

A graphical data analyzer that analysis trend of a portal traffic based on big data
stored by Google Analytic



Inspiring Excellence

Student Name: S.M. Zamshed Farhan

Student ID: 15173001

A project report submitted in fulfillment of the requirements for the award of
the degree of Master of Engineering in Computer Science and Engineering

Department of Computer Science and Engineering

BRAC University

December, 2016

Declaration

I hereby certify that this material, which I now submit for assessment on the programme Master of Engineering in Computer Science and Engineering is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Student Name: S.M. Zamshed Farhan

Student ID: 15173001

Date:

A graphical data analyzer that analysis trend of a portal traffic based on big data
stored by Google Analytic

Project report

A project report submitted in fulfillment of the requirements for the award of
the degree of Master of Engineering in Computer Science and Engineering

By

S.M. Zamshed Farhan

Approved by

Dr. Amitabha Chakrabarty, Supervisor

BRAC University
66 Mohakhali,
Dhaka 1212,
Bangladesh

December, 2016

Abstract

Google Analytics is a web analytics service offered by Google that tracks and reports website traffic, as per every pages to tell it to track, as well it's presenting visual representation of that data. But, to do so demands expertise on Google Analytical data manipulation. Our project helps the users to visualize the data as their own demand. The users also store any instance of traffic data, as well draw a comparison between any two instances. The main goal of this project to provide a simplified structure to analyze Google analytic data analysis users' trend.

Acknowledgement

I would first like to thank my supervisor Dr. Amitabha Chakrabarty of the Computer Science and Engineering Department at BRAC University. The door to Dr. Amitabha Chakrabarty office was always open whenever I ran into a trouble spot or had a question about my project or writing. He consistently allowed this project to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank the Information Technology department, icddr,b for providing us the technical help to work with this project. Without their cordial help this project could not have been successfully conducted.

Contents

Declaration.....	i
Abstract.....	iii
Acknowledgement	iv
List of Figures.....	vii
Chapter 1.....	1
1.1 Background.....	1
1.2 Motivation.....	1
1.3 Problem statement.....	1
1.4 Solution.....	2
1.5 Methodology	2
1.6 Who are the stakeholders?	2
1.7 Outline of project	2
Chapter 2.....	4
Literature Review.....	4
2.1 Web Analytics (a brief overview).....	4
2.2 Quantitative and Qualitative Data.....	4
2.3 The Continuous Improvement Process	4
2.4 Measuring Outcomes	5
2.5 What Google Analytics Contributes	6
2.6 Creating an Implementation Plan.....	7
2.6.1 Create an Account and Configure Our Profile.....	8
2.6.2 Configure the Tracking Code and Tag Pages	8
2.6.3 Tag Marketing Campaigns.....	8
2.6.4 Create Additional User Accounts and Configure Reporting Features	9
2.6.5 Perform Optional Configuration Steps.....	9
Chapter 3.....	10
How Google Analytics Works	10
3.1 Data Collection and Processing	10
3.2 Reports	12
3.3 About the Tracking	13
3.4 Understanding Pageviews	14
3.5 Platform used for developing referral application	15
3.5.1 PHP, MySQL and CanvasJS.....	16
3.5.2 Google Analytic API.....	17
Chapter 4.....	18
Implementation	18

4.1	Google Analytic API.....	18
4.2	Use of GAPI class.....	23
4.3	User Interface.....	25
4.3.1	API Information.....	25
4.3.2	Analytical Data.....	26
4.3.3	Data Archive.....	31
4.3.4	Trend Analysis.....	32
4.4	IKR (Institutional Knowledge Repository).....	36
Chapter 5.....		38
Conclusion & Future Work.....		38
References.....		39

List of Figures

Figure 1: Outline of the project	3
Figure 2: The web analytics process: measure, analyze, and change	5
Figure 3: The Map Overlay report shows traffic from individual countries	6
Figure 4: Viewing a segment of traffic along with all traffic in Google Analytics	7
Figure 5: Google Analytics processing flow	10
Figure 6: This report shows the conversion rate (a metric) for the City dimension	12
Figure 7: Top Content report showing pageview “names”	15
Figure 8: Components of Google Analytic	18
Figure 9: One of the web page where to put tracking code	20
Figure 10: JS tracking code in page	21
Figure 11: Creation of API file on Google Analytic	22
Figure 12: List of available service accounts	22
Figure 13: Download the API file in p12 format	23
Figure 14: Activity diagram for Google Analytic Project	24
Figure 15: Analytic Account Information are being collected through API	25
Figure 16: Analytical Data	26
Figure 17: Data Visualization	27
Figure 18: Visual representation of Analytical data (Spline chart)	31
Figure 19: Trend Analysis between real-time data and archive data	32
Figure 20: Trend Analysis between archive data	33
Figure 21: Trend Analysis - Quarterly News Analysis	34
Figure 22: Communities in IKR	35
Figure 23: Page views and quantities per community to analysis	36
Figure 24 : Analyzed data to compare among the popularity of all communities, showing the items view in every communities, number of total items per communities and the percentage of popularity.	37

Chapter 1

Introduction

1.1 Background

Google Analytic is a web analytics service offered by Google that tracks and reports website traffic. Google launched the service in November 2005 after acquiring Urchin. Google Analytics is now the most widely used web analytics service on the Internet. Google Analytics is offered also in two additional versions: the subscription based Google Analytics 360, previously Google Analytics Premium, targeted at enterprise users and Google Analytics for Mobile Apps, an SDK that allows gathering usage data from iOS and Android Apps. [1] The aim of this project is to prepare a decision set based on that analytical data which make a sense for the site.

1.2 Motivation

Though Google Analytic is a very strong tool to visualize the analyze data, but it's so easy for general users to customize the report. As well, Google Analytic collect traffic data based on URL (Uniform Resource Locator), so it's so hard to analyze the category-wise data. For example, our website a menu category as News which publishing 3-4 news per month to inform the target people about our current research themes, here the list view of news is a URL, and every news has separate URL, so it's very hard to analyze the traffic for all news at a time, as all have separate URL.

1.3 Problem statement

Google Analytic may visualize data of specific instance of time, but it has no provision to do that for multiple instances at a time. The aim of this project is to give a glimpse to solve of these limitations. As well, we have an organization Institutional Knowledge Repository (IKR), which contains sets of publications grouped by MeSH (Medical Subject Headings). [11] Additional goal of this project is too to help the Library to select the MeSH appropriately.

1.4 Solution

This project aims to propose a prototypical implementation of a system that can be reduce the hassle to visualization of data and their comparisons. This system contains the selective area of traffic data on a separate database to create room for general users.

1.5 Methodology

An web-based application that can be used by the non-technical users so that they can access formatted web traffic data to analyze and prepare a decision set about the presentation of contents of the site. This application is being used as simple graphical tool which shares the Google Analytic data to take decision about content formation.

1.6 Who are the stakeholders?

Actually this project is an organizational initiative, where the Communication Department and Library is looking for a easy environment to analyze Google Analytic data to prepare a set of decisions on web contents preparing, and presenting.

1.7 Outline of project

Figure 1 shows a quick overview of the report structure. To meet the objectives, a literature review was conducted, of which the results can be found in Chapter 2. The part of the literature review focuses on overall concept of Google Analytic and the factors that shape presentation and the second part focuses into software engineering challenges to develop a third-party web traffic analyzer identified by current researches.

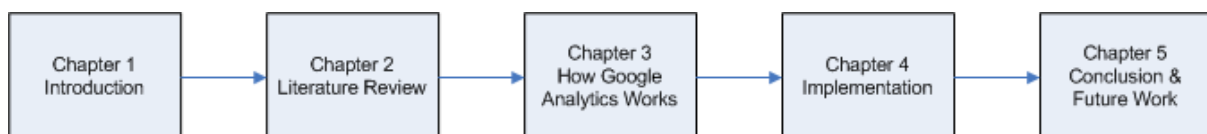


Figure 1: Outline of the project

Chapter 3 states the methodological approach to this paper. The functional and non-functional requirements, including the data model, system design of the proposed tool are introduced in Chapter 3. The implementation of the traffic analytic application, user interface evolution, is subject of Chapter 4. Chapter 5 concludes the report by discussing the findings and proposing future research endeavors in the area of web traffic application.

Chapter 2

Literature Review

2.1 Web Analytics (a brief overview)

The analysis of qualitative and quantitative data from our website and the competition, to drive a continual improvement of the online experience that our customers, and potential customers have, which translates into our desired outcomes (online and offline). This definition encapsulates three main tasks every business must tackle when doing web analytics: [2]

- Measuring quantitative and qualitative data
- Continuously improving our website
- Aligning our measurement strategy with our business strategy

2.2 Quantitative and Qualitative Data

Web analytics is not possible without data. But many organizations fail to realize that they need many different types of data to understand the performance of their website. Tools like Google Analytics, Omniture, WebTrends, and Yahoo! Web Analytics generate quantitative, or click stream, data. This data identifies where website traffic comes from and what it does on the site. It more or less tells what happened on a website. While click stream data is critical, we must collect more than quantitative data—we must also collect qualitative data. While quantitative data describes what happens on our website, qualitative describes why it happens. Qualitative data comes from different sources, like user interviews and usability tests. But the easiest way to get qualitative data is through surveys. [4]

2.3 The Continuous Improvement Process

Web analytics is used to drive a continual improvement of the online experience that our customers, and potential customers have. All of the data and analysis must drive a continuous improvement process. This is the most critical part of web analytics. We must take action on the data. That's the whole purpose of web analytics—to improve over time. Figure 2 shows a very basic representation of the web analytics process.

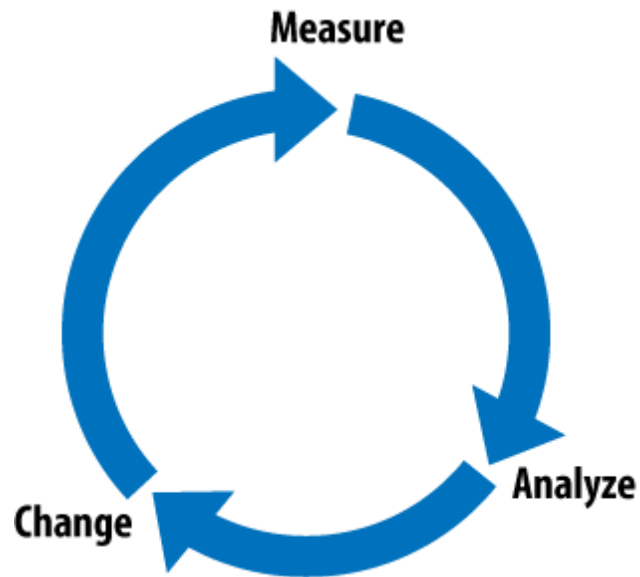


Figure 2: The web analytics process: measure, analyze, and change[2]

2.4 Measuring Outcomes

The final part of web analytics is that it translates into our desired outcomes (online and offline). The entire goal of the web analytics process is to increase our desired business outcomes. This means measuring metrics that relate directly to our overall business goals. Every website exists for a reason, and our measurement strategy must align with the business goals of the website.

2.5 What Google Analytics Contributes

Google Analytics provides a core set of tools that supports some of the primary tasks that web analysts perform.

First and foremost, Google Analytics tracks many standard website metrics, like visits, unique visitors, page views, bounce rate, and abandonment rate. But, more importantly, it can track business outcomes, called goals. Remember, we want to move beyond tracking basic traffic to our websites and begin understanding if our websites are adding to the bottom line of our business. [6]

A simple example of segmentation is viewing website traffic based on the physical location of the visitors. Google Analytics does this using the Map Overlay report, shown in Figure 3.

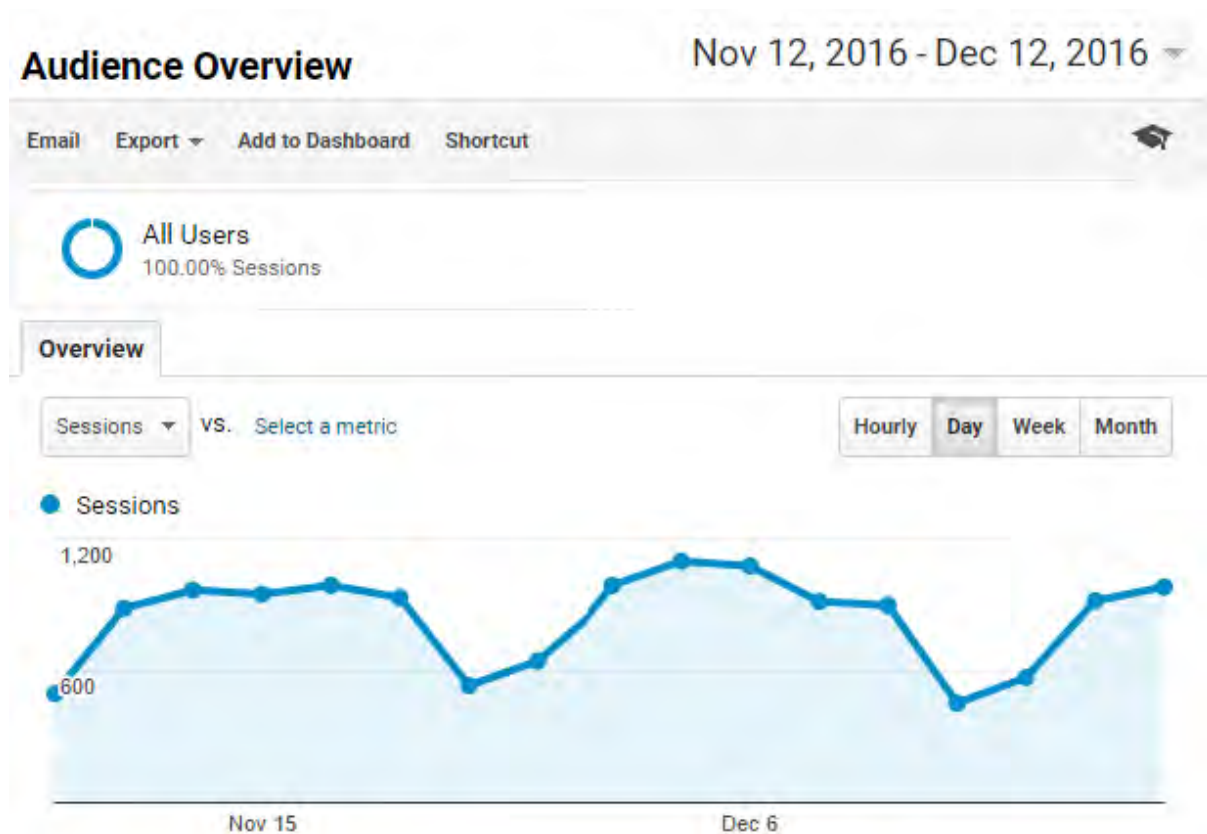


Figure 3. The Map Overlay report shows traffic from individual countries

In this case, the dimension is the country. The metrics for that dimension are shown in the columns of the report. Now notice the tabs at the top of the report. The Goal tab displays conversions for the same dimension of traffic. So, if we click the Goal Set 1 tab, Google Analytics will display conversions for each goal for each country.

