifying and excluding non-human traffic, spammy interactions, or other low-quality data that does not represent genuine user engagement.

Pattern Recognition: NWDL likely includes advanced pattern recognition algorithms to identify meaningful user interactions and behavioral patterns within web data. By distinguishing between typical user behavior and anomalous activities, NWDL helps in understanding user interests and preferences.

Learning from Data: NWDL emphasizes the learning aspect, implying that it continuously adapts and improves its noise reduction capabilities based on the data it encounters. This may involve iterative model training, feedback mechanisms, or reinforcement learning techniques to enhance the accuracy and effectiveness of noise reduction algorithms over time.

Performance Evaluation: NWDL facilitates the evaluation of noise reduction techniques through metrics such as confusion matrices. By quantifying the performance of classification models, NWDL enables developers to assess the efficacy of their noise reduction strategies and make informed decisions for refinement.

6.5 PROGRAM CODE

6.5.1 NWDL

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
import datetime
```

```
from UserProfile import *
import numpy as np
from collections import defaultdict
from tkinter.filedialog import askopenfilename
from tkinter import simpledialog
import webbrowser
global total_count
def upload():
    userprofile.clear()
    j = 0;
    global filename
    filename = askopenfilename(initialdir = "dataset")
    pathlabel.config(text=filename)
    with open(filename, "r") as file:
      for line in file:
       line = line.strip('\n')
       arr = line.split("\t")
       if j > 0:
         up = UserProfile()
         up.setServer(arr[0])
         up.setUser(arr[1])
         up.setWebpage(arr[2])
         up.setDate(datetime.datetime.strptime(arr[3], '%Y-%m-%d %H:%M:%S'))
         up.setURL(arr[4])
         userprofile.append(up);
        f.write(dataset)
    f.close()
    text.insert(END,"Total Frequent Users Size : "+str(total_count))
getPageDepth())+"\t\t"+up.getWebpage()+"\n");
          text.insert(END,"Complete Page URL : "+up.getURL()+"\n\n")
def confusionMatrix():
    interest = 0
    noise = 0
    potential = 0
   sinterest = 0
    snoise = 0
    spotential = 0
    for k, v in depth.items():
      for up in v:
        if up.getWeight() >= 10:
          interest = interest + 1
        if up.getWeight() < 10:
          noise = noise + 1
    dataset = pd.read_csv('dataset.csv')
    dataset = dataset.values
    X = dataset[:,0:dataset.shape[1]-1]
    Y = dataset[:,dataset.shape[1]-1]
    print(X)
    print(Y)
    X = normalize(X)
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
    cls = GaussianNB()
title = Label(main, text='Noise Reduction in Web Data: A Learning Approach Based on Dynamic User
Interests')
title.config(bg='brown', fg='white')
title.config(font=font)
title.config(height=3, width=80)
title.place(x=5,y=5)
font1 = ('times', 14, 'bold')
upload = Button(main, text="Upload Weblog Dataset", command=upload)
upload.place(x=50,y=100)
upload.config(font=font1)
pathlabel = Label(main)
pathlabel.config(bg='brown', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=300,y=100)
depthbutton = Button(main, text="Calculate Depth User Visit", command=findSession)
depthbutton.place(x=50,y=150)
depthbutton.config(font=font1)
userinterest = Button(main, text="View User Interest Pages", command=viewinterest)
userinterest.place(x=330,y=150)
userinterest.config(font=font1)
matrix = Button(main, text="View Confusion Matrix", command=confusionMatrix)
matrix.place(x=610,y=150)
matrix.config(font=font1)
graph = Button(main, text="Dynamic Interest Category Graph", command=graph)
graph.place(x=870,y=150)
graph.config(font=font1)
```

```
openpage = Button(main, text="Open Interested Page", command=openpage)
openpage.place(x=50,y=200)
openpage.config(font=font1)
font1 = ('times', 12, 'bold')
text=Text(main,height=25,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=250)
text.config(font=font1)
main.config(bg='brown')
main.mainloop()
```

6.5.2 USER PROFILE:

