iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VII **Month of publication:** July 2023

DOI: https://doi.org/10.22214/ijraset.2023.54767

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VII Jul 2023- Available at www.ijraset.com

I-Detox

Prof. Geocey Shejy[1], Varun Salvi[2], Athurva Sawant[3], Muskan Chhabria[4]
Dept of Computer Engineering, Vivekanand Education Society's Institute of Technology, Mumbai, Maharashtra

Abstract: ***Internet addiction refers to excessive internet use that interferes with daily life. Due to its negative impact on college students' study and life, discovering students' internet addiction tendencies and making correct guidance for them timely is necessary. In this system, we provide an approach to estimatinga person's internet addiction levels using their behavior data on the system. In detail, we consider a person's addiction to the internet as a hidden variable that affects students' daily time online together with other behavior. By predicting a person's daily time online, we will find a person's internet addiction levels. Finally, extensive experiments are conducted on a real- world dataset. The experimental results show the effectivenessof our method, and it is also consistent with some psychological findings. "I-detox" aims to provide real-time analysis of the exact number of websites visited by the user, giving them the freedom toforesee the probable results given the lack of schedule one followswhile browsing the internet. Therefore, this gives a standardized analysis of the web history of the user also inclining the number of times a user visits a particular website. Our solution considers the suitability of terming the user as an addict or not based onthe analysis provided upfront.***

Index Terms: addiction, analysis, negative impact, browsing, internet

I. INTRODUCTION

Addictive digital experiences are becoming extremely common in the tech industry and the adjective "addictive" is now considered one of the highest compliments paid to an artifact. Existing research has explored what is considered addictive, what are the approaches and strategies performed by tech companies that eventually lead to addiction, and how cultural and societal pressures have contributed to this phenomenon. Addiction to technology has grown into a fully recognized disorder [1]. "I-detox" can be used from the young to theold because, in the age of digitalization, one never knows the extent of addiction one is suffering. Our solution gives the user a clear idea about the number of websites being seen, the type of content being surfed upon, the category it belongs to, and the URL.

II. MOTIVATION

Our motivation is to aim and target the new age internet addiction. This addiction can spread from the young to oldand can be equally dangerous no matter what age group it subsides too. Our application aims to help individuals who want to analyze their browsing pattern, and decide the chancesof addiction..

III. LIMITATION EXISTING SYSTEM ORRESEARCH GAP

On analyzing the current systems existing in the market,we find that the system's do track their history into a file/text document but don't go further to categorize it. This may serve as a problem since not every website can be identified by its URL. "I - detox" not only provides the history informationinto an email but also categorizes the data into different types of websites adding onto the display of bag of words searched on a particular website. Thus, this makes the system user-friendly providing clarity which website provides what purpose and service and further analyzing how much timewas spent on a website. Browsing history/ or what does the user access internet for can prove a major aspect to analyzethe human-being as a whole and what he/she is thinking.On analyzing one can find if he/she is facing some mental problems. Currently, no system or software goes out of the way to analyze a user's browsing history and analyze their browsing pattern or so. That's what "I-detox" brings to the table.

IV. LITERATURE SURVEY

A study on rough set-aided feature selection for automatic web-page classification by Wakaki et.al gives a basic idea about how the web-pages are classified based on categories. [1]

Keras: The Python Deep Learning library by Son Eris¸imis a well built python library which makes our thoughts clear on the concepts of deep learning and provides an API for implementing it. [2]

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VII Jul 2023- Available at www.ijraset.com