This is the way all Google Analytics reports work. Every row of data is a different value of the dimension of traffic. For example, in the Traffic Sources report, each row in the table is a different source of traffic (organic search, marketing campaigns, etc.).

This is a complicated segmentation that we can build and apply in real time! The result is the ability to view the segment we created above, along with other segments of website traffic (Figure 4).

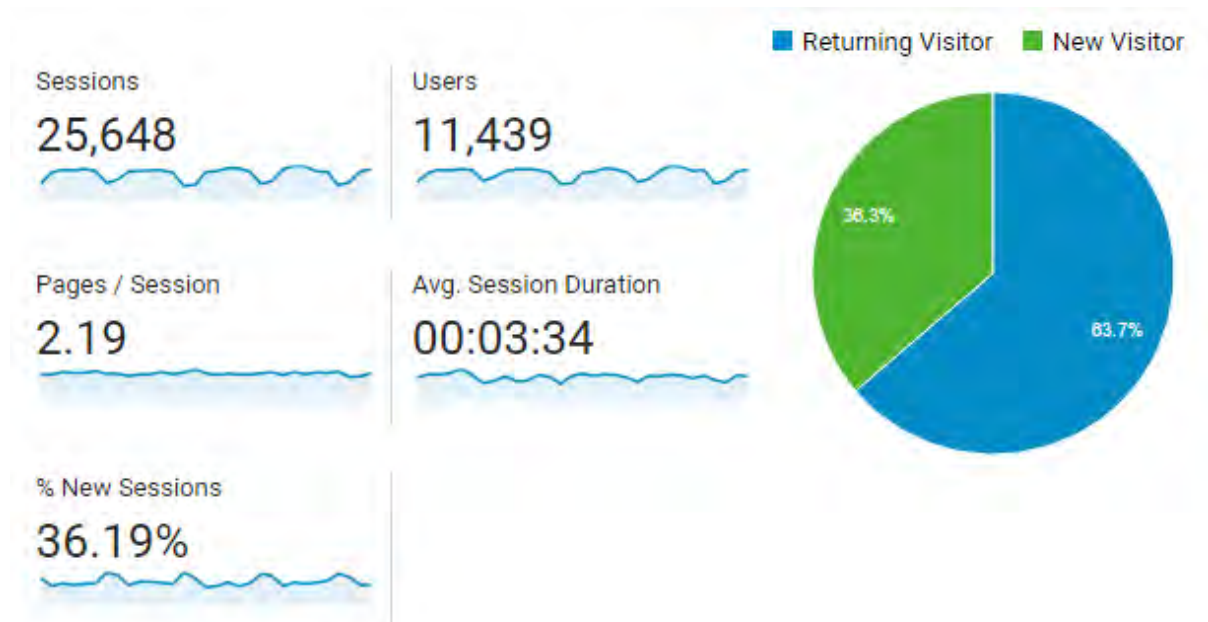


Figure 4: Viewing a segment of traffic along with all traffic in Google Analytics

This ability to drill down and focus on various segments of traffic is key to all analysis.

We want to identify the segments of traffic that are performing well and determine how to promote those segments. We also want to identify the segments of traffic that suck and figure out how to fix them.

Advanced Segmentation is not the only tool that helps facilitate analysis. Google Analytics also contains a custom reporting tool that can greatly simplify our daily reporting and even help simplify common segmentations.

2.6 Creating an Implementation Plan

Google Analytics is a business intelligence tool and, because every business has different data needs, our implementation may be very different from someone else's. Do not believe that we can simply slap some tags on the site and collect valid data. It is very rare that an implementation involves only page tagging. There are many configuration steps required to generate accurate, actionable data.

With that said, there are some standard things that everyone should do to get reliable data for analysis. Implementing Google Analytics does take some planning and foresight. The Google Analytics

support documentation does contain a rough implementation guide that includes the various steps to get Google Analytics installed and running. I have modified that process as follows: [2]

1. Gather and document business requirements.
2. Analyze and document website architecture.
3. Create a Google Analytics account and configure profiles.
4. Configure the Google Analytics tracking code and tag website pages.
5. Tag marketing campaigns.
6. Create additional user accounts and configure the following reporting features: Report access, Automated email report delivery, Reporting customizations (Custom Reports, Advanced Segments)
7. Perform the following optional configuration steps: Enable e-commerce transaction tracking, Implement event tracking, Implement custom variables

2.6.1 Create an Account and Configure Our Profile

Once we've got all of our business requirements, it's time to start working in Google Analytics. Begin by creating an account. If we've already got a Google Analytics account, there is no need to create another one. Once we have an account, configure our profile settings, such as Site Search, Filters, and Goals.

2.6.2 Configure the Tracking Code and Tag Pages

We'll discuss how to configure the tracking code later, but, briefly, based on our website architecture, we may need to alter the tracking code to compensate for things like sub-domains or multiple domains. Luckily, there is a code configuration tool that makes changing the JavaScript rather simple.

Once we have created and configured the profiles and the tracking code, it's time to tag the pages. Because most sites use some type of template system, like WordPress, Drupal, or some custom content management system, this makes tagging pages fairly easy. In most cases, we can place the page tag in our footer template, and in about three hours we should start to see data. If our website does not use a templating system, we will need to manually add the tag at the bottom of all the pages on our site.

2.6.3 Tag Marketing Campaigns

Tagging marketing campaigns is one of the most critical parts of configuring Google Analytics. This is the process of identifying our different marketing activities (like paid search, display advertising,

and email marketing) to Google Analytics. We do this using a process called link tagging. If we do not tag our marketing campaigns, it will be impossible to measure the success of our online marketing initiatives.

2.6.4 Create Additional User Accounts and Configure Reporting Features

Once analytics is up and running and we've started to collect data, it's time to configure various features that provide access to data. This is the time to create user accounts so coworkers and others can access analytics.

This is also the time to configure some of the reporting tools that Google Analytics provides. Features like automated report email messages and custom reports can greatly reduce the time it takes to generate any standard reports that an organization may need.

2.6.5 Perform Optional Configuration Steps

There are many Google Analytics features that generate additional data. E-commerce tracking, custom variables, and event tracking are optional features that all collect other types of data. While it is not necessary to implement these features, they often provide additional information that can provide we with more insight.

For example, custom variables can collect demographic information about our site visitors, event tracking can measure how people interact with different types of content, and e-commerce tracking can collect revenue and transactional data in real time. Some organizations may make these features a high priority based on the metrics they provide. If we're one of them, these implementation steps may not be optional for us. Schedule the implementation of these features based on our priority and implementation resources.

Chapter 3

How Google Analytics Works

Understanding the Google Analytics architecture—how it collects data, processes data, and creates reports—is the key to understanding many of the advanced topics that we will discuss later in this book. Google Analytics can collect data from a number of different platforms using different tracking technologies, which makes things complicated.

Google Analytics is no longer a simple "hit collector" for websites, but rather an information aggregation system that collects data from standard websites, mobile websites, Adobe Air applications, and iPhone and Android apps. Google has progressively added more data collection methods as technology has driven new and different ways of distributing content to people. [2]

3.1 Data Collection and Processing

Figure-5 shows how Google Analytics collects, processes, and displays data. Google Analytics uses a common data collection technique called page tags. A page tag is a small piece of JavaScript that we must place on all the website pages we want to track. We affectionately call this code the Google Analytics Tracking Code (GATC). If we do not place the code on a page, Google Analytics will not track that page. The data collection process begins when a visitor requests a page from the web server. The server responds by sending the requested page back to the visitor's browser (step 1 in Figure-5). As the browser processes the data, it contacts other servers that may host parts of the requested page, like images, videos, or script files. This is the case with the GATC.

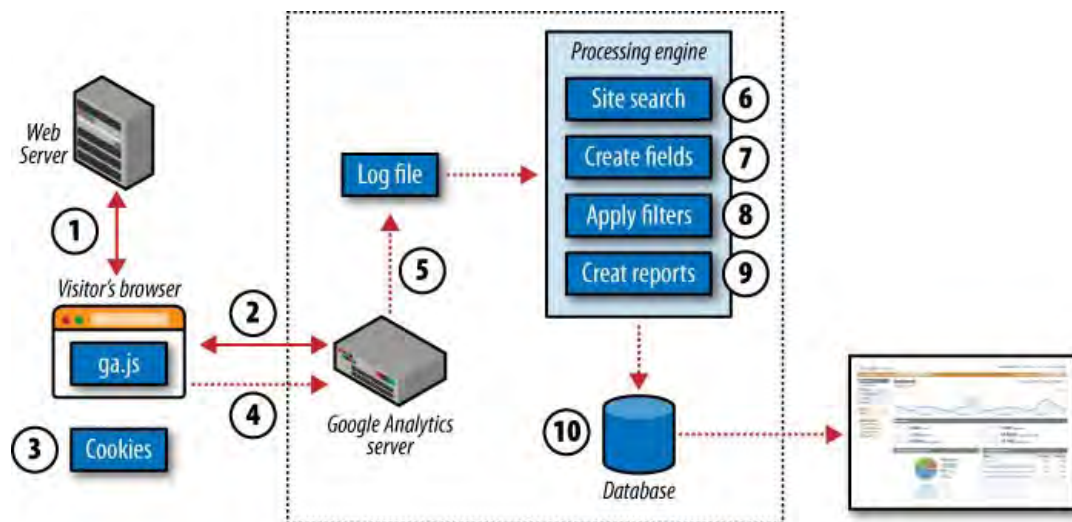


Figure 5: Google Analytics processing flow

When the visitor's browser reaches the GATC, the code begins to execute. During execution, the GATC identifies attributes of the visitor and her browsing environment, such as how many times she's been to the site, where she came from, her operating system, her web browser, etc.

After collecting the appropriate data, the GATC sets (or updates, depending on the situation) a number of first-party cookies (step 2), which are discussed later in this section. The cookies store information about the visitor. After creating the cookies on the visitor's machine, the tracking code waits to send the visitor data back to the Google Analytics server.

While the data is collected and the cookies are set, the browser is actively downloading a file named `ga.js` from a Google Analytics server (also step 2). All of the code that Google Analytics needs to function is contained within `ga.js`. Once the `ga.js` file is loaded in the browser, the data that was collected is sent to Google in the form of a pageview. A pageview indicates that a visitor has viewed a certain page on the website.

There are other types of data, like events and e-commerce data, that can be sent to Google Analytics (we will discuss these later). The pageview is transmitted to the Google Analytics server via a request for an invisible GIF file (step 4) named `__utm.gif`. Each piece of information the GATC has collected is sent as a query-string parameter in the `__utm.gif` request, as shown below:

```
http://www.google-analytics.com/__utm.gif?utmwv=4.6.5&utmhn=1881501226&utmhl=1881501226&utmcc=UA-XXXX-1&utmccn=direct&utmcmd=none&utmfl=10.0%20r42&utmdt=Analytics%20Talk%20by%20Justin%20Cutroni&utmhid=465405990&utmr=-&utmp=%2Fblog%2F&utmz=3D32856364.1269919681.1.1.utmcsr%3D(direct)%7Cutmccn%3D(direct)%7Cutmcmd%3D(none)%3B&gaq=1
```

When the Google Analytics server receives this pageview, it stores the data in some type of temporary data storage. Google has not indicated exactly how the data is stored, but we know that there is some type of storage for the raw data. Think of this data storage as a large text file or a logfile (step 5).

Each line in the logfile contains numerous attributes of the pageview sent to Google. This includes:

- When the data was collected (date and time)
- Where the visitor came from (referring website, search engine, etc.)
- How many times the visitor has been to the site (number of visits)
- Where the visitor is located (geographic location)
- Who the visitor is (IP address)

After the pageview is stored in the logfile, the data collection process is complete. The data collection and data processing components of Google Analytics are separate. This ensures Google Analytics will always collect data, even if the data processing engine is undergoing maintenance.