```
class UserProfile:
    def getServer(self):
        return self.server
    def setServer(self, server):
        self.server = server
    def getUser(self):
        return self.user
    def setUser(self, user):
        self.user = user
    def getWebpage(self):
        return self.webpage
    def setWebpage(self, webpage):
        self.webpage = webpage
    def getDate(self):
        return self.date
    def setDate(self, date):
        self.date = date
    def getURL(self):
        return self.url
    def setURL(self, url):
        self.url = url
    def setFrequency(self,frequency):
        self.frequency = frequency;
    def getFrequency(self):
        return self.frequency;
    def getWeight(self)
```

7. EXPERIMENTAL RESULTS

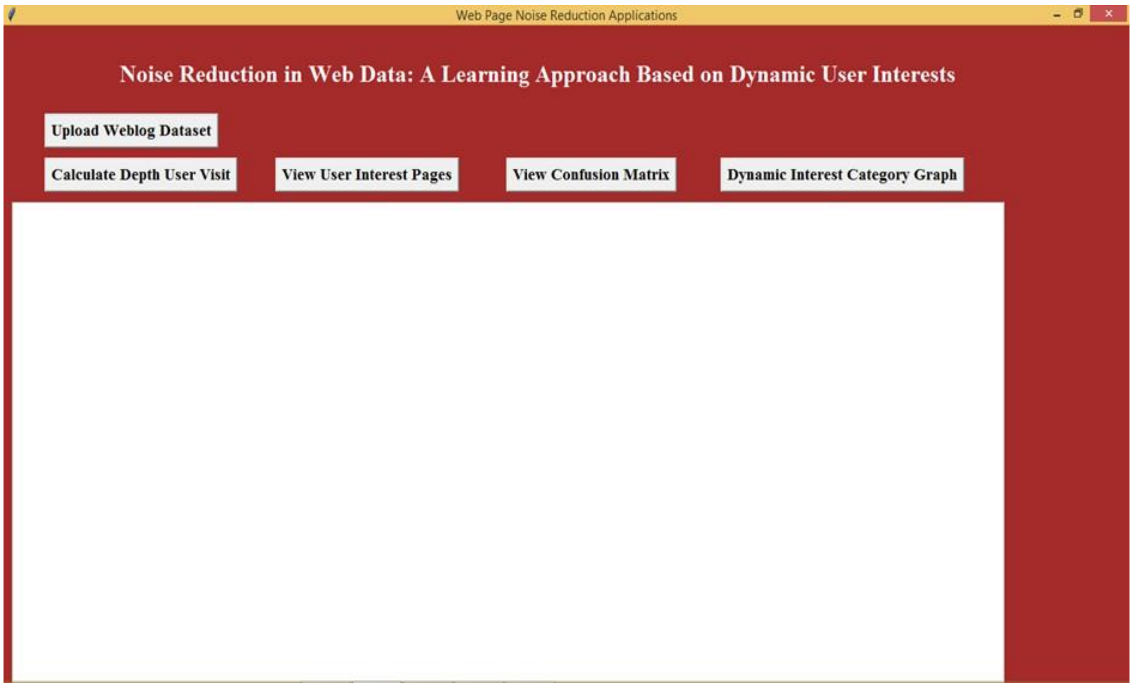
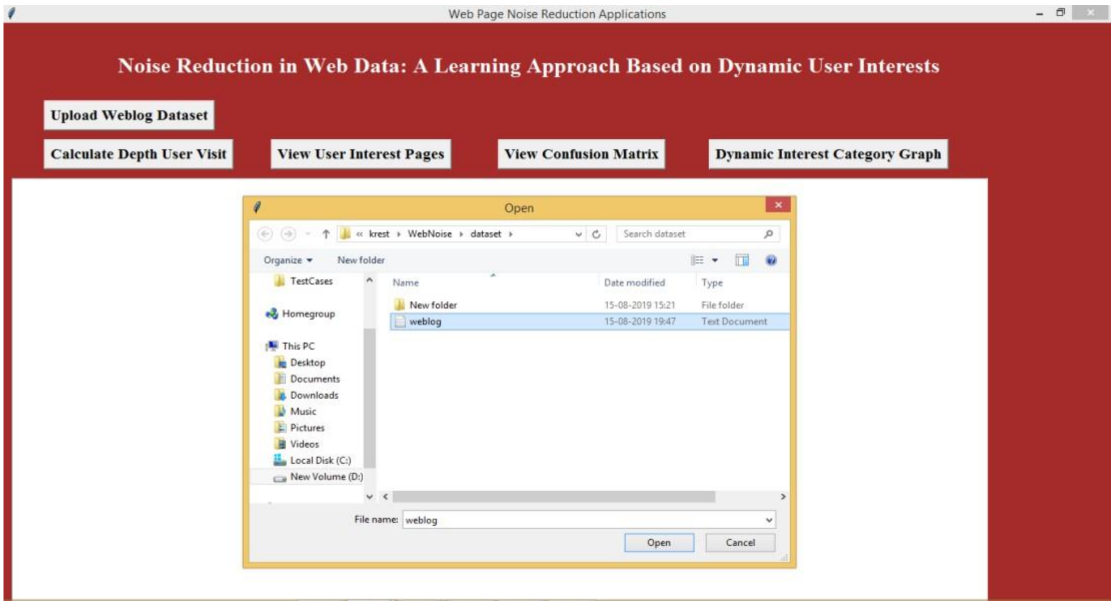

Web Page Noise Reduction Applications
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Upload Weblog Dataset
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph

Fig 7.1 Click on Upload web data

In above screen click on 'Upload Weblog Dataset' button to upload dataset



Name	Date modified	Type
New folder	15-08-2019 15:21	File folder
weblog	15-08-2019 19:47	Text Document

File name: weblog

Fig 7.2 Upload web data

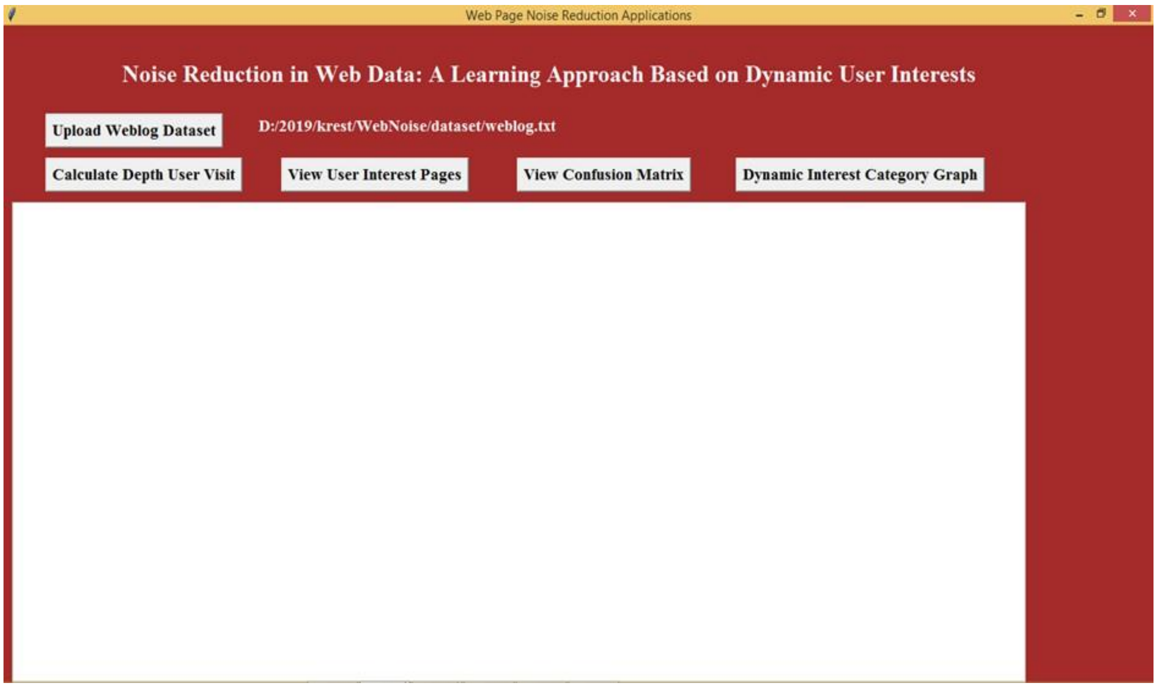
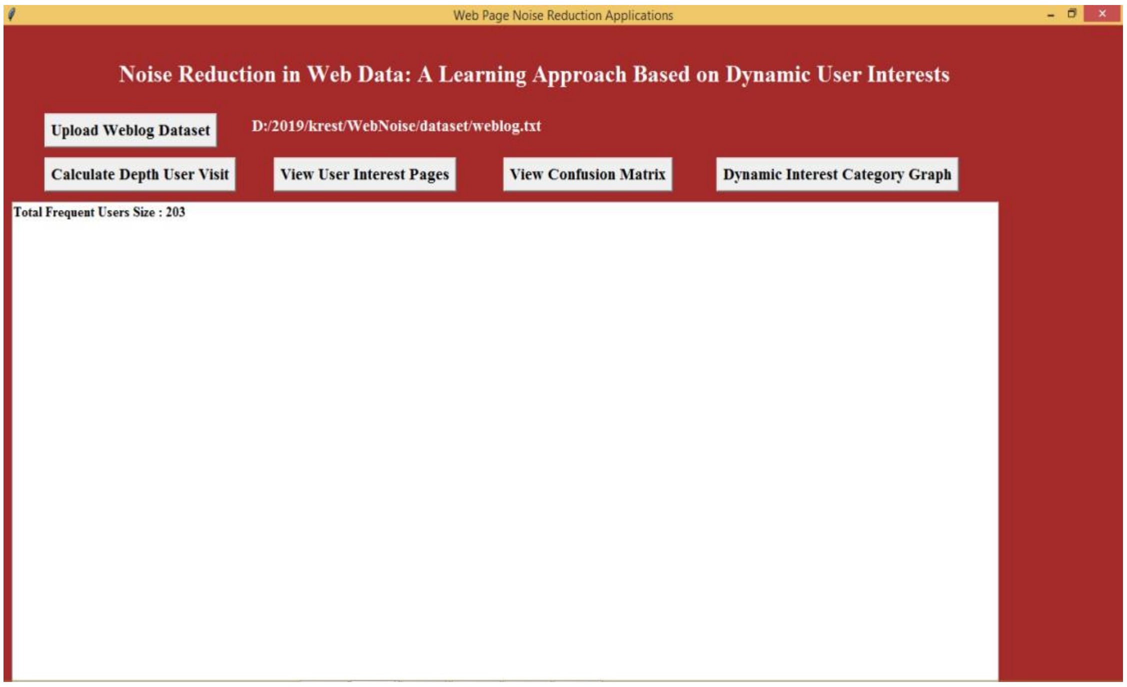

Web Page Noise Reduction Applications
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Upload Weblog Dataset
D:/2019/krest/WebNoise/dataset/weblog.txt
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph

Fig 7.3 Click on Calculate depth user

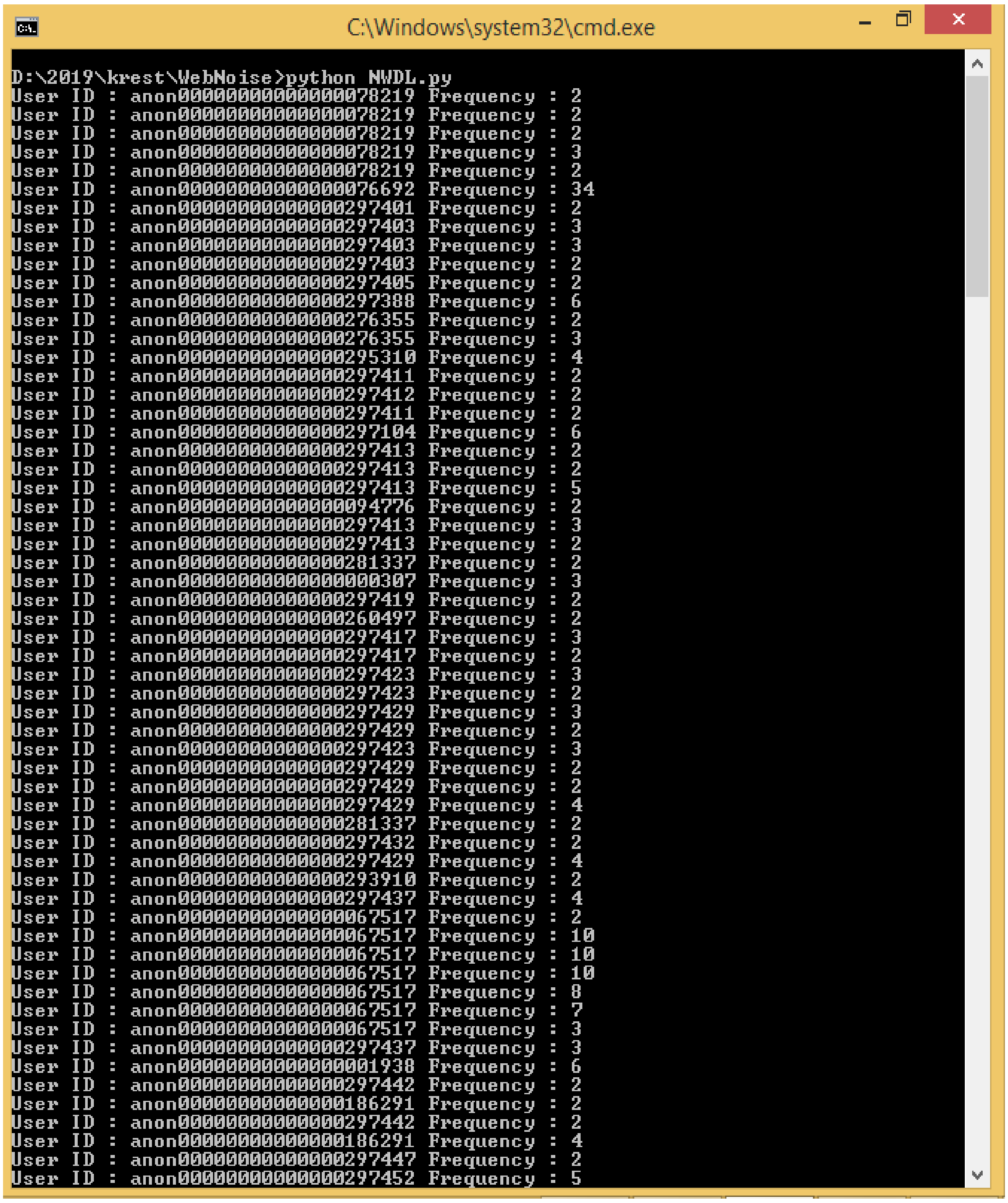
After dataset upload click on ‘Calculate Depth User Visit’ button to calculate frequency weight of each page visit by single users.


Web Page Noise Reduction Applications
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Upload Weblog Dataset
D:/2019/krest/WebNoise/dataset/weblog.txt
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph

```
Total Frequent Users Size : 203
```

Fig 7.4 Frequency User Size

In above screen we can see total 203 webpages which access more frequently. To see frequency of each access page see command prompt console. See below screen.

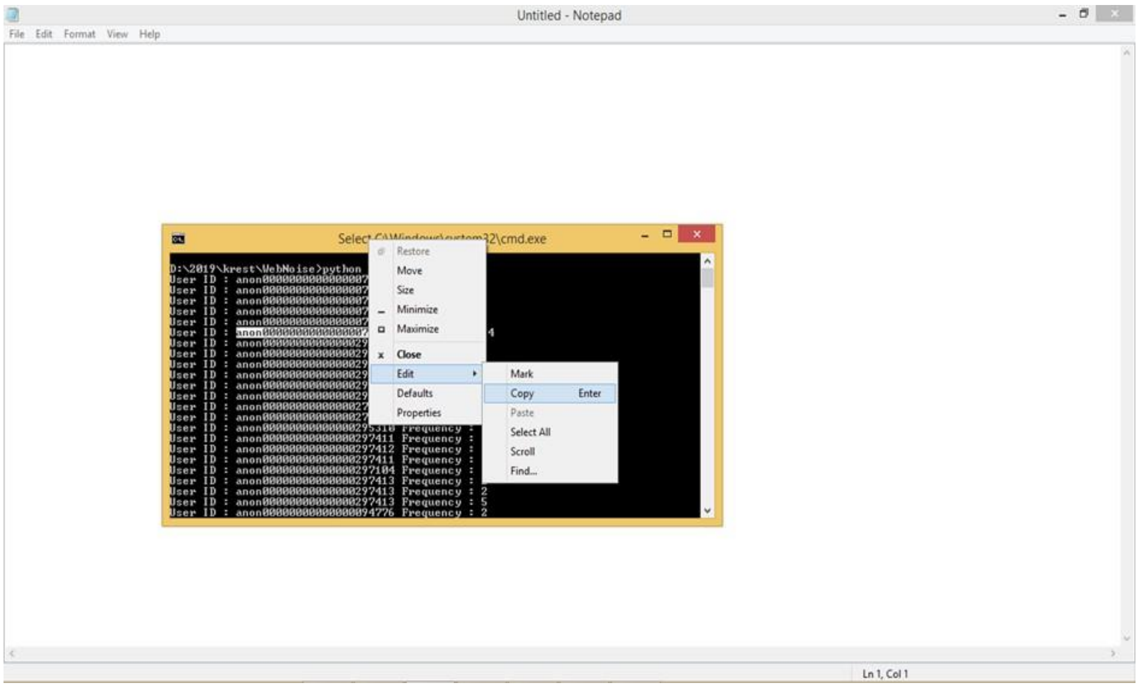