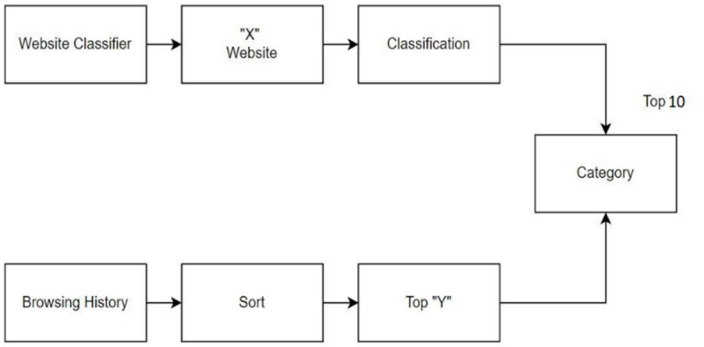


Fig. 1. Website Classification

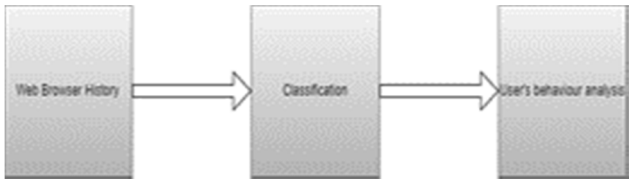


Fig. 2. Block diagram

A study on Internet addiction and its relation to psy- chopathology and self-esteem among college students by Kumar M et.al gives basic ideas about Internet use and its relation to psychopathology and self-esteem among college students.[3]
Large Scale Text Classification using Semi-supervised Learning Multinomial Naive Bayes Jiang Su,Jelber Shirabad,Stan Matwin A research paper which clearly specifies how multi-novial Naive Bayes can be used for effective classification with outstanding results. [4]
Internet Addiction Facts and Statistics By The Recovery Village. By Megan Hul. Medically Reviewed By Dr. Andrew Proulx, MD helps us to understand what addiction is and how it is caused. [5]
Text classification techniques: A literature review By M. Thangaraj, M. Sivakami. A paper which explains the back- ground of various classification techniques their uses way of functioning and their implementation [5].
Internet addiction and problematic Internet use: A system- atic review of clinical research by Daria J Kuss and Olatz Lopez-Fernandez gives a comprehensive overview of clinical studies on the clinical picture of Internet-use related addictionsfrom a holistic perspective. A literature search was conducted using the database Web of Science. [6]

V. METHODOLOGY APPLIED

To provide the user with his/her browsing history through email. The data would be processed. Basic analysis will be performed on the data. The system would be trained for analysis, the training data would be regressed and passed tothe user. Model would be assessed using fresh test data.[4][5] Training set: This set of data must be used to train the model for it to accurately provide the search history. Testing set:The model will be tested using some test data after it hasbeen trained. After that, the training data will be passed intoa regressor, which is used to predict continuous values. [1][2] Finally, our model will be assessed using fresh test data.

A. Specific Requirements

1) Software Requirements

a) Language: Python
b) IDE: Spyder
c) ML Model Platform: ColabHardware Requirements:
d) 2Gb Ram
e) Intel Pentium Gold Processor and above / Ryzen Athlonand above
f) 512 Mb Physical Storage and above

2) Algorithm

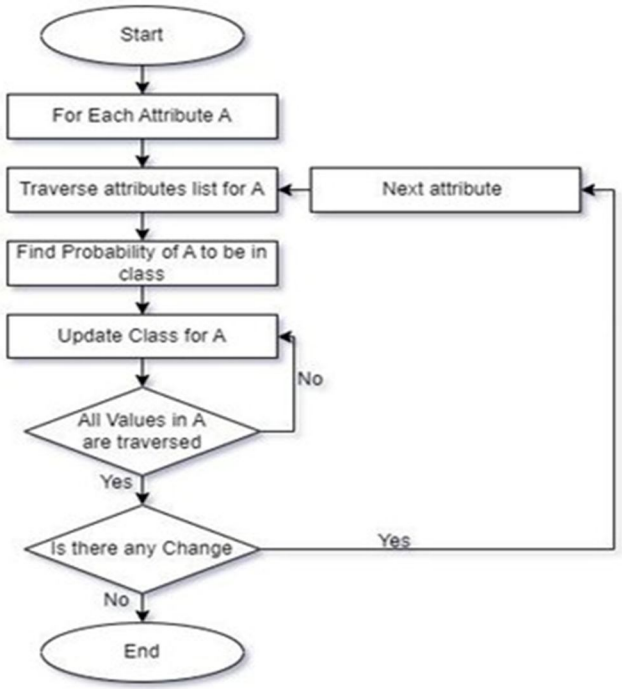
a) *Multinomial NB:* An technique to Bayesian learning thatis well-liked in Natural Language Processing (NLP) is the Multinomial Naive Bayes algorithm. Using the Bayes princi- ple, the programme makes an educated prediction about thetag of a text, such as an email or news article. It determines thelikelihood of each tag for a particular sample and outputs the tag with the highest likelihood. Each feature being classified by the Naive Bayes classifier is distinct from every otherfeature, which unites the various algorithms that make up the classifier. The inclusion or removal of one characteristic does not depend on the presence or absence of another feature. The Naive Bayes method is an effective technique for examining text input and