The next step is data processing. At some regular interval, approximately every 3 hours, Google Analytics processes the data in the logfile. Data processing time does fluctuate. Google Analytics does not process data in real time. While data is normally processed about every 3 hours, it's not normally complete until 24 hours after collection. The reason the data is not complete until 24 hours after processing is that the entire day's data is reprocessed after it has been collected.

3.2 Reports

When we log in to Google Analytics to view a report, Google Analytics creates that report in real time. Reports are created by comparing a dimension, like the Visitor City, to a numerical piece of information called a metric. Metrics include common web analytics numbers like visits, pageviews, bounce rate, conversion rate, revenue, etc.

When viewed alone, a metric provides a site-wide total for that metric. But when viewed compared to a dimension, the metric represents the total for that specific dimension. For example, a website may have a conversion rate of 2.87%. The metric in this case is conversion rate and the value is 2.87%. However, if we view conversion rate based on the City dimension, Google Analytics will display the conversion rate for each country in the database (see Figure-6).

Site Usage		Goal Set 1		Ecommerce	
Visits 700,323 % of Site Total: 100.00%		Goal1: All Purchases 1.76% Site Avg: 1.76% (0.00%)		Goal2: PWS Purchases 0.67% Site Avg: 0.67% (0.00%)	
		Goal Conversion Rate 2.43% Site Avg: 2.43% (0.00%)			
Detail Level: City		Visits ↓	All Purchases	PWS Purchases	
1.	Calgary	17,293	0.74%	0.23%	
2.	New York	16,410	1.65%	0.74%	
3.	Houston	14,936	2.00%	0.94%	
4.	San Francisco	12,830	2.01%	0.83%	
5.	Edmonton	11,725	0.72%	0.14%	
6.	Atlanta	9,623	2.38%	1.06%	
7.	Los Angeles	9,352	1.30%	0.45%	
8.	Dallas	8,836	2.25%	1.06%	
9.	Chicago	8,362	1.89%	0.88%	
10.	Denver	6,227	1.81%	0.82%	

Figure 6: This report shows the conversion rate (a metric) for the City dimension

Each row in Figure 3-2 is a different value for the City dimension. Notice that there are many columns, or metrics, in the report. Google Analytics can associate many different metrics for a single dimension.

Almost every report is created in the same manner. Google Analytics displays various metrics for a given dimension. If we are interested in a certain metric that Google Analytics does not include in a report, we can create a custom report to display that metric for the dimension.

3.3 About the Tracking

Google Analytics uses a very common web analytics technology called page tags to identify visitors, track their actions, and collect the data. Each page on our website that we want to track must be “tagged” with a small snippet of Java-Script. If the tracking code is not on a page, that page will not be tracked.

No tracking code, no data. It’s as simple as that. The following is the GATC JavaScript snippet:

```
<script type="text/javascript">
  var _gaq = _gaq || [];❶
  _gaq.push(['_setAccount', 'UA-XXXXXX-YY']);
  _gaq.push(['_trackPageview']);

  (function() {❷
    var ga = document.createElement('script'); ga.type = 'text/javascript';
    ga.async = true; ga.src = ('https:' == document.location.protocol ?
      'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0];
    s.parentNode.insertBefore(ga, s);
  })();
</script>
```

(1) This part of the tracking code does all of the work. It starts by creating a queue (named *_gaq*), or list of Google Analytics commands. These commands are also called methods. When we want Google Analytics to do something, we add, or push, a command into the queue.

The standard page tag adds the first two methods to the queue for us. The first method is *_setAccount()*. This method links the data collected from our site to our Google Analytics account using a unique number. The number is found directly after the "UA-". Once the *_setAccount()* method has been added, the tracking code knows where to send all of the data.

The second method, *_trackPageview()* is the part of the code that collects information about the visitor, stores it in cookies, and sends the data back to Google. This is the true workhorse of Google Analytics.

Remember, this part of the tracking code is a queue of methods. Up to this point, the only thing that has happened is two methods have been added to the queue. The queue has not been processed and no data has been sent to Google Analytics.

(2) Before any code can execute the `ga.js` file, which contains all of the Google Analytics logic, this code loads that logic into the browser and requests `ga.js` from a Google server. This file is geo-load-balanced across all of Google's global data centers. This means the visitor's browser will connect with the closest data center to reduce the time it takes to retrieve the file. The visitor's browser caches this file, so it's possible that it doesn't even need to load from a Google data center. Once the browser has retrieved the file, the queue of commands begins to execute. Notice the `ga.async=true` part of the code. This part of the code tells the browser to load the `ga.js` file asynchronously. This means that the browser can load `ga.js` while it continues to render the page for the visitor. Even if there is a communication issue between the browser and Google's Analytics servers and Google's servers stop transmitting `ga.js` to the browser, the browser will continue to render the page for the visitor.

3.4 Understanding Pageviews

The most important part of the GATC is `_trackPageview()`. This method collects visitor data, stores that data in cookies, and sends the data to the Google Analytics server. The `_trackPageview()` portion of the GATC is shown in bold in the following code:

```
<script type="text/javascript">

  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-XXXXXX-YY']);
  _gaq.push(['_trackPageview']);

  (function() {
    var ga = document.createElement('script'); ga.type = 'text/javascript';
    ga.async = true; ga.src = ('https:' == document.location.protocol ?
    'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0];
    s.parentNode.insertBefore(ga, s);
  })();

</script>
```

Every time `_trackPageview()` executes, a pageview is created by sending the data to Google Analytics (step 4 in Figure 5). Remember, each pageview includes many, many dimensions, like the visitor's IP address, city, country, region, etc. The actual page the visitor was looking at is captured in a

dimension called page. We can use the page dimension to create many of the reports in the Content section of Google Analytics. Figure 7 shows some sample data from the Top Content report.

Content Performance		
Pageviews ? 7,500 % of Site Total: 100.00%	Unique Pageviews ? 5,564 % of Site Total: 100.00%	Time on Page ? 00:02:23 Site Avg: 00:02:23 (0.00%)
URL	Pageviews ↓	
/blog/index.php	1,060	
/blog/2006/08/04/getting-more-out-of-google-analytics-goals/index.j	574	
/blog/2007/05/08/welcome-to-the-new-google-analytics/index.php	376	
/blog/2007/03/19/tracking-clicks-with-ga-pt-1-about-urchintracker/i	363	
/blog/2006/11/10/google-analytics-campaign-tracking-pt-1-link-tag	312	
/blog/2006/11/10/how-does-google-analytics-track-conversion-refe	294	

Figure 7: Top Content report showing pageview “names”

During the data collection process, `_trackPageview()` copies the information from the location bar of the visitor’s browser. It modifies the value by removing the domain name and domain extension. The only things left are the directories, filename, and query string parameters. This becomes the Request URI field or page dimension, and it is created during data processing. For example, the URL

`http://www.bracu.ac.bd/pages/index.php?id=110` will appear in the Top Content report as `/pages/index.php?id=110`. So, in this example, the request URI is the part of the URL that comes after `http://www.cutroni.com`.

3.5 Platform used for developing referral application

To develop this application mainly a open-source platform PHP-MySQL has been used. CanvasJS, a JavaScript graphics library is used too for visual presentation. To fetch the real-time Google Analytic data, a Google API has been used.

3.5.1 PHP, MySQL and CanvasJS

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

During the 2010s there have been increased efforts towards standardization and code sharing in PHP applications by projects such as PHP-FIG in the form of PSR-initiatives as well as Composer dependency manager and the Packagist repository. MySQL is an open-source relational database management system (RDBMS).[6] Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality. [4]

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube. [5]

CanvasJS Charts have a simple API and can render across devices including iPhone, iPad, Android, Windows Phone, Desktops, etc. This allows us to create rich dashboards that work across devices without compromising on maintainability or functionality of our web application. Graphs include several good looking themes and are 10x faster than conventional Flash / SVG based Charting Libraries – resulting in lightweight, beautiful and responsive dashboards. Checkout an overview of our JavaScript Charts. [10]

3.5.2 Google Analytic API

Google Analytics is implemented with "page tags", in this case, called the Google Analytics Tracking Code, which is a snippet of JavaScript code that the website owner adds to every page of the website. The tracking code runs in the client browser when the client browses the page (if JavaScript is enabled in the browser) and collects visitor data and sends it to a Google data collection server as part of a request for a web beacon.

The tracking code loads a larger JavaScript file from the Google web server and then sets variables with the user's account number. The larger file (currently known as ga.js) is typically 18 KB. The file does not usually have to be loaded, however, due to browser caching. Assuming caching is enabled in the browser, it downloads ga.js only once at the start of the visit. Furthermore, as all websites that implement Google Analytics with the ga.js code use the same master file from Google, a browser that has previously visited any other website running Google Analytics will already have the file cached on their machine.

In addition to transmitting information to a Google server, the tracking code sets a first party cookie (If cookies are enabled in the browser) on each visitor's computer. This cookie stores anonymous information, called the *ClientId*. Before the launch of Universal Analytics, there were several cookies storing information such as whether the visitor had been to the site before (new or returning visitor), the timestamp of the current visit, and the referrer site or campaign that directed the visitor to the page (e.g., search engine, keywords, banner, or email).

If the visitor arrived at the site by clicking on a link tagged with Urchin Traffic Monitor (UTM) codes such as:
`http://toWebsite.com?utm_source=fromWebsite&utm_medium=bannerAd&utm_campaign=fundraiser2012`, then the tag values are passed to the database too.[1]

Chapter 4

Implementation

4.1 Google Analytic API

Platform components [7]

Developers interact and influence processing through a rich user interface, client libraries, and APIs that are organized into 4 main components: collection, configuration, processing, and reporting.

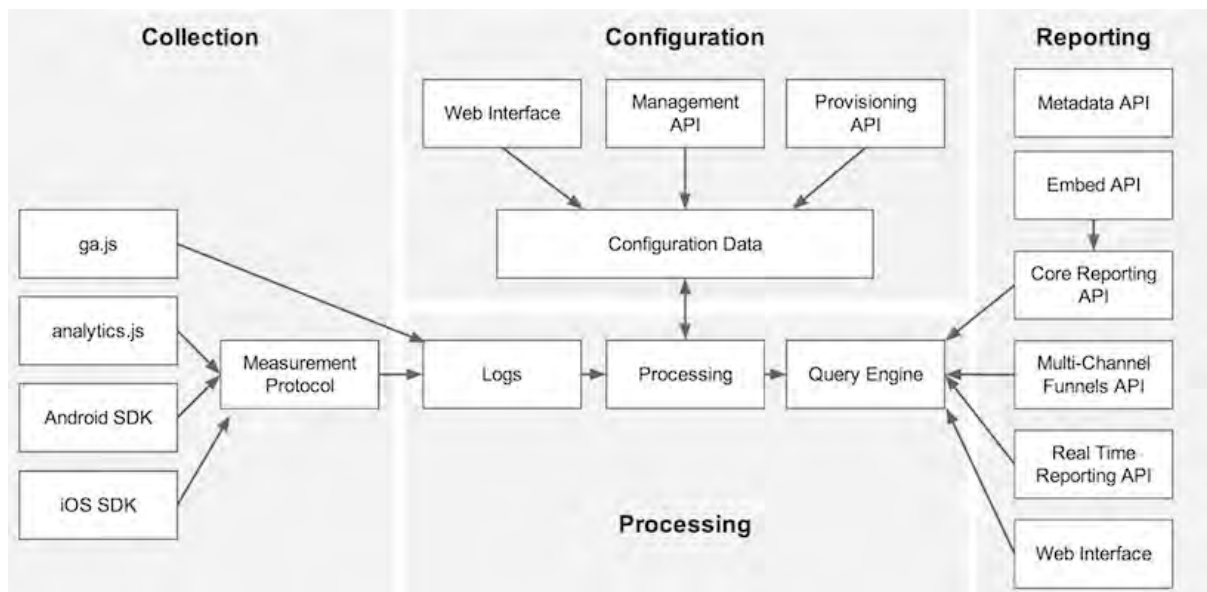


Figure 8: Components of Google Analytic

1. **Collection:** collects user-interaction data.
2. **Configuration:** allows we to manage how the data is processed.
3. **Processing:** processes the user-interaction data, with the configuration data.
4. **Reporting:** provides access to all the processed data.

The analytics.js library is a JavaScript library for measuring how users interact with our website. This document explains how to add analytics.js to our site.

The JavaScript tracking snippet [6]

Adding the following code (known as the "JavaScript tracking snippet") to our site's templates is the easiest way to get started using analytics.js.

The code should be added before the closing `</head>` tag, and the string `'UA-XXXXX-Y'` should be replaced with the property ID (also called the "tracking ID") of the Google Analytics property we wish to track.

```
<!-- Google Analytics -->
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');

ga('create', 'UA-XXXXX-Y', 'auto');
ga('send', 'pageview');
</script>
<!-- End Google Analytics -->
```

The above code does four main things:

1. Creates a `<script>` element that starts asynchronously downloading the *analytics.js* JavaScript library from *https://www.google-analytics.com/analytics.js*
2. Initializes a global *ga* function (called the *ga()* command queue) that allows we to schedule commands to be run once the *analytics.js* library is loaded and ready to go.
3. Adds a command to the *ga()* command queue to create a new tracker object for the property specified via the `'UA-XXXXX-Y'` parameter.
4. Adds another command to the *ga()* command queue to send a pageview to Google Analytics for the current page.