```
D:\2019\krest\WebNoise>python NWDL.py
User ID : anon00000000000000078219 Frequency : 2
User ID : anon00000000000000078219 Frequency : 2
User ID : anon00000000000000078219 Frequency : 2
User ID : anon00000000000000078219 Frequency : 3
User ID : anon00000000000000078219 Frequency : 2
User ID : anon00000000000000076692 Frequency : 34
User ID : anon00000000000000297401 Frequency : 2
User ID : anon00000000000000297403 Frequency : 3
User ID : anon00000000000000297403 Frequency : 3
User ID : anon00000000000000297403 Frequency : 2
User ID : anon00000000000000297405 Frequency : 2
User ID : anon00000000000000297388 Frequency : 6
User ID : anon00000000000000276355 Frequency : 2
User ID : anon00000000000000276355 Frequency : 3
User ID : anon00000000000000295310 Frequency : 4
User ID : anon00000000000000297411 Frequency : 2
User ID : anon00000000000000297412 Frequency : 2
User ID : anon00000000000000297411 Frequency : 2
User ID : anon00000000000000297104 Frequency : 6
User ID : anon00000000000000297413 Frequency : 2
User ID : anon00000000000000297413 Frequency : 2
User ID : anon00000000000000297413 Frequency : 5
User ID : anon00000000000000094776 Frequency : 2
User ID : anon00000000000000297413 Frequency : 3
User ID : anon00000000000000297413 Frequency : 2
User ID : anon00000000000000281337 Frequency : 2
User ID : anon00000000000000000307 Frequency : 3
User ID : anon00000000000000297419 Frequency : 2
User ID : anon00000000000000260497 Frequency : 2
User ID : anon00000000000000297417 Frequency : 3
User ID : anon00000000000000297417 Frequency : 2
User ID : anon00000000000000297423 Frequency : 3
User ID : anon00000000000000297423 Frequency : 2
User ID : anon00000000000000297429 Frequency : 3
User ID : anon00000000000000297429 Frequency : 2
User ID : anon00000000000000297423 Frequency : 3
User ID : anon00000000000000297429 Frequency : 2
User ID : anon00000000000000297429 Frequency : 2
User ID : anon00000000000000297429 Frequency : 4
User ID : anon00000000000000281337 Frequency : 2
User ID : anon00000000000000297432 Frequency : 2
User ID : anon00000000000000297429 Frequency : 4
User ID : anon00000000000000293910 Frequency : 2
User ID : anon00000000000000297437 Frequency : 4
User ID : anon00000000000000067517 Frequency : 2
User ID : anon00000000000000067517 Frequency : 10
User ID : anon00000000000000067517 Frequency : 10
User ID : anon00000000000000067517 Frequency : 10
User ID : anon00000000000000067517 Frequency : 8
User ID : anon00000000000000067517 Frequency : 7
User ID : anon00000000000000067517 Frequency : 3
User ID : anon00000000000000297437 Frequency : 3
User ID : anon00000000000000001938 Frequency : 6
User ID : anon00000000000000297442 Frequency : 2
User ID : anon00000000000000186291 Frequency : 2
User ID : anon00000000000000297442 Frequency : 2
User ID : anon00000000000000186291 Frequency : 4
User ID : anon00000000000000297447 Frequency : 2
User ID : anon00000000000000297452 Frequency : 5
```

Fig 7.5 Frequencies in web pages

In above screen we can see user id and frequency of each page access by them, if u wants to see page name and weight details then copy one user id from black console.

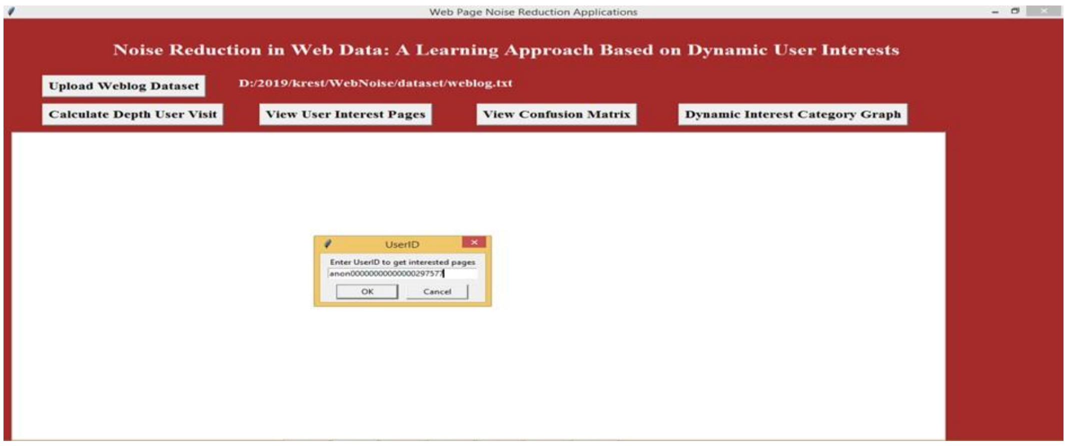
See below screen.



```
D:\2019\krest\WebNoise>python
User ID : anon0000000000000007
User ID : anon0000000000000007
User ID : anon0000000000000007
User ID : anon0000000000000007
User ID : anon0000000000000007
User ID : anon0000000000000007                4
User ID : anon0000000000000029
User ID : anon0000000000000029
User ID : anon0000000000000029
User ID : anon0000000000000029
User ID : anon0000000000000029
User ID : anon0000000000000029
User ID : anon0000000000000027
User ID : anon0000000000000027
User ID : anon0000000000000295310 Frequency :
User ID : anon0000000000000297411 Frequency :
User ID : anon0000000000000297412 Frequency :
User ID : anon0000000000000297411 Frequency :
User ID : anon0000000000000297104 Frequency :
User ID : anon0000000000000297413 Frequency :
User ID : anon0000000000000297413 Frequency : 2
User ID : anon0000000000000297413 Frequency : 5
User ID : anon0000000000000094776 Frequency : 2
```

Fig 7.6 Copy User ID

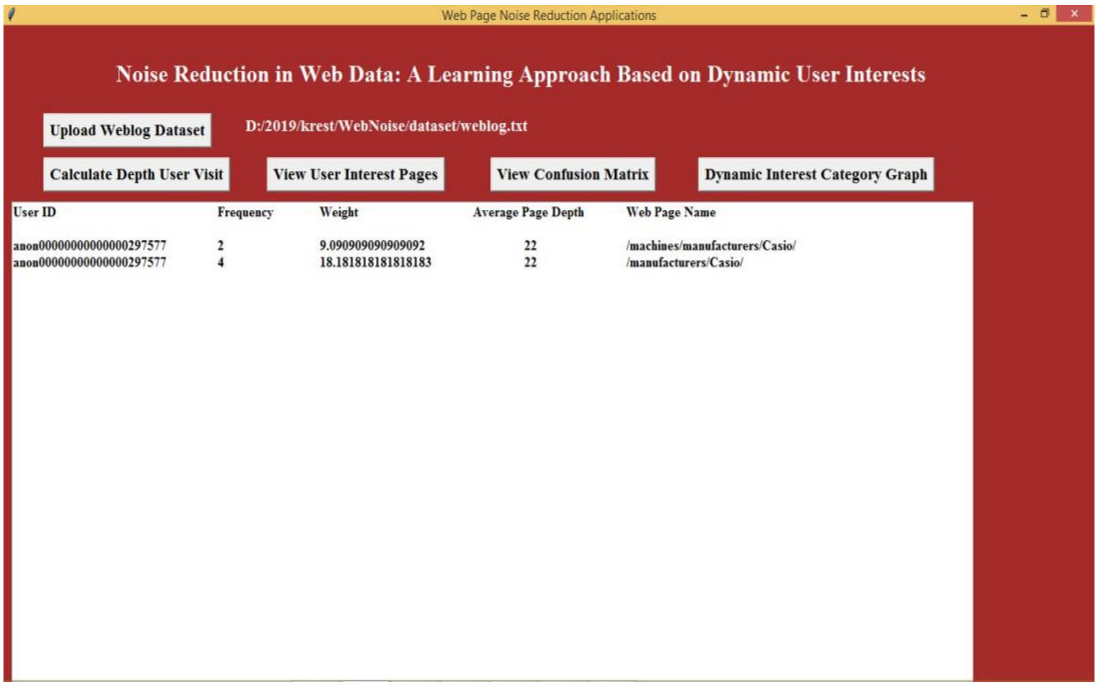
In above screen from command prompt I am copying one user id who access web pages frequently.


Web Page Noise Reduction Applications
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Upload Weblog Dataset
D:/2019/krest/WebNoise/dataset/weblog.txt
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph
UserID
Enter UserID to get interested pages
OK
Cancel

anon0000000000000297577

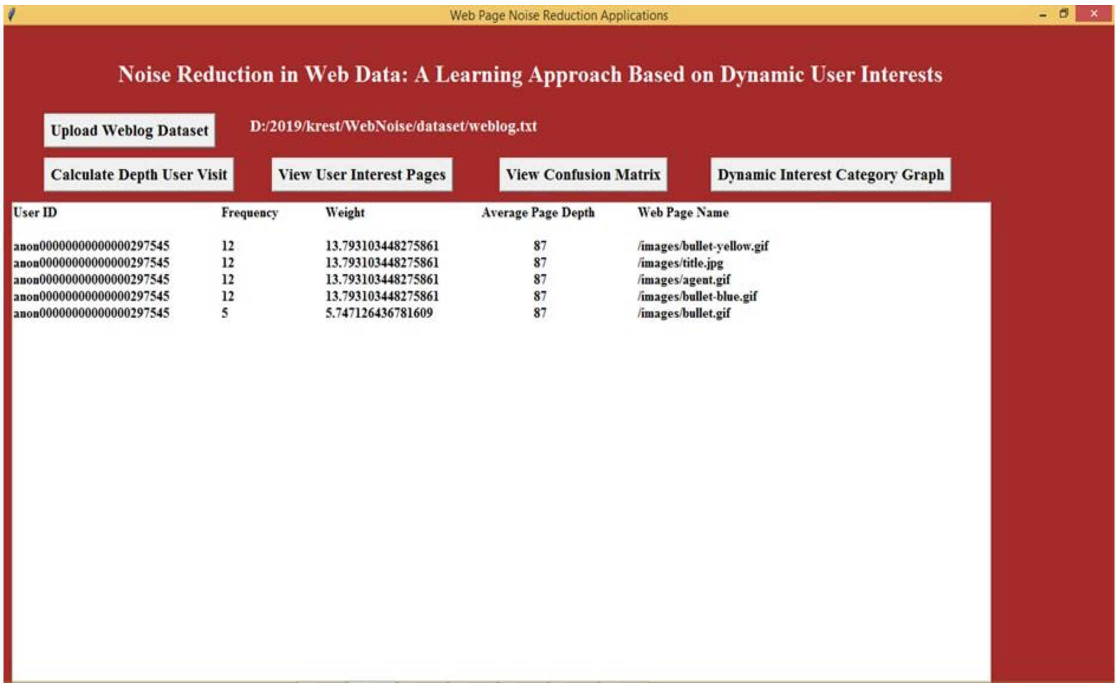
Fig7.7 Select User ID

In above screen whatever user Id I selected from command prompt I pasted in dialog box. Now click on ok button to get all frequent access pages.


Web Page Noise Reduction Applications
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Upload Weblog Dataset
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph

```
D:/2019/krest/WebNoise/dataset/weblog.txt

User ID                    Frequency   Weight               Average Page Depth   Web Page Name

anon0000000000000297577    2           9.090909090909092           22            /machines/manufacturers/Casio/
anon0000000000000297577    4           18.181818181818183          22            /manufacturers/Casio/
```

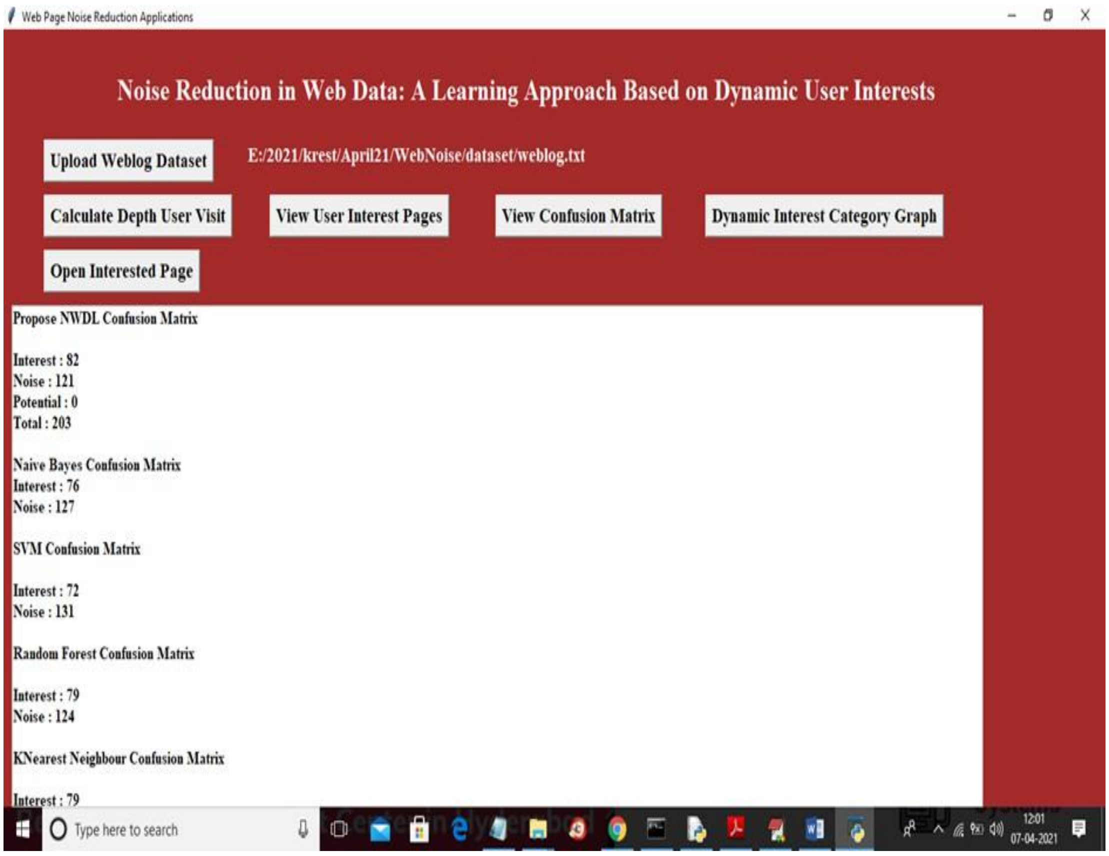
Fig 7.8 Calculate Depth User Visit


Web Page Noise Reduction Applications
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Upload Weblog Dataset
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph

```
D:/2019/krest/WebNoise/dataset/weblog.txt

User ID                    Frequency   Weight               Average Page Depth   Web Page Name

anon0000000000000297545    12          13.793103448275861          87            /images/bullet-yellow.gif
anon0000000000000297545    12          13.793103448275861          87            /images/title.jpg
anon0000000000000297545    12          13.793103448275861          87            /images/agent.gif
anon0000000000000297545    12          13.793103448275861          87            /images/bullet-blue.gif
anon0000000000000297545    5           5.747126436781609           87            /images/bullet.gif
```

Fig 7.9 Web Page Name

In above screens we can see user interest pages in the ‘Web Page Name’Details which are access by this user more frequently and will be added to user interested list. Now click on ‘View Confusion Matrix’ button to know no of interested and noise pages obtained by propose NWDL and existing SVM technique


Web Page Noise Reduction Applications
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Upload Weblog Dataset
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph
Open Interested Page
Type here to search

E:/2021/krest/April21/WebNoise/dataset/weblog.txt

```
Propose NWDL Confusion Matrix

Interest : 82
Noise : 121
Potential : 0
Total : 203

Naive Bayes Confusion Matrix
Interest : 76
Noise : 127

SVM Confusion Matrix

Interest : 72
Noise : 131

Random Forest Confusion Matrix

Interest : 79
Noise : 124

KNearest Neighbour Confusion Matrix

Interest : 79
```

Fig 7.10 View Confusion Matrix

In above screen we can see propose NWDL predict more interested pages compare to existing algorithm like SVM, Naïve Bayes, Random forest and KNN. In above screen with NWDL we got 82% predicted interested pages and naive bayes we got 72% predicted interested page. Scroll down above text area to view all details of KNN algorithm. Now click on 'Dynamic Interest Category Graphs' button to know various categories web pages in the form of graph.

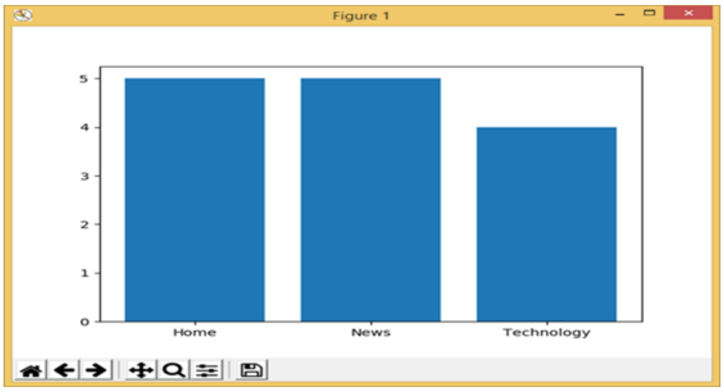
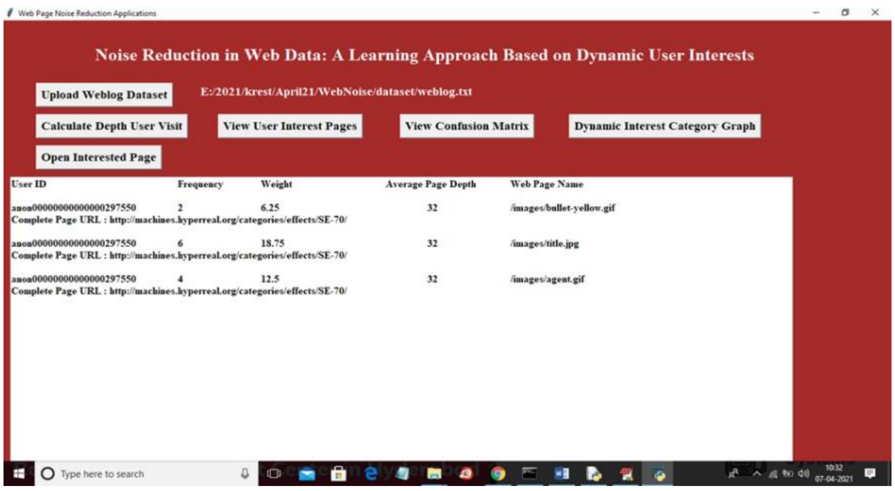


Fig 7.11 Dynamic Interest Category Graph

In above screen x-axis represents categories such as Home News and technology and yaxis represents total count of those News and technology and y-axis represents total count of those categories. Like this in dataset many categories are available but I am showing only 3 categories.Now if user wants to open and interested page then he has to click on ‘View User Interest Pages’ again to get below screen.

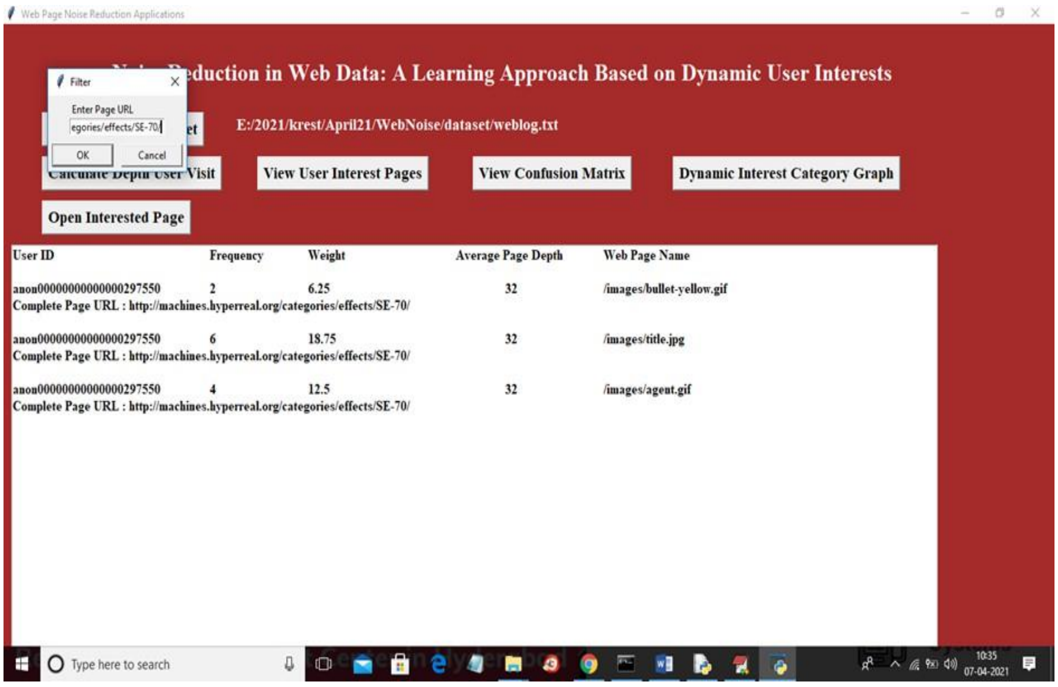

Web Page Noise Reduction Applications
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Upload Weblog Dataset
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph
Open Interested Page
Type here to search

E:/2021/krest/April21/WebNoise/dataset/weblog.txt