resolving issues involving several classes. It is vital to first understand the Bayes theorem concept because the Naive Bayes theorem is built on it. The Bayes theorem, whichwas created by Thomas Bayes, calculates the probability of anevent happening based on information about its circumstances.We determine the chance of class A when predictor B isavailable. Its foundation is the following equation: P(A—B) = P(A) * P(B—A)/P(B). Because all that is required to accomplish it is probability calculation, it is straightforward. Both continuous and discrete data can be used in this strategy. Real-time applications can be predicted using this simple method. Large datasets may be handled with ease and it is highly scalable. The prediction accuracy of this method is less accurate than other probability algorithms. Regression is not applicable. It is impossible to estimate numerical values using the Naive Bayes technique; it can only be used to classify textual input.

b) *Beautiful Soup:* Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works withyour favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.[4][1]

3) Website Classification

First, the local system is used to access the web browserhistory. The URL, the number of visits, and the time stamp (the most recent visit) are all included in the browser history. The model categorises the urls using this web browser history.The URLs are categorised using lovely soup, which uses the url's html description to do so. The user report is producedfor a period of one month based on this categorisation. The concerned party might examine user behaviour after receiving the report.[1]



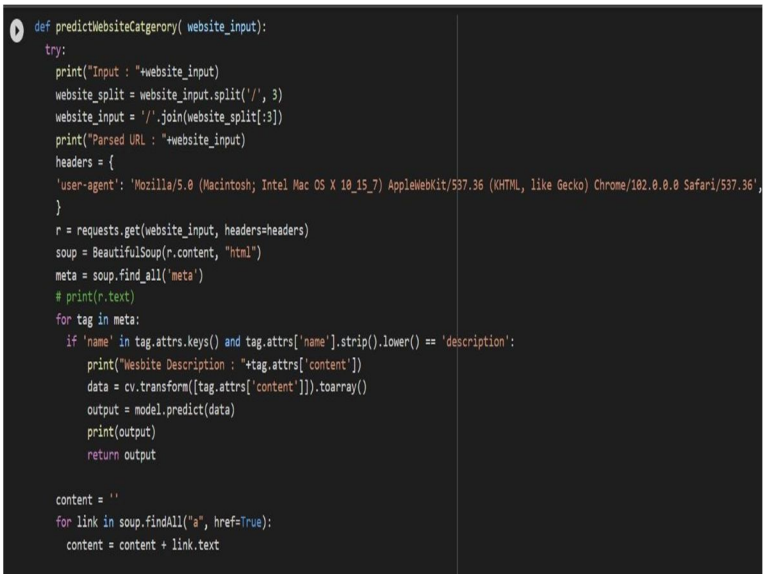
4) Count Vectorizer

Count Vectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector based on the frequency (count) of each word that occursin the entire text. This is helpful when we have multiple such texts, and we wish to convert each word in each text into vectors (for using in further text analysis)