Custom implementations may require modifying the last two lines of the JavaScript tracking snippet (the create and send commands) or adding additional code to track more interactions. However, we should not change the code that loads the *analytics.js* library or initializes the *ga()* command queue function.

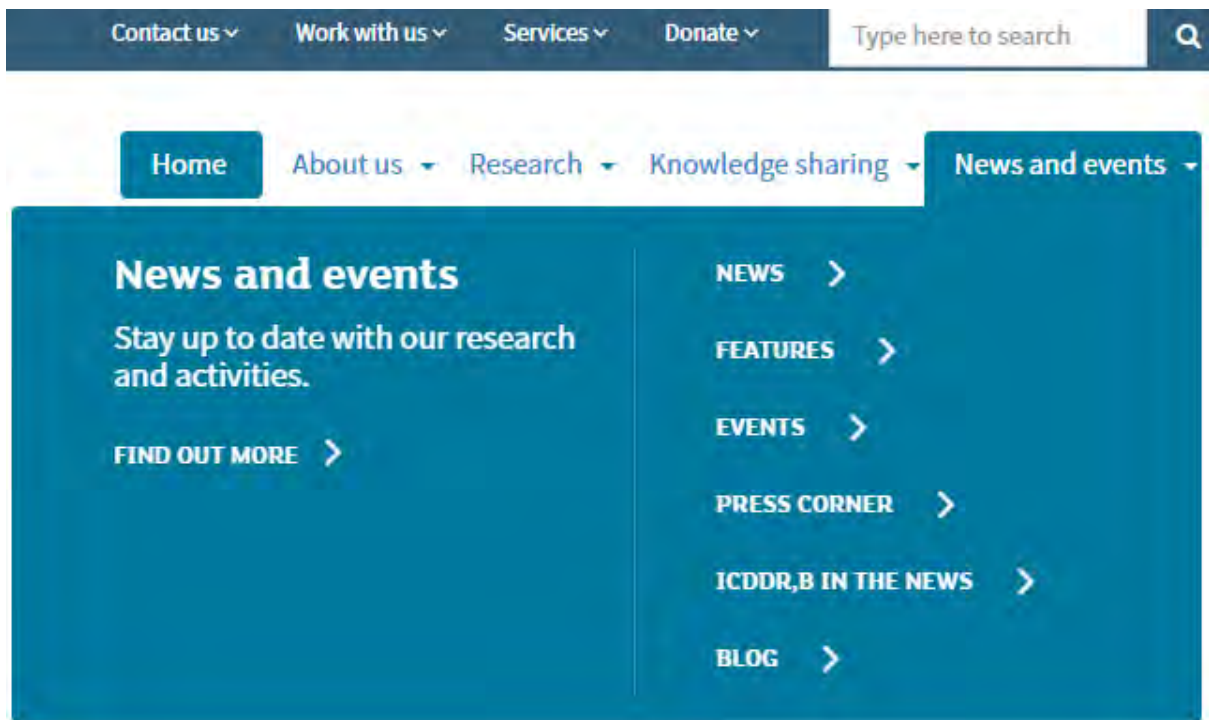
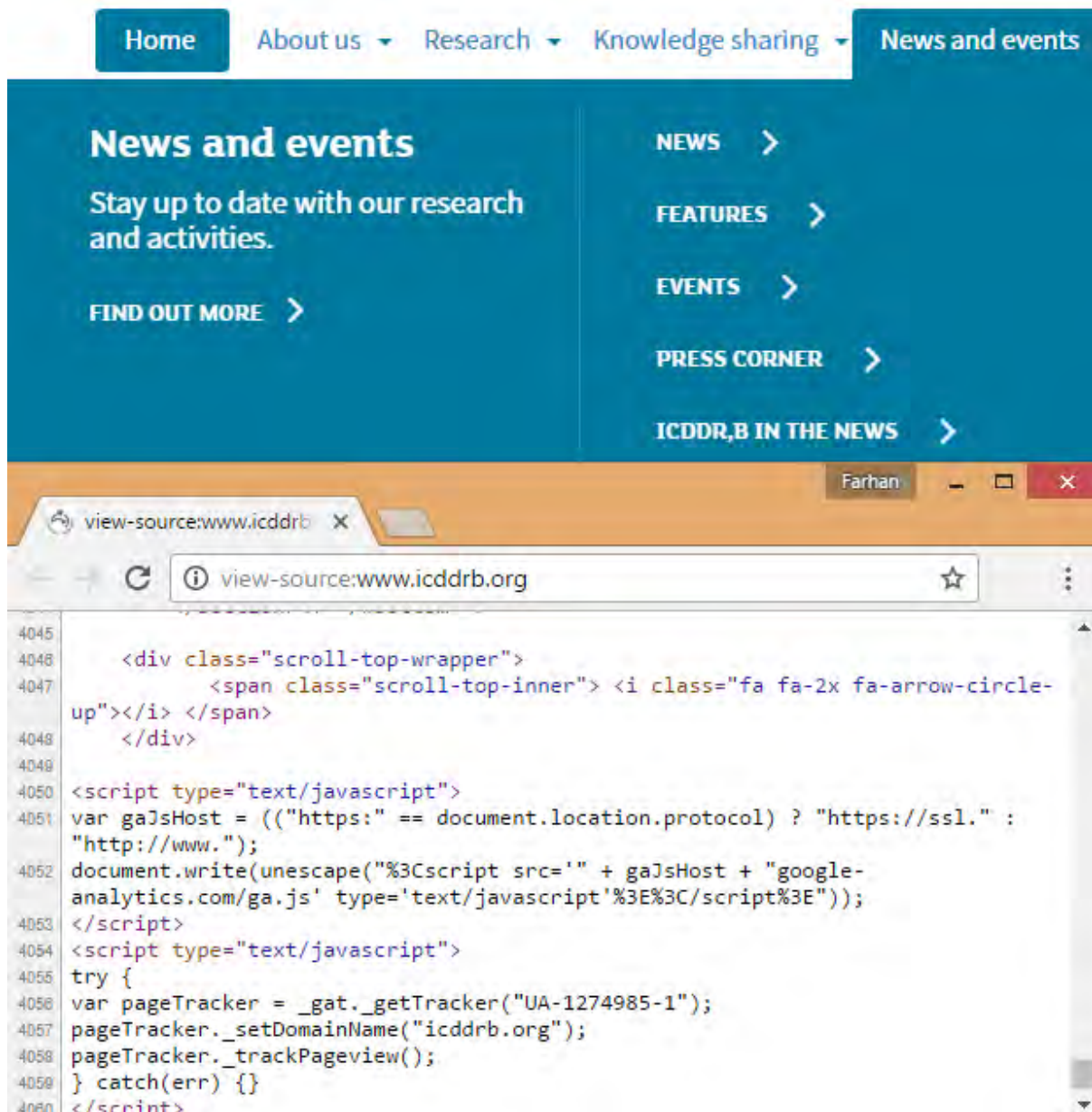


Figure 9: One of the web page where to put tracking code

As an example, the above figure contains a real web page, where to set the analytical tracking code written in JS, collected from Google Analytic.



The above figure contains the HTML source where we found that JS tracking code, where we found the tracker code as *UA-1274985-1*, as well the registered domain name.

What data does the tracking snippet capture? [6]

When we add either of these tracking snippets to our website, we send a page-view for each page our users visit. Google Analytics processes this data and can infer a great deal of information including:

1. The total time a user spends on our site.
2. The time a user spends on each page and in what order those pages were visited.
3. What internal links were clicked (based on the URL of the next pageview).

In addition, the IP address, user agent string, and initial page inspection analytics.js does when creating a new tracker is used to determine things like the following:

1. The geographic location of the user.
2. What browser and operating system are being used.
3. Screen size and whether Flash or Java is installed.
4. The referring site.

How to enable Google API to fetch data? [8]

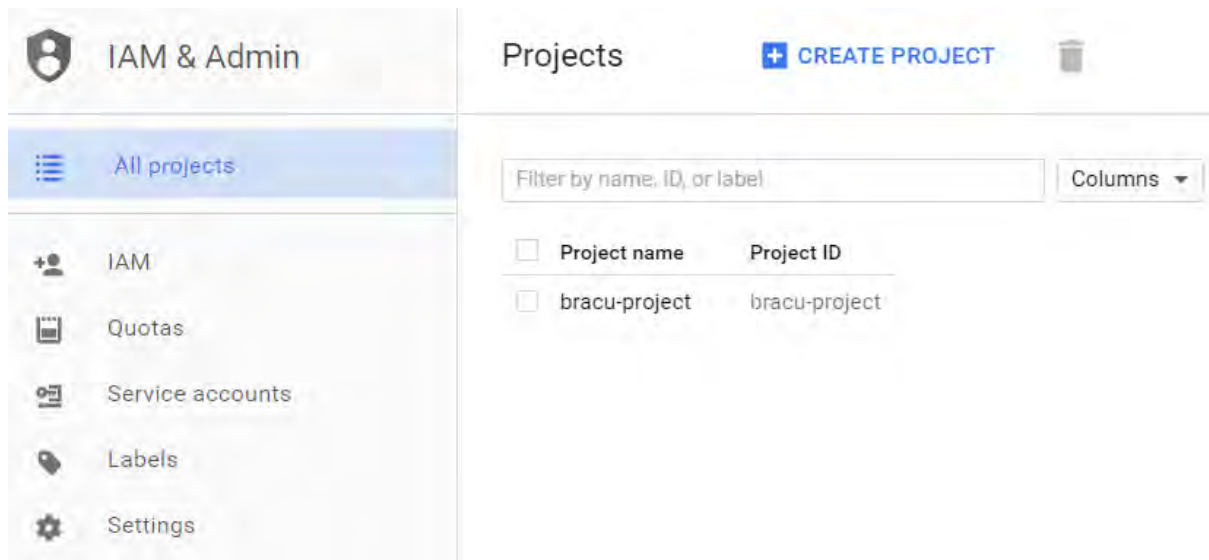


Figure 11: Creation of API file on Google Analytic

Here, we created a OAuth 2.0 client ID and Service Account first, then download the Service Account in a file format by p12 to use that via GAPI to access analytical data.

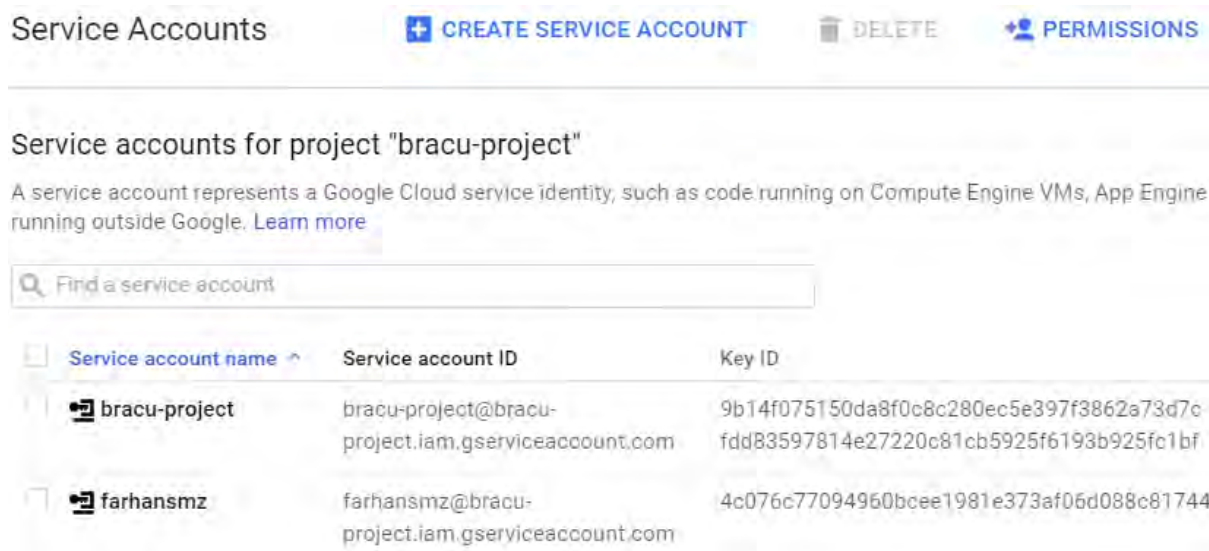


Figure 12: List of available service accounts

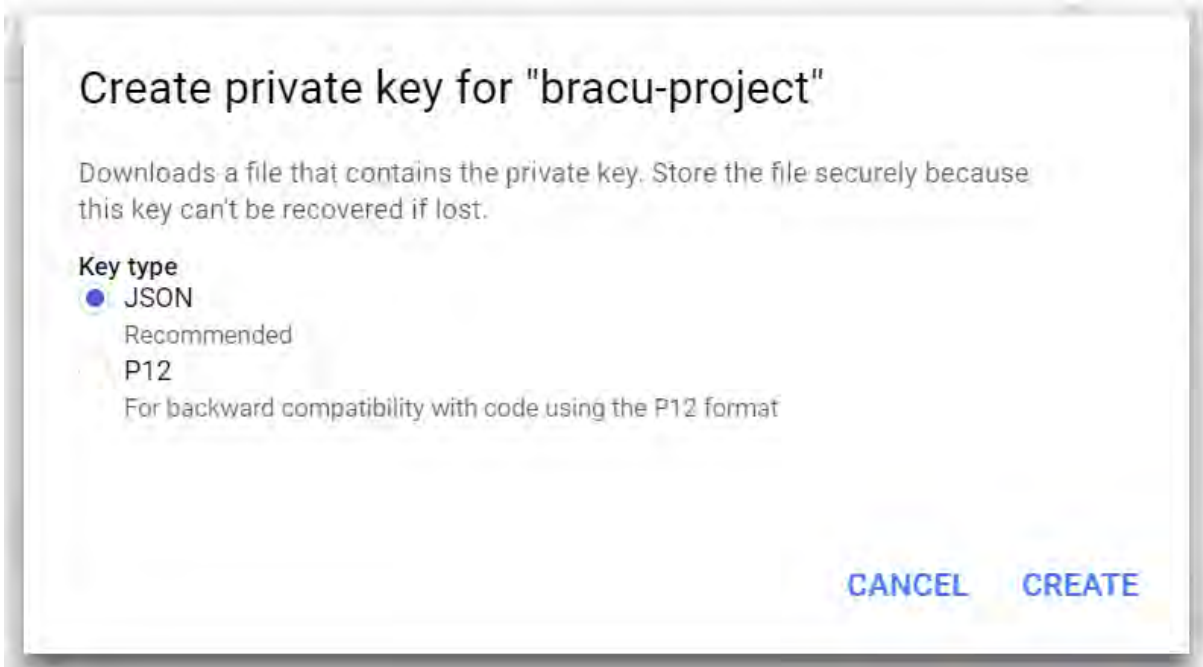


Figure 13: Download the API file in p12 format

4.2 Use of GAPI class

GAPI (Google Analytics PHP Interface) is a simple yet powerful Google Analytics library written in PHP to fetch data from Google Analytics. The features are as follows: [9]