```
User ID                      Frequency    Weight       Average Page Depth    Web Page Name
anon0000000000000297550      2            6.25         32                    /images/bullet-yellow.gif
Complete Page URL : http://machines.hyperreal.org/categories/effects/SE-70/
anon0000000000000297550      6            18.75        32                    /images/title.jpg
Complete Page URL : http://machines.hyperreal.org/categories/effects/SE-70/
anon0000000000000297550      4            12.5         32                    /images/agent.gif
Complete Page URL : http://machines.hyperreal.org/categories/effects/SE-70/
```

Fig 7.12 Select URL

In above screen user can select any URL like below screen and then click on OpenInt.


Web Page Noise Reduction Applications
Filter
Enter Page URL
OK
Cancel
Noise Reduction in Web Data: A Learning Approach Based on Dynamic User Interests
Calculate Depth User Visit
View User Interest Pages
View Confusion Matrix
Dynamic Interest Category Graph
Open Interested Page
Type here to search

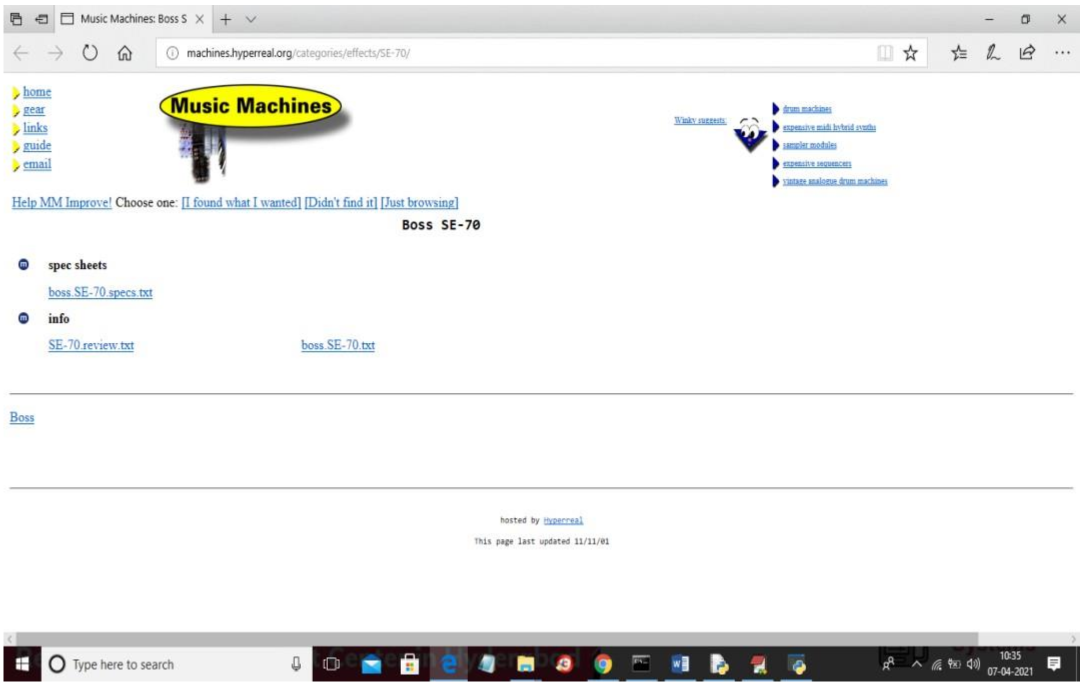
egories/effects/SE-70/

E:/2021/krest/April21/WebNoise/dataset/weblog.txt

```
User ID                      Frequency    Weight       Average Page Depth    Web Page Name
anon0000000000000297550      2            6.25         32                    /images/bullet-yellow.gif
Complete Page URL : http://machines.hyperreal.org/categories/effects/SE-70/
anon0000000000000297550      6            18.75        32                    /images/title.jpg
Complete Page URL : http://machines.hyperreal.org/categories/effects/SE-70/
anon0000000000000297550      4            12.5         32                    /images/agent.gif
Complete Page URL : http://machines.hyperreal.org/categories/effects/SE-70/
```

Fig 7.13 Paste URL

In above screen dialog box paste that URL and click OK button to view that page in browser like below screen. interested Page button.



Help MM Improve! Choose one: [I found what I wanted] [Didn't find it] [Just browsing]

Boss SE-70

spec sheets

boss.SE-70.specs.txt

info

SE-70.review.txt   boss.SE-70.txt

Boss

hosted by Hyperreal

This page last updated 11/11/01

Fig 7.14 Web page

8. SYSTEM TESTING

8.1 TESTING STRATEGIES

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

- Functional testing is centered on the following items: Valid Input: identified classes of valid input.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special testcases. In addition, systematic coverage pertaining to identify. Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such asspecification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box

.you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Test strategy and approach :-

Field testing will be performed manually and functional tests will be written in detail.

1. **Test objectives**

• All field entries must work properly.

• Pages must be activated from the identified link. • The entry screen, messages and responses must not be delayed.

2. **Features to be tested**

• Verify that the entries are of the correct format

• No duplicate entries should be allowed.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9. CONCLUSION AND FUTURE SCOPE

CONCLUSION:-

A machine learning algorithm capable of learning noise in web data prior to elimination is proposed. The starting point of this paper defines and identifies challenges with current research work in the noise web data reduction process. For example, elimination of noise in web data is based on preexisting noise data patterns and when user interests change, the stored noise data patterns can longer be relied, and hence not relevant. Moreover, current research works consider noise as any data that does not form part of the main web page. Therefore, it is difficult to identify and eliminate noise in web data without taking into dynamic interests of a web user.

FUTURE SCOPE :-

This project undertakes various steps to address the identified problems. Firstly, a machine learning algorithm that considers dynamic changes in user interests by learning the depth of a user visit in a specific web page is presented. Secondly, an algorithm that learns noise web data taking into account changes in user interests and evolving web data. The proposed algorithm is able to identify what users are interested in a given time, how they are searching and if they are interested in what they searching prior to elimination. Finally, the proposed tool contributes towards improving the quality of a web user profile.

10. REFERENCES

[1] L. Yi, B. Liu, and X. Li, “Eliminating Noisy Information in Web Pages for Data Mining,” in Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2003, pp. 296–305.
[2] C. Ramya, G. Kavitha, and D. K. Shreedhara, “Preprocessing: A Prerequisite for Discovering Patterns in Web Usage Mining Process,” ArXiv Prepr. ArXiv11050350, 2011.
[3] S. Dias and J. Gadge, “Identifying Informative Web Content Blocks using Web Page Segmentation,” entropy, vol. 1, p. 2, 2014. [4] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, “Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data,” SIGKDD Explor Newsl, vol. 1, no. 2, pp. 12–23, Jan. 2000.
[5] M. Jafari, F. SoleymaniSabzchi, and S. Jamali, “Extracting Users’Navigational Behavior from Web Log Data: a Survey,” J. Comput. Sci. Appl. J. Comput. Sci. Appl., vol. 1, no. 3, pp. 39–45, Jan. 2013.
[6] S.Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, “User profiles for personalized information access,” in The adaptive web, Springer, 2007, pp. 54–89.
[7] P. Peñas, R. del Hoyo, J. Vea-Murguía, C. González, and S. Mayo, “Collective Knowledge Ontology User Profiling for Twitter – Automatic User Profiling,” in 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013, vol. 1, pp. 439– 444.