5) Results

Web scraping is done in this section using a technique calledbeautiful soup. We ask the user to enter the website url first. Once the url has been acquired, it is split on the third '/' and the remaining portion is split across the '/' before being joinedto provide the parsed url that has been optimised. The header for the URL is taken based on the browser once the url has been acquired. The header is obtained from the website-input and added to "r," after which the html page is obtained from the content of "r." Metadata is present on the HTML page.The categorisation is based on this meta data.The 'description' element, which holds the text for the

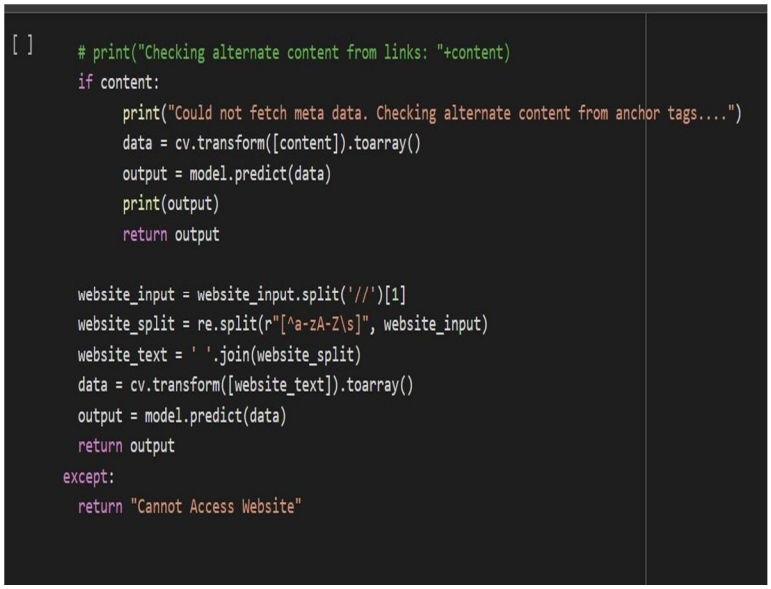
url classification, isnow acquired from the meta data. The description's data is then transformed into an array and saved as an array. The model is then given this content to determine the website's categorization. The anchor element that contains the url link makes up the link. This link is included in the material and kept there. The contents of the url itself are broken and kept asan array and utilised for prediction if there is a website with security measures that make it difficult to access the meta data. As a result, the website is classified in this way. Additionally, a "cannot access website" warning is displayed if a website is inaccessible.



```
def predictWebsiteCatgerory( website_input):
  try:
    print("Input : "+website_input)
    website_split = website_input.split('/', 3)
    website_input = '/'.join(website_split[:3])
    print("Parsed URL : "+website_input)
    headers = {
    'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36',
    }
    r = requests.get(website_input, headers=headers)
    soup = BeautifulSoup(r.content, "html")
    meta = soup.find_all('meta')
    # print(r.text)
    for tag in meta:
      if 'name' in tag.attrs.keys() and tag.attrs['name'].strip().lower() == 'description':
          print("Wesbite Description : "+tag.attrs['content'])
          data = cv.transform([tag.attrs['content']]).toarray()
          output = model.predict(data)
          print(output)
          return output

    content = ''
    for link in soup.findAll("a", href=True):
      content = content + link.text
```

Fig. 4. Taking website as input.



```
[ ]
    # print("Checking alternate content from links: "+content)
    if content:
          print("Could not fetch meta data. Checking alternate content from anchor tags....")
          data = cv.transform([content]).toarray()
          output = model.predict(data)
          print(output)
          return output


    website_input = website_input.split('//')[1]
    website_split = re.split(r"[^a-zA-Z\s]", website_input)
    website_text = ' '.join(website_split)
    data = cv.transform([website_text]).toarray()
    output = model.predict(data)
    return output
  except:
    return "Cannot Access Website"
```

Fig. 5. Fetching the website.