1. Supports cURL[12] and fopen HTTP access methods, with auto-detection
2. PHP arrays for Google Analytics metrics and dimensions
3. Account data object mapping - get methods for parameters
4. Report data object mapping - get methods for metrics and parameters
5. Easy filtering, use a GAPI query language for Google Analytics filters
6. Full PHP5 Object Oriented code, ready for use in our PHP application

It has a configuration file, where we have to put following information establish the connection to pool desired data:

Service Account	icddrb@bracu-project.iam.gserviceaccount.com
p12file	bracu-project.p12
profileID	2264928

The proper process diagram has been shown below.

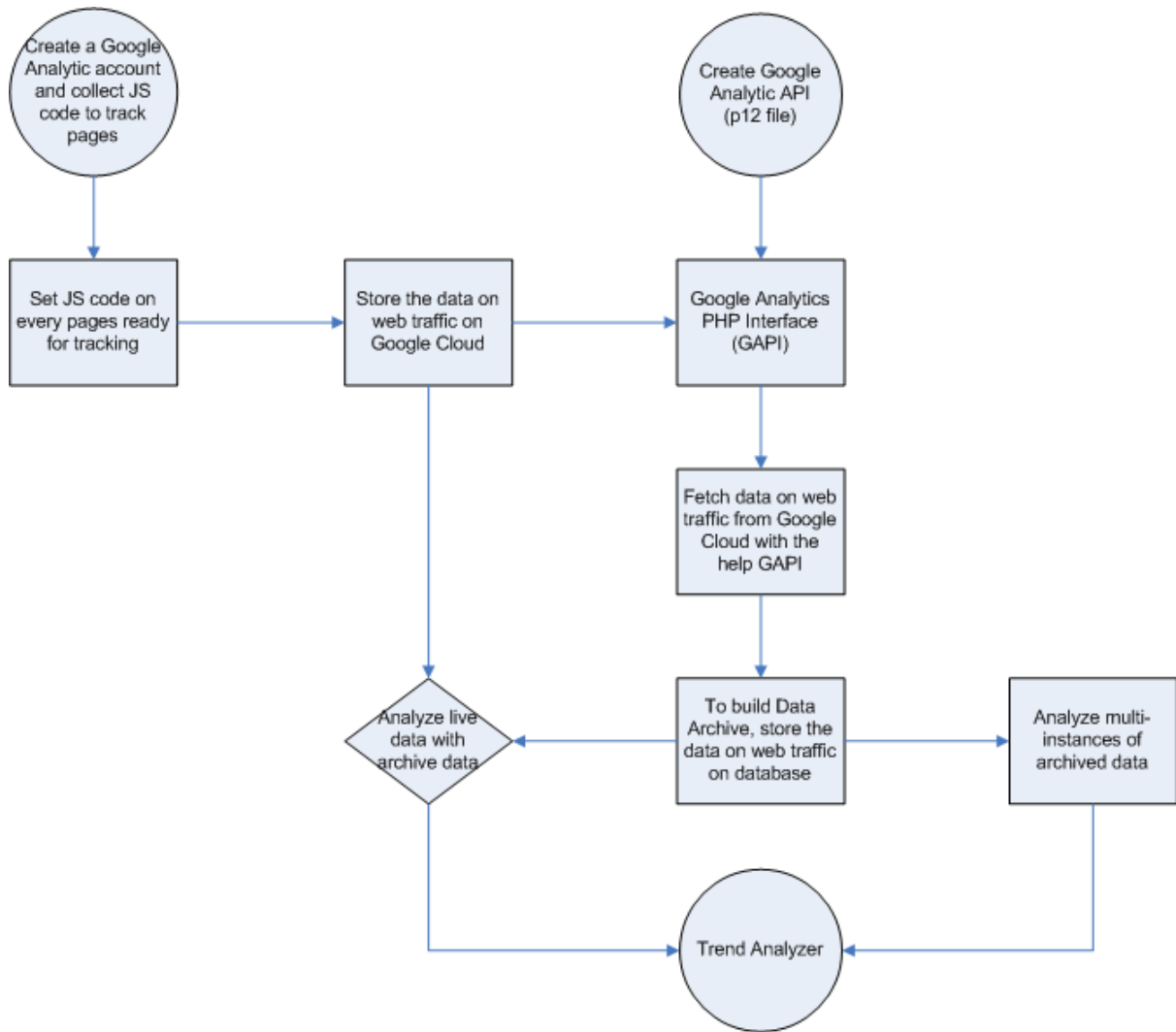


Figure 14: Activity diagram for Google Analytic Project

4.3 User Interface

4.3.1 API Information

API Information

Through Google API, the project is connecting with the Web Analytic Data of target websites.

ACCOUNT NAME:	ANALYTIC ID:	PROFILE ID:
e-procurement	UA-1274985-6	54198541
http://centre.icddrb.org	UA-1274985-3	46375203
http://dspace.icddrb.org	UA-1274985-5	48105385
http://www.icddrb.net.bd	UA-1274985-4	47890944
http://www.icddrb.org	UA-1274985-1	2264928
http://www.jhpn.net	UA-1274985-2	46373543
LMS	UA-1274985-7	99523193

Figure 15: Analytic Account Information are being collected through API

Through the Google Analytic API connection, the project is being connected with the various Google analytical profile. The Analytic ID is used to generate the connection with analytical data storage to store the data, and Profile ID is used to fetch data from storage.

4.3.2 Analytical Data

Analytical Data

Category	Page Views
News	13563
Features	1980
Press Releases	590
Media resources	397

Graphical Presentation

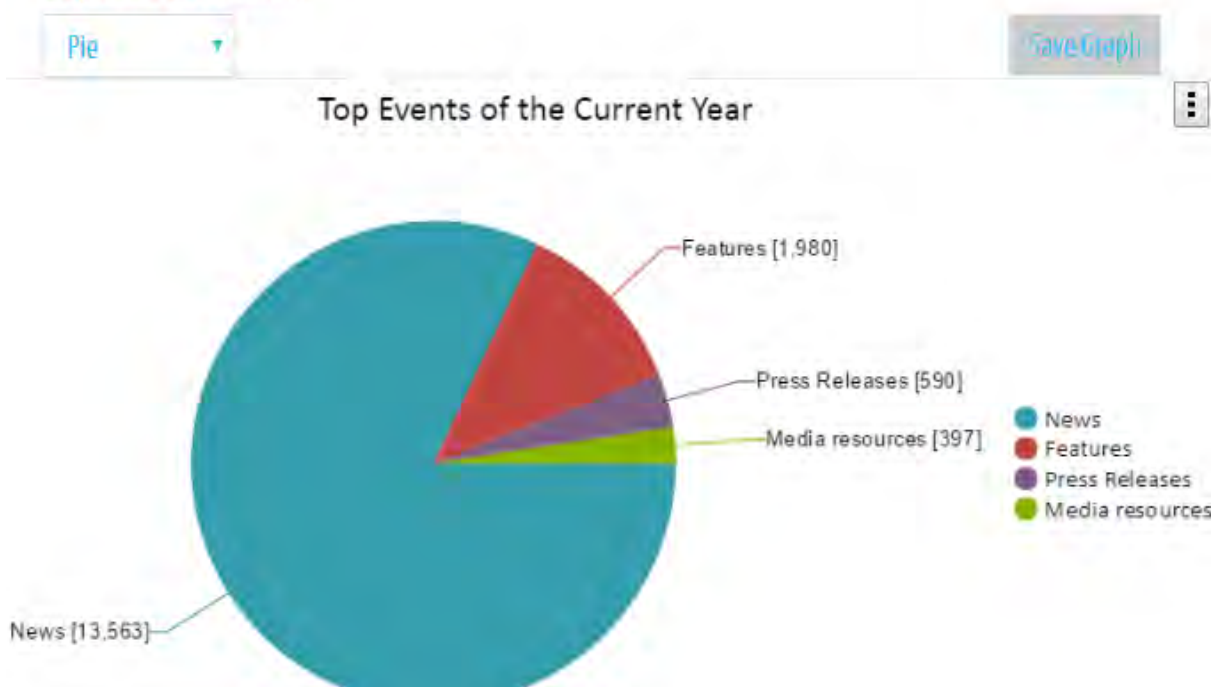


Figure 16: Analytical Data

Here, we studied real-time Google Analytic data for some pre-defined pages, and visualize that data in any types of charts as:

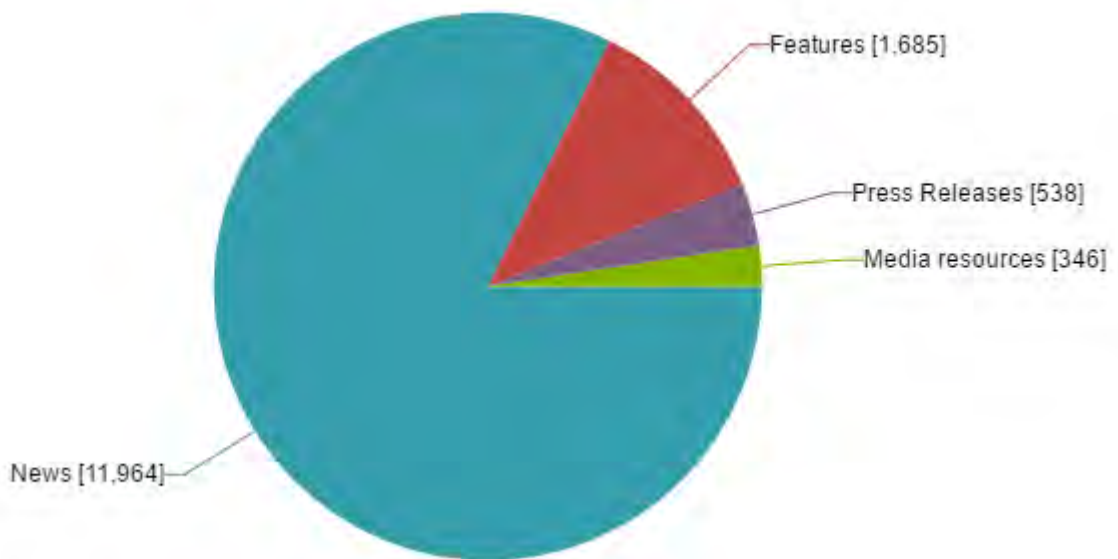


Figure 17A: Data Visualization (Pie chart)

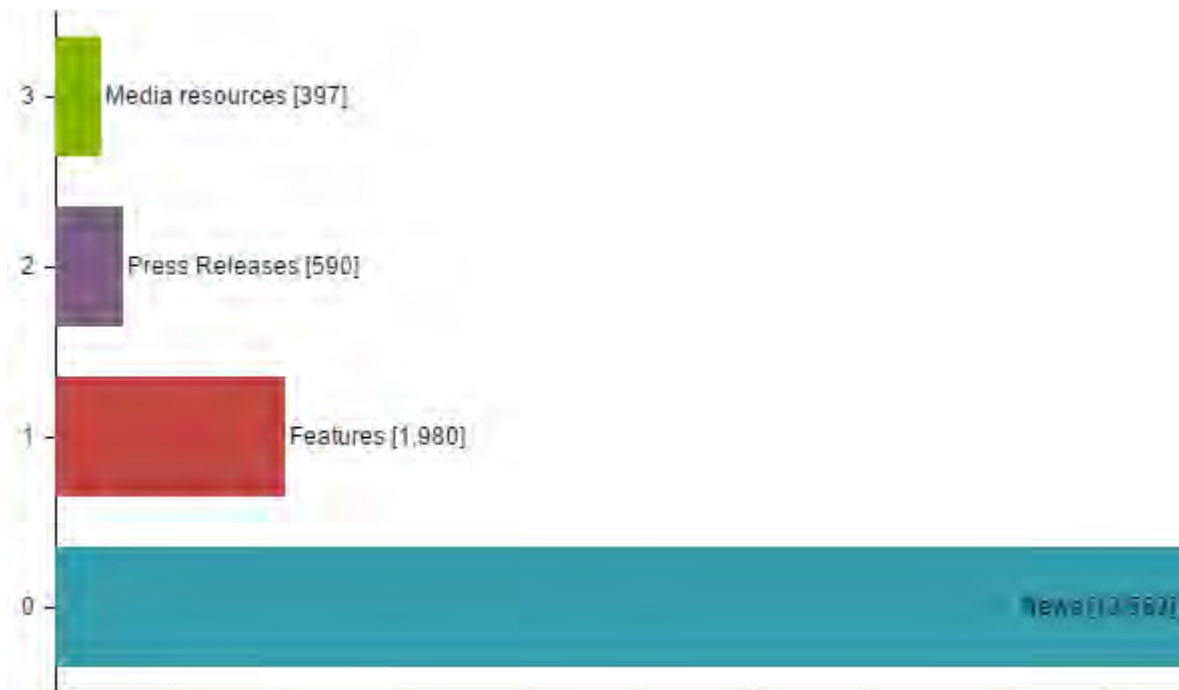


Figure 17B: Data Visualization (Bar chart)

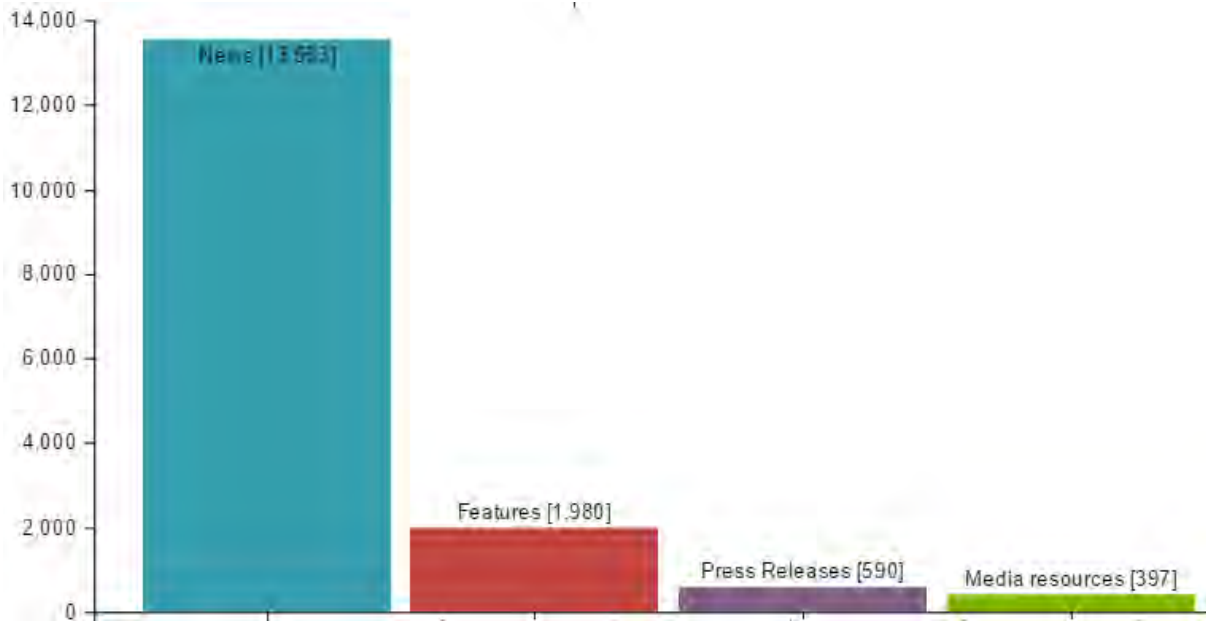


Figure 17C: Data Visualization (Column chart)

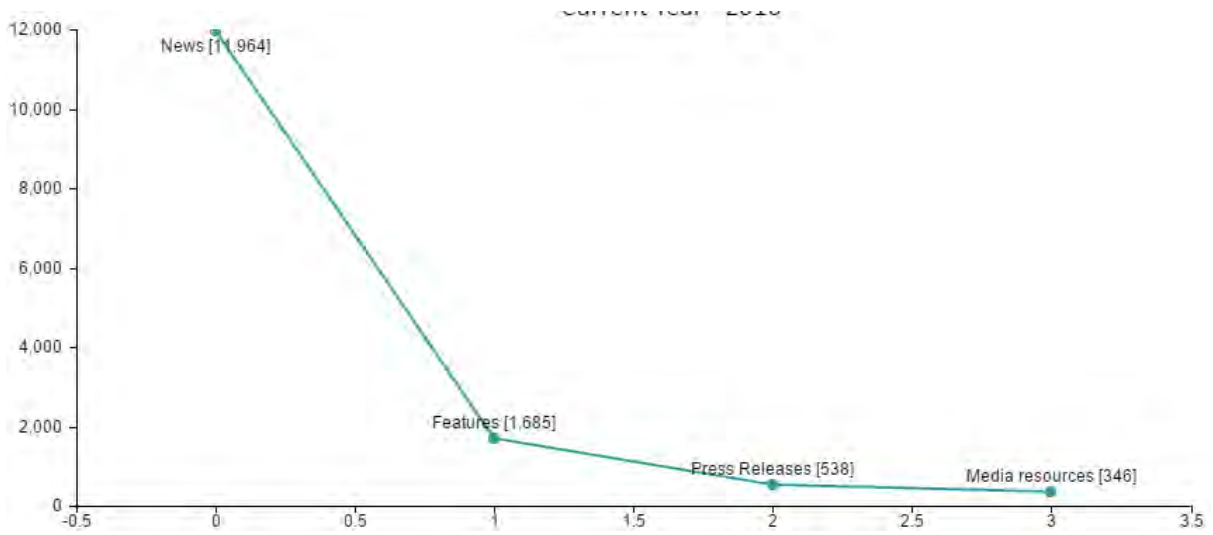


Figure 17D: Data Visualization (Line chart)

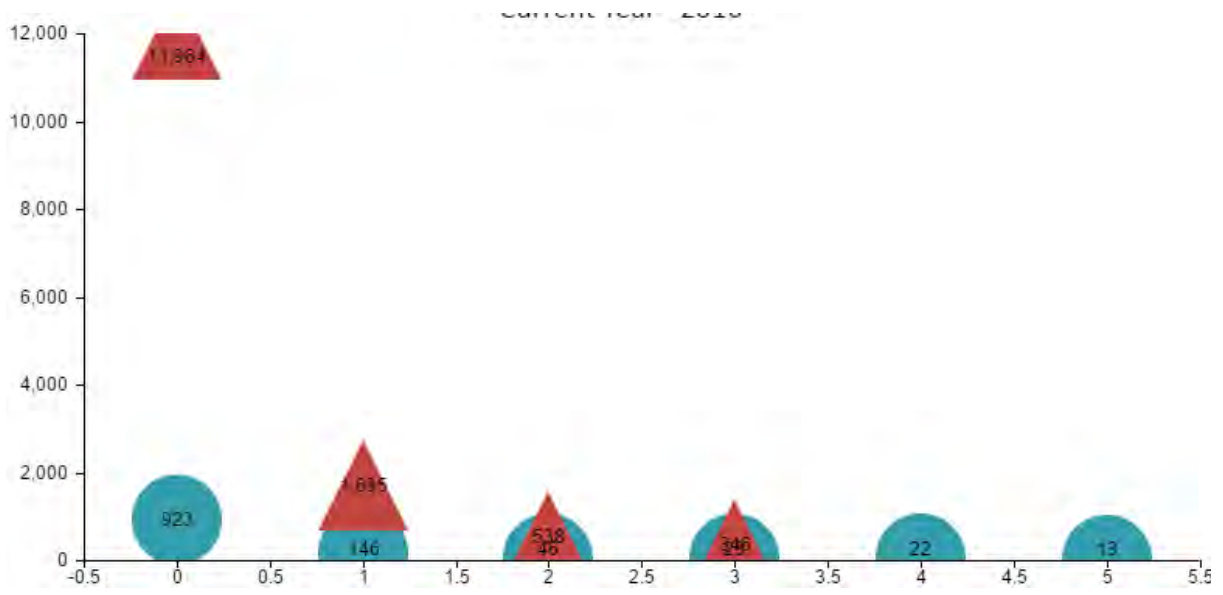


Figure 17E: Data Visualization (Bubble chart)

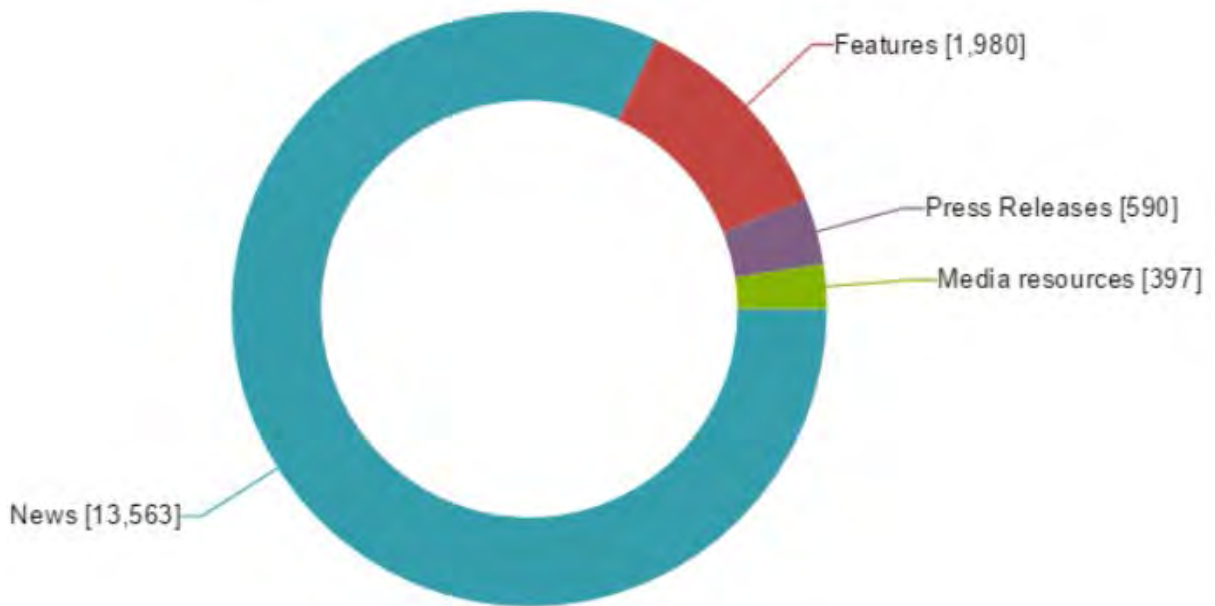


Figure 17F: Data Visualization (Doughnut chart)

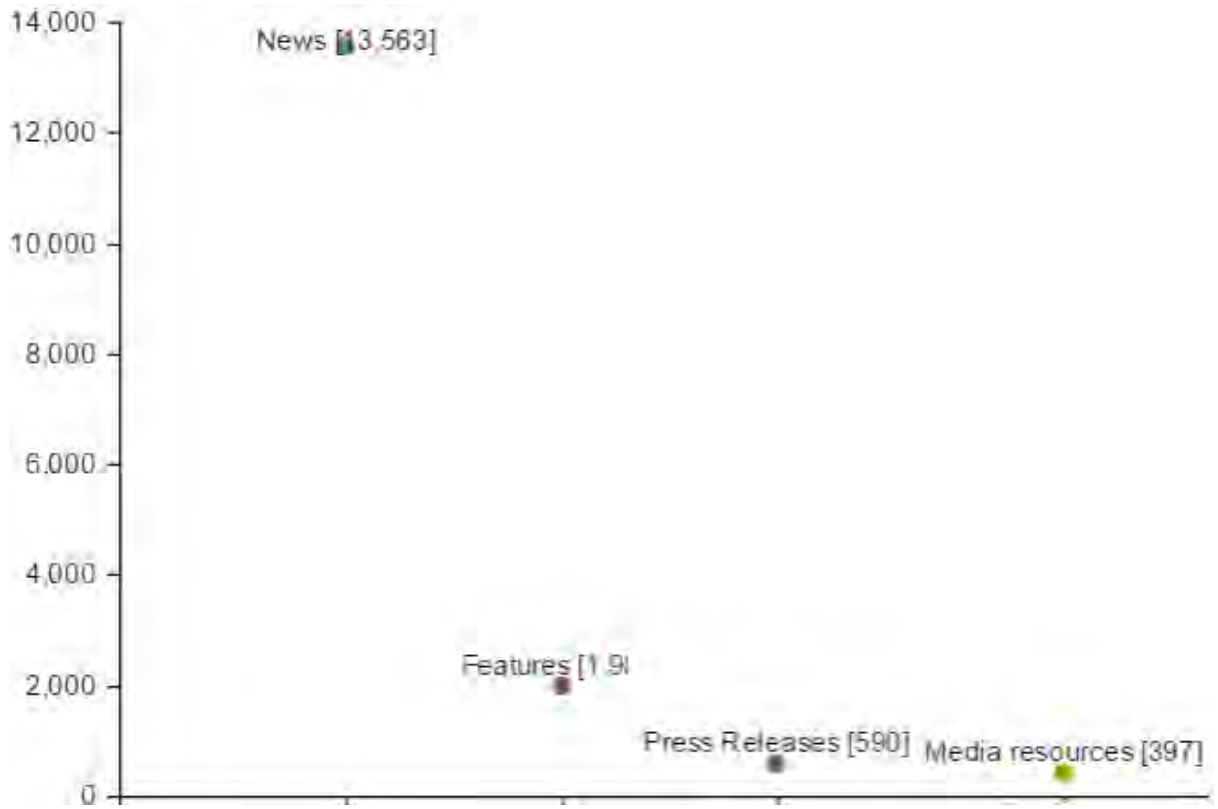


Figure 17G: Data Visualization (Scatter chart)