VI. CONCLUSION

In this project, we propose an artificial intelligence-based system that can help the recipient analyze for himself his use of the internet. This system can work with people of all age groups with parents using it to keep a check on their ward's internet life to the people who find themselves lost in thishuge digital world. This system would help the user have a personalized analysis and a fact check about his presence on the internet, the activities he performs, the content he streams to the games he plays because of the availability of timestampsrecording details of every thin activity. Thus, the question in person with the help of our system can reconcile his use of theinternet thus helping him curb and storm out his way through these addiction demons.

```
[ ] website_input=input("Enter a website: ")
    output = predictWebsiteCatgerory(website_input)
    print(output)

    Enter a website: https://imfs-lms.azurewebsites.net/#/view/eBook
    Input : https://imfs-lms.azurewebsites.net/#/view/eBook
    Parsed URL : https://imfs-lms.azurewebsites.net
    Wesbite Description :
    ['Education']
    ['Education']
```

Fig. 6. Prediction of Website category

```
[ ] top_10_website
```

	url	visit_count	Category
4	https://mail.google.com/mail/u/0/#inbox	33264	[Computers and Technology]
5	https://github.com/shawon100	27528	[Computers and Technology]
9	https://www.youtube.com/	14792	[Streaming Services]
13	http://www.cricbuzz.com/	13328	[Sports]
16	https://web.facebook.com/	11725	[Streaming Services]
18	https://mail.google.com/mail/u/0/?tab=wm#inbox	9718	[Computers and Technology]
22	http://facebook.com/	8404	[Streaming Services]
24	http://codeforces.com/	7677	Cannot Access Website
28	https://drive.google.com/drive/	6754	[Streaming Services]
33	http://shawon.mektekbd.com/	5227	Cannot Access Website

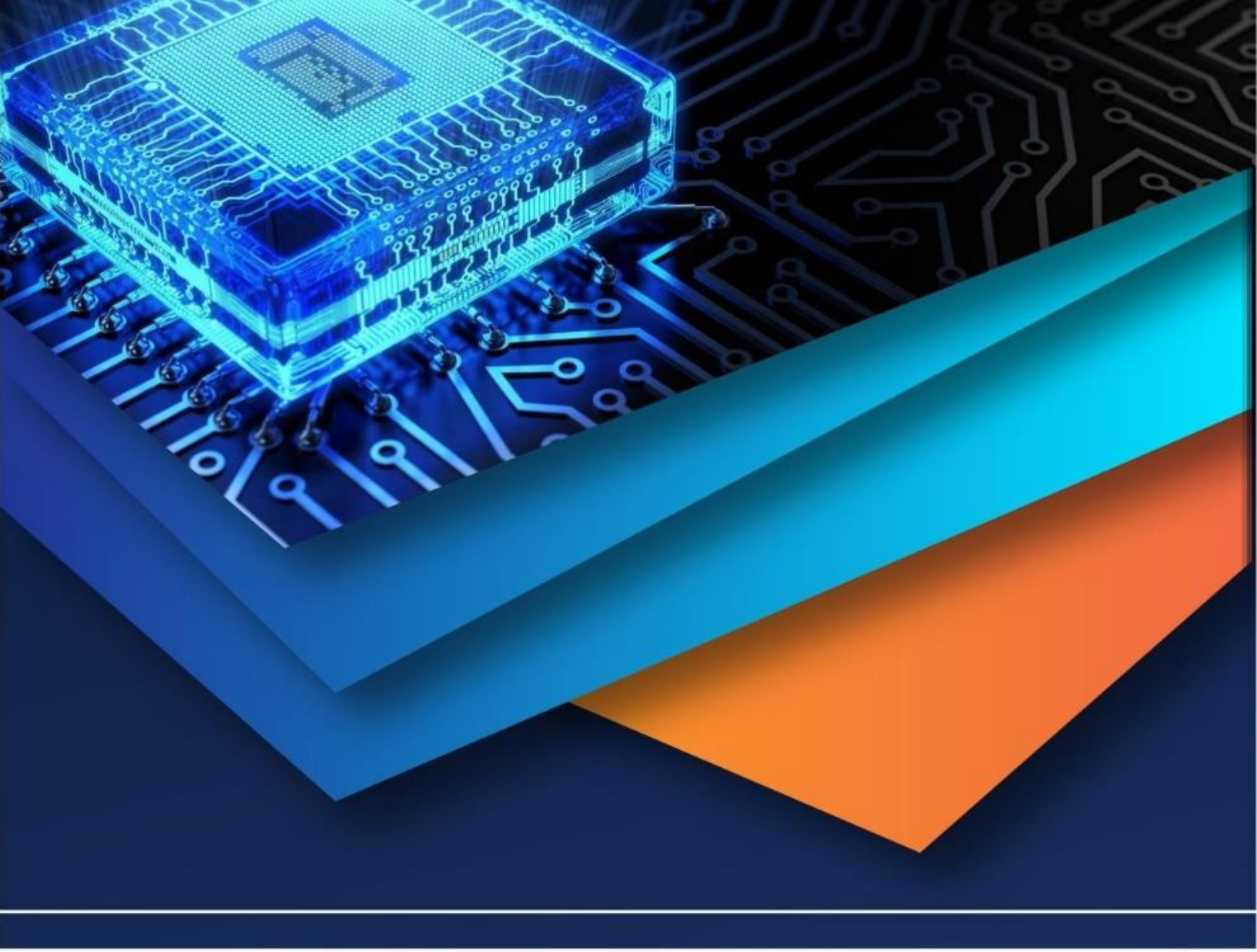
Fig. 7. Website categorization along with visit count