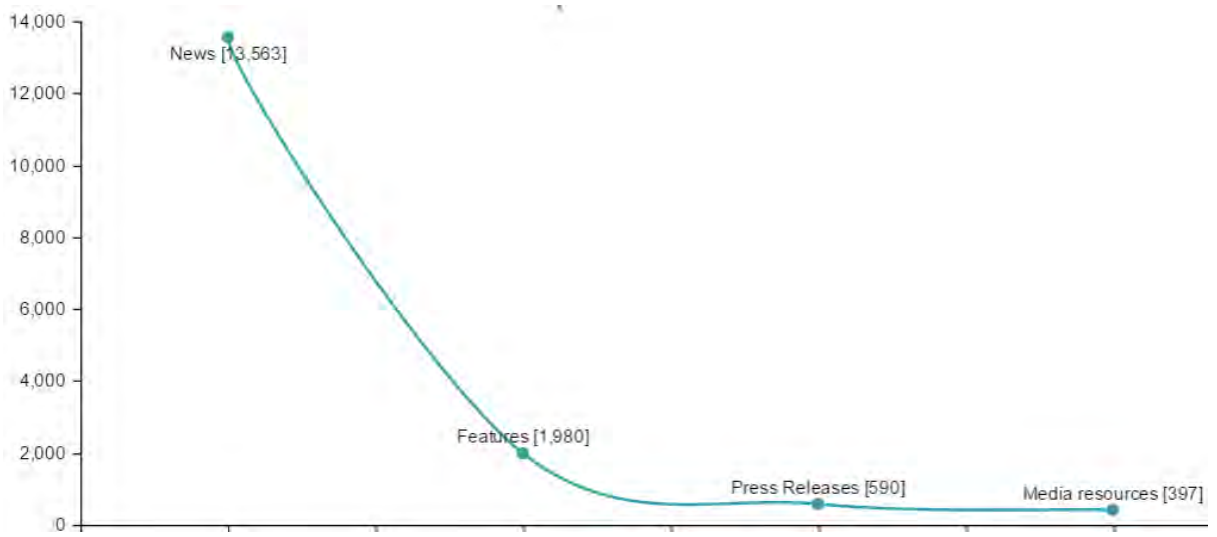


Figure 17H: Data Visualization (Spline chart)

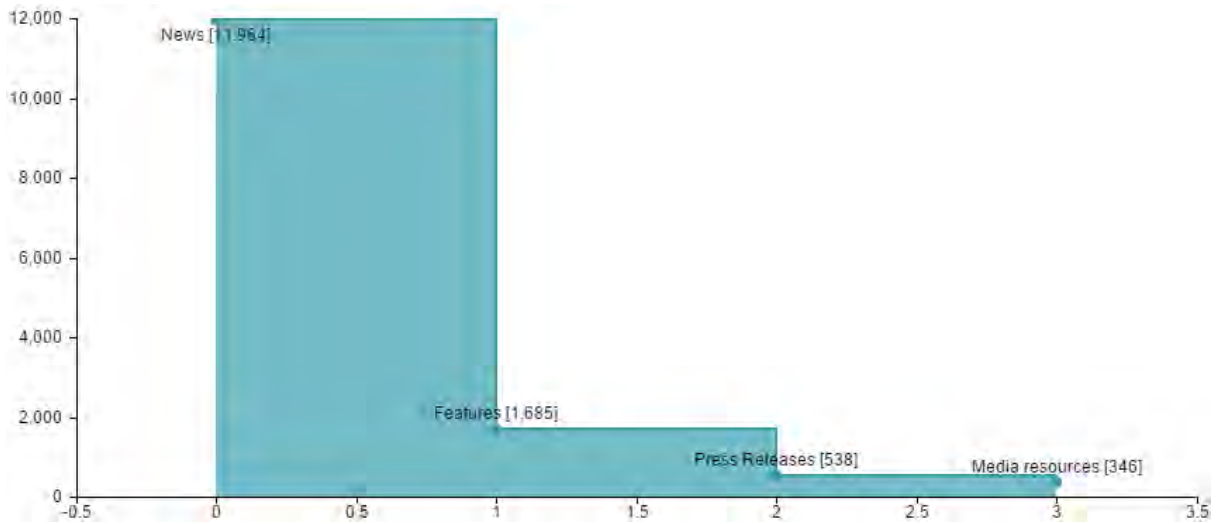


Figure 17I: Data Visualization (Step Area chart)

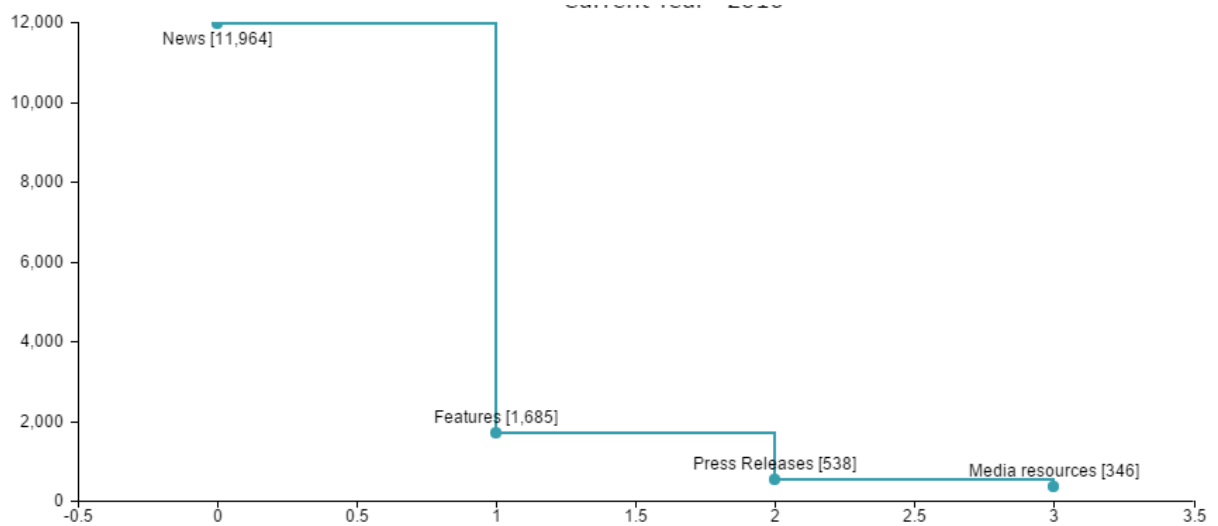


Figure 17J: Data Visualization (Step Line chart)

We also archive that data of any moment to future uses and visualization in any types of chart type.

4.3.3 Data Archive

The main objective to Data Archive to visualize the archived data in desired format to analyze.

News n Events [1] 2016-01-31 23:50:00

Title	Page Views
News	923
Features	146
Events	46
Press Releases	23
Media resources	22
In the News	13

Graphical Presentation

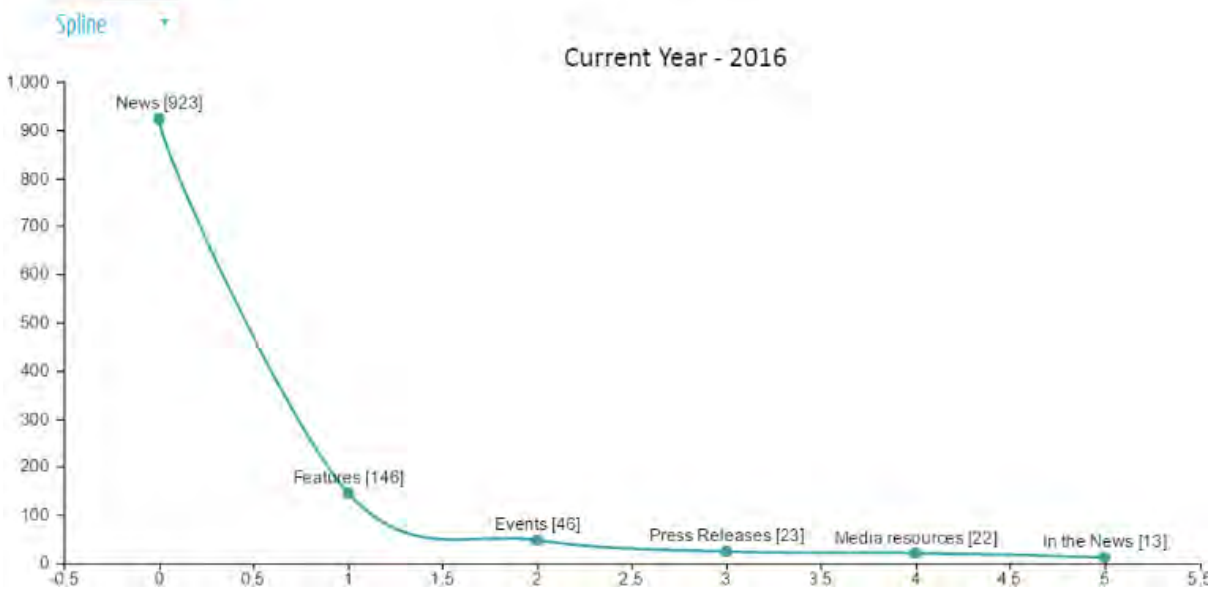


Figure 18: Visual representation of Analytical data (Spline chart)

4.3.4 Trend Analysis

There are three types of trend analysis, we're incorporating here as:

1. Analysis live data with the any instant of data archived
2. Analysis any two instances of archived data
3. Quarterly New Analysis

Live Data

-- All Types --

Title	Page Views
News	13563
Features	1980
Press Releases	590
Media resources	397

Saved Data

News n Events [19] 2016-11-13 20:53:23

Title	Page Views
News	11964
Features	1685
Press Releases	538
Media resources	346

Graphical Comparison

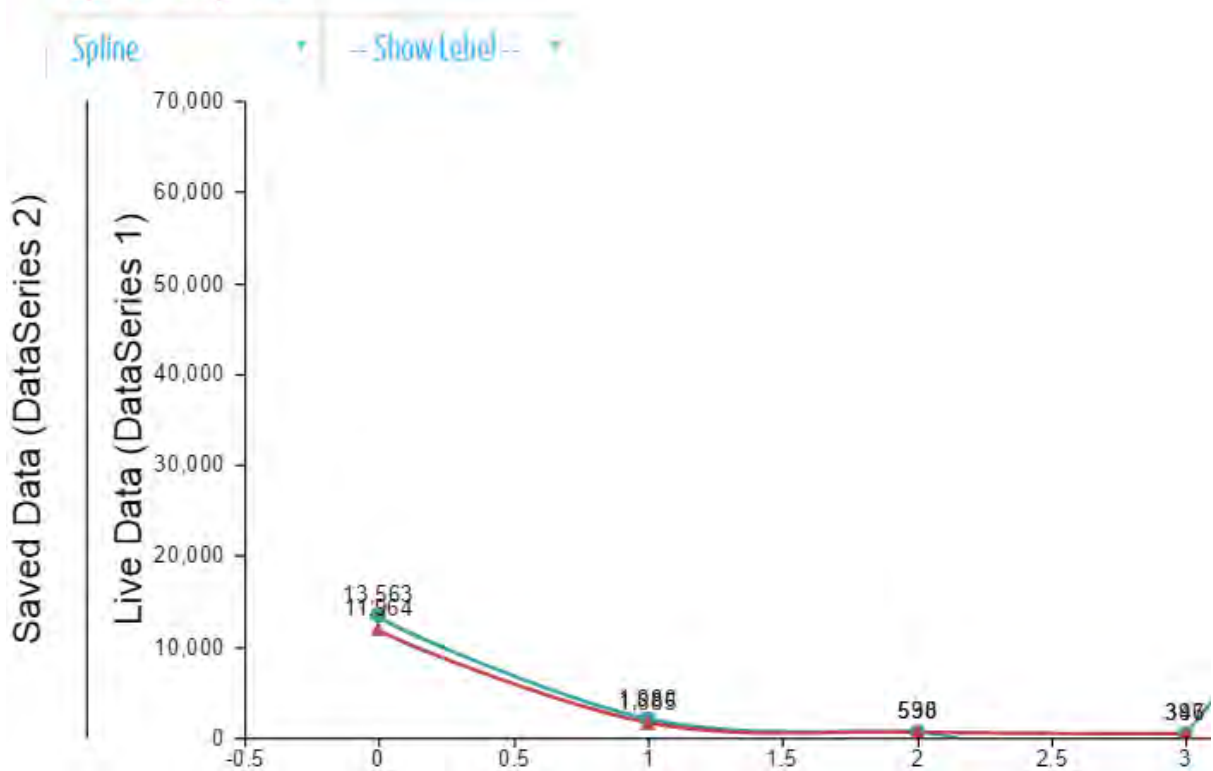


Figure 19: Trend Analysis between real-time data and archive data

In analysis live data with the any instant of data archived, we analyze the visual representation of any archived instance with live scenario to find out significant deviation on page views over time.

Saved Data # A

News n Events [1] 2016-01-31 23:50:00

Title	Page Views
News	923
Features	146
Press Releases	23
Media resources	22

Saved Data # B

News n Events [19] 2016-11-13 20:53:23

Title	Page Views
News	11964
Features	1685
Press Releases	538
Media resources	346

Graphical Comparison

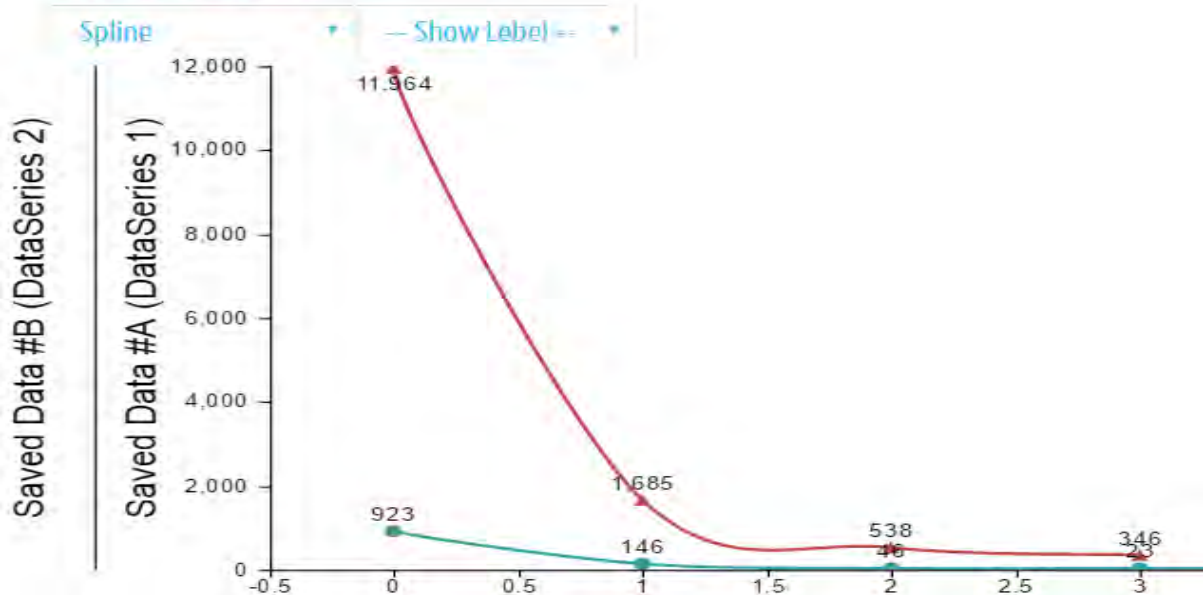


Figure 20: Trend Analysis between archive data

In analysis any two instances of archived data, we analyze the visual representation of any two archived instances to find out significant deviation on page views over time.

List of News - 2016

Title	Page Views
icddr;b wins major award to help prevent Hepatitis E in pregnancy	197
New guidelines to improve neonatal care for premature babies in Bangladesh	142
Patients with diabetes should be screened for depression to lower threats from both	39
Low-cost thermal blanket could save the lives of preterm babies	100
icddr;bs executive director joins Ebola Vaccine Team B to improve response to infectious disease threats	50
Targeting the families of cholera patients could have a major impact on the reduction of disease	157
icddr;b researchers attempt to determine populations most at risk from Zika virus	58
Hepatitis E - a preventable cause of one in five maternal deaths in Bangladesh	385
icddr;b researchers united to end TB	243

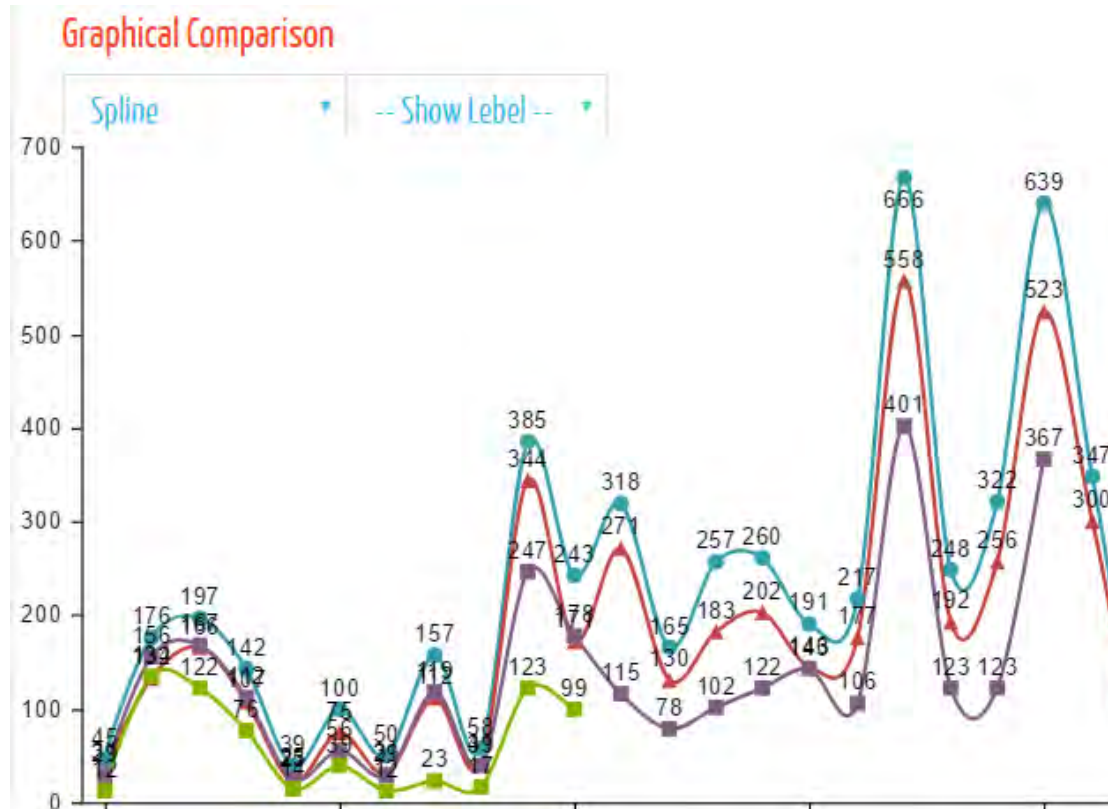


Figure 21: Trend Analysis - Quarterly News Analysis

In Quarterly New Analysis, as it is so important to analyze the page views of all the news contents, we're archiving quarterly instances of all the news for current year, and generate a visual presentation of that data in a single chart to point out the significant deviation in every quarters.

4.4 IKR (Institutional Knowledge Repository)

The organizational publications are available for free online, from annual reports, to journal articles, policy briefs and our health science bulletin. Browse by community, collection, author or publication date, or use the search function to find a specific article. Here, there is a demand to justify the communities any time based on their popularity per communities.

Communities

Shown below is a list of communities and sub-communities within them. Click on a name to view that community home page.

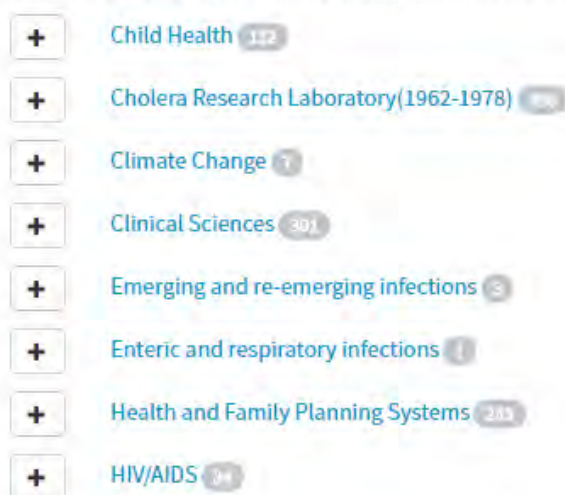


Figure 22: Communities in IKR

IKR Communities

Communities	Quantities	Page Views
Child Health	132	1466
Cholera Research Laboratory(1962-1978)	356	40
Climate Change	7	10
Clinical Sciences	301	1951
Emerging and re-emerging infections	3	6
Enteric and respiratory infections	1	2
Health and Family Planning Systems	283	18
HIV/AIDS	34	53

Figure 23: Page views and quantities per community to analysis

So, based on popularity (views) over quantities of publications, a visualization has been implanted as percentage of popularity as following figure.

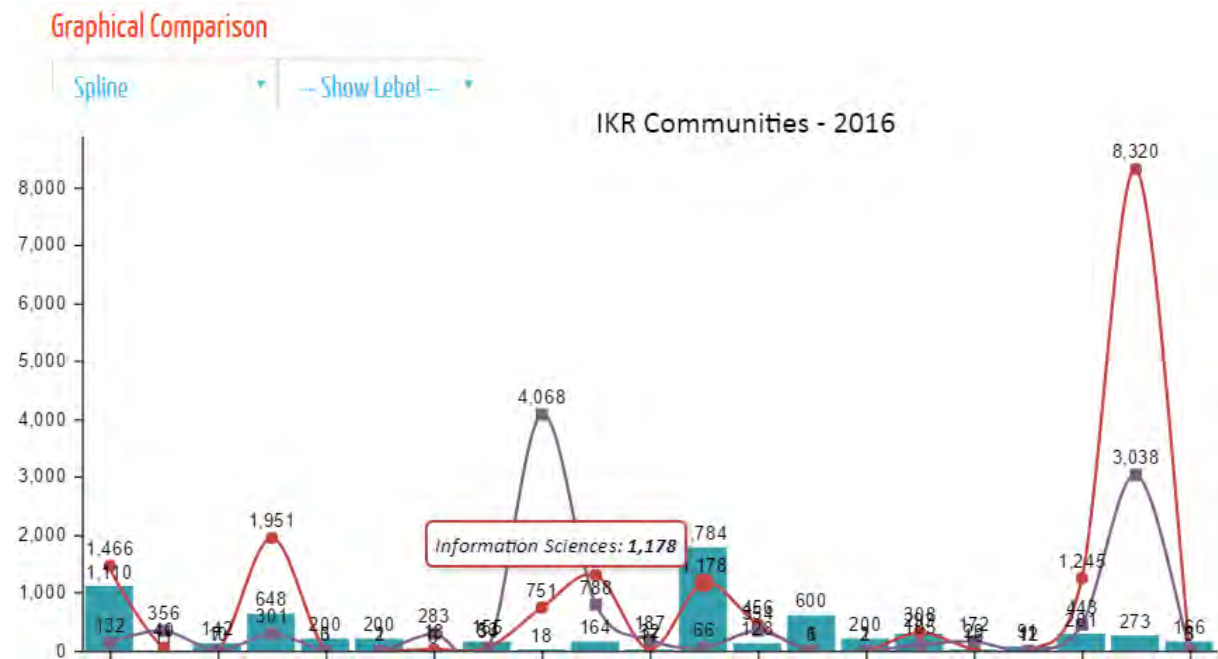


Figure 24 : Analyzed data to compare among the popularity of all communities, showing the items view in every communities, number of total items per communities and the percentage of popularity.

Chapter 5

Conclusion & Future Work

In conclusion, the work and contribution of this paper is summarized below. A literature review was conducted that revealed a need for applying modern technologies to practice on trend analyzing with web analytical data. Furthermore a literature review was also conducted to identify what are the major software engineering challenges to develop a analytical software.

A lot of content are being covered in this report - tracking across multiple domains to filters, I'm tracking to explain how the Google Analytics system works and the pitfalls to avoid. The goal is to help we get Google Analytics set up correctly for web site and business needs.

On IKR, we're analyzing more trends among collections for every collections, to take decision as same as communities too.

Yet a prototype have been proposed in this report, a major work need to be It is very much necessary that in future an evaluation is conducted to find out potential shortcomings of the application from the performance perspectives.

References

1. Google Analytics, https://en.wikipedia.org/wiki/Google_Analytics, Accessed on November 15, 2016
2. Cutroni, J. (2010) Google Analytics.
3. APIs for reporting and configuration, <https://developers.google.com/analytics/>, Accessed on November 20, 2016
4. PHP, <https://en.wikipedia.org/wiki/PHP>, Accessed on November 20, 2016
5. MySQL, <https://en.wikipedia.org/wiki/MySQL>, Accessed on November 20, 2016
6. Dibrova, A. (2013) Web analytics. Website analysis with Google Analytics and Yandex Metrics
7. Adding analytics.js to Our Site, <https://developers.google.com/analytics/devguides/collection/analyticsjs/>, Accessed on November 20, 2016
8. Learn about Google Analytics, <https://developers.google.com/analytics/devguides/platform/>, Accessed on November 20, 2016
9. Google API Client Libraries PHP (Beta), https://developers.google.com/api-client-library/php/guide/aaa_overview, Accessed on November 21, 2016
10. Google Analytics PHP Interface, <https://github.com/erebusnz/gapi-google-analytics-php-interface>, Accessed on November 21, 2016
11. HTML5 JavaScript Charts, <http://canvasjs.com>, Accessed on November 21, 2016
12. MeSH, <https://www.ncbi.nlm.nih.gov/mesh>, Accessed on November 22, 2016
13. cURL Requests with PHP, <http://codular.com/curl-with-php>, Accessed on November 23, 2016