URL	VisitCount	Timestamp	Category
docs.google.com	835	2023-04-07 08:48:29.825409	Education
docs.google.com	601	2023-03-20 16:04:48.890087	Education
mail.google.com	290	2023-04-01 19:57:13.441958	Computers and Technology
mail.google.com	156	2023-04-07 17:53:50.861456	Computers and Technology
docs.google.com	128	2023-03-24 12:21:19.307014	Education
docs.google.com	126	2023-03-24 12:21:18.400705	Education
docs.google.com	122	2023-03-13 11:30:14.428573	Education
imfs-lms.azurewebsites.net	121	2023-04-05 12:14:33.105605	Education
docs.google.com	120	2023-03-21 05:38:07.964968	Education
docs.google.com	118	2023-03-19 13:50:52.944940	Education
imfs-lms.azurewebsites.net	100	2023-03-30 12:11:41.383998	Education
docs.google.com	98	2023-03-13 15:06:28.162702	Education
mail.google.com	94	2023-03-24 12:43:29.675323	Computers and Technology
docs.google.com	93	2023-04-05 05:24:21.234119	Education
docs.google.com	91	2023-04-05 12:14:27.391168	Education
docs.google.com	90	2023-03-20 17:53:31.030928	Education
mail.google.com	86	2023-04-07 15:18:57.476324	Computers and Technology
docs.google.com	78	2023-03-22 13:06:13.318453	Education
www.youtube.com	74	2023-04-06 12:56:45.531160	Streaming Services
docs.google.com	74	2023-03-22 05:34:33.397792	Education

Fig. 8. Report Generated

REFERENCES

[1] Wakaki, T., Itakura, H., Tamura, M., Motoda, H., Washio, T. (2006).A study on rough set-aided feature selection for automatic web-page classification. Web Intelligence and Agent Systems: An International Journal, 4, 431–441
[2] Keras: The Python Deep Learning library, https://keras.io/, Son Eris¸im:Mayıs 2018.
[3] Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes Jiang Su, Jelber Shirabad, Stan Matwin https://icml.cc/2011/papers/93¡cmlpaper.pdf Text classification techniques: A literature review. M. Thangaraj, M. Sivakami https : //www.proquest.com
[4] Internet Addiction Facts and Statistics by The Recovery Village —Editor Megan Hull
[5] Internet Addiction: Definition, Assessment, Epidemiology and??ClinicalManagement by Donald Black.
[6] https://www.helpguide.org/articles/addictions/smartphone-addiction.htm



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY