Social Networking and YouTube

Chapter-I

**Introduction:**

YouTube is the most popular video sharing website on the internet today. It’s the biggest user driven video content provider in the world (Wattenhofer, M. 2012). For this research I chose YouTube as it is one of the biggest and most popular Social Networks. “500 Years of YouTube video are watched on Facebook every day, and over 700 YouTube videos are shared on Twitter each minute” (YouTube Statistics\*). The main aim of this project is to determine the role other social networks like Facebook and Twitter have in the popularity of the videos on YouTube which become viral. Facebook allows users to share the video links on their Facebook home page called wall or post a link on other user’s wall. Those videos can be viewed within the Facebook page where the video plays in a player embedded within the page and can be shared using the “Share” option under the video and the video becomes available on the user`s home page wall where his friends can view and share it.

During this study it was determined where these videos get their maximum views, and the effect on their popularity. YouTube provides the statistical information where we can view the video link has been visited from which includes referral from other videos, post on other social networks, blogs and sites, view from subscriber module and view from mobile devices. A crawler was specifically designed to gather the required information from YouTube videos which had more than a hundred thousand views. The aim behind the analysis of the data was to:

1. Get the information about the videos that were embedded and viewed from Facebook. What relationship can we establish with Facebook and those videos?
2. Find where these videos get their maximum views from? Are there more popular means of sharing the videos?
3. What other things can we determine from that data?

This will help us determine the role Facebook plays is in the distribution of viral videos via YouTube and which “Category” they belong to.

This project revolves around studying the aspects of YouTube which have not been examined in other YouTube researches however it analyses a small dataset compared to them.

**About other Social Networks and Facebook**

Social Networking has always been an important topic for research and has gained much popularity in the last decade. Today a large amount of internet population is based on file sharing and social networking. The purpose of online social networks is to allow users to access each other`s information as long as they have a friend relationship with each other. Almost every person who accesses the internet uses social network to connect with others. These networks are as old as the internet itself, for example the graph formed by a simple exchange of email message or a chat message among users forms an online social network (Mislove et al 2007).

The most popular social networks today are MySpace, Facebook, Twitter, LinkedIn, Google+, and Orkut etc. YouTube is a video sharing website where users can upload and view videos. Unlike other conventional social networks YouTube is primarily based on sharing videos, but it also provides features like messaging, subscribing, adding friends, sharing and adding comments. We will discuss about the details later in this paper.

Many social networking websites like Facebook allow users to hide information from unknown users who are not in their friends list. Users can assign friends to specific groups and allow each group some attributes giving privilege to users of groups to view content and share information accordingly, for example one group user might be able to access complete information while others may get partial or less information to see. Users also have the option to display complete available information to other users who are not in the friend list, but such action is avoided as it is critical to safety and security of the user. On the other hand YouTube also has a user profile page available known as Channel where other users can visit and access the user’s information. YouTube videos are publically available to users and guest users of YouTube. Some videos might not be viewable if the users are not logged in so as to validate age authentication. Therefore social networks are composed of user accounts where users are connected through links with each other. Information will be available depending upon the users profile preferences. Users can search for other users and form a link by sending a friend request. To establish a friendship link between users, a mutual consent is required from both the target user and the sender (Mislove et al 2007).

**About YouTube and other Video Sharing sites:**

YouTube is a platform which does not only provide a video sharing service but a service which facilitates all the qualities of a social network. There are two ways where users can connect with each other, either a user subscribes to other users channel or add the other user as a fiend in their own friend list (based on mutual agreement). There are some major differences between the two connections:

Subscriber will get updates from users when they upload a new video on their channel. It does not matter if they are friends with them or not, whereas the friends of the users can also be subscribers and will get the update on the new videos uploaded by the user if they are subscribers. Friends can accept, share and invite other friends but users who are only subscribers do not have this option. Friends can send, receive messages and receive comments on the channel while Subscribes do not have this option.

There are many other websites which provide video sharing; some of the most popular ones are MetaCafe, DailyMotion and Flickr. Compared to its competitors YouTube is more popular as it allows users to upload unlimited number of videos as long as the content is not copyrighted. The video length provided for all users is up to a maximum of 15 minutes. Users can upload longer videos but the need to go through a verification process where the user account should be in good standing which is determined by YouTube community guidelines (having no copyright or community guideline violations), verifying the account with a mobile phone SMS verification and the account has no worldwide content ID block on any of the videos.

Chapter – II

**Background and Related Work**

Many researches have been made on Social Networks to understand their structure. They were based on crawling the network. A study by Mislove A. et al

Chapter – III

**YouTube Crawler: Design Specification**

In this chapter we will discuss about the design, methods used and functions of the Crawler.

* 1. ***Goals and Objectives***

The main objective of the crawler is to get the specific information about the videos on YouTube using the YouTube API and store the information in a database. The information includes:

* Video ID of the videos which have more than a hundred thousand views
* Their total view count
* View count from Facebook and Twitter if available
* View count from a mobile device
* Average rating of the video
* Uploader Name
* Category
* Upload Date

Some other information was also included and we will discuss it later in this chapter.

For the above information, crawler is supposed to follow a step cycle where it will keep on crawling the YouTube unless the program/application is stopped manually. It starts from scrapping the links from the video page and gets the video id from the links on that page and stores it in the database with the appropriate information associated with it. After completing this step, crawler will search and get the video id from the database, access it and follow the previous step as described. This is supposed to form a well-established cyclic process and continue without any hindrance. If it encounters any errors, it should bypass the exceptions and continue repeating the process. The safety of the data is also of concern as hardware or software errors would cause the data to be lost, so a back-up plan should be implemented to store the data on a back-up machine or a central server from time to time after each milestone. Once the data is acquired, the process of analysing can be started by transferring the data from the database to a more sophisticated application which can interpret the data accordingly.

These were the objectives before I started this project and implemented the design of my crawler program on this basis.

* 1. ***Program Design/Algorithm/Flowchart***

The design of the YouTube crawler is based on a mechanism where it will take a link to crawl, get the related links from that page, store it in the database then access the stored links in the database one by one and repeat the same procedure. The basic idea can be seen in these steps:

10->PROVIDE A VIDEO URL;

20-> GET the video URL;

30-> READ the PAGE

40-> SCRAPE the related links/URL`s from the page;

40-> ADD the filtered video id`s from the URL`s and related Information in the database;

50-> Access the Video id`s from the last added entry in the database;

60-> TRANSFORM Video Id to URL;

70-> GOTO 20;

In the beginning, the crawler is given a video URL/link, then it access that URL, looks only for the related videos on the page, gets the video id from the related video URL`s and adds them in a List of Strings. After scrapping the video id from the URL`s, it connects to the YouTube API and checks whether those videos have the view-count more than a hundred thousand or not. If the videos have more than a hundred thousand views, it contacts the API for further information for each video id. After all the processing, it adds the information to the MySQL database. Once this whole process is completed, it calls a method where it searches for video id`s in the database from the bottom to the top and each video id is transformed back into the URL and follows the same procedure from the beginning. A simple example of the crawler can be seen in figure 3.2(a).

A simple flowchart depicting the working of the YouTube crawler:

YouTube

API 2.0

CRAWLER

MySQL

Database

Transform Video ID to URL

Fig. 3.2(a): Figure describing the working of the YouTube crawler in a simple structure.

* 1. ***Programming the Crawler***

After attaining the design for the crawler I started with writing the code on a step by step basis. First thing I did is to make a simple crawler which can get the URL`s inside the html page. There are many ways to parse the html; the most basic way is to use the “inputstream” reader and the “bufferedstream” reader, when given a URL will return the html text of that page. Once it starts streaming the text, we can search for the text after “href=” inside the streamed line which will be a URL and add the result to a list. This method has a shortcoming as it uses “scanner” to find the URL, it will not remove extra whitespaces and characters present in the URL text and extra method will be required which will use conditions for checking and removing characters and whitespaces from the URL list. This method will not be accurate every time as there may be different characters in some of the URL`s which will be left unchecked. This method is useful for scraping data between the html tags.

The other methods are using the open source Java libraries like XPath, JSoup, Regex, JTidy, HTML parser, TagSoup etc. They provide the same basic function to parse the HTML page, and can be used depending on the way the coder wants to use them. For my crawler I used JSoup to parse the html as it is easy to understand, simple to use and excels in retrieving the URL`s from HTML tags. The JSoup API is very convenient to use for the extraction and manipulation of data. It can be downloaded from the official JSoup website.

*Example:*

{

Document doc = Jsoup.connect(url).get(); //Connect to the URL and get the html content inside the doc.

Elements links = doc.select("a[href]"); // Assign the data inside the href attributes to links which is an Elements type

for (Element link : links) { // Using for each statement to get each element inside links one at a time.

result.add(link.attr("abs:href")); //add those elements inside the link with attributes as href

}

The above example connects to a URL, gets the html inside the document and adds the URL`s inside the document to the Elements. This is simple and clean approach as it does not require extra conditions to check for whitespaces and extra characters in the URL.

For the second part, I filtered the URL`s as I only wanted the ones which have YouTube videos. These links were easy to distinguish as they contain “watch?v” in each video URL. After that I extracted video ID from those URL by looking for the 11 character long unique id present in all the YouTube video URL`s after “watch?v=”. For example, a YouTube video URL is like “**http://www.youtube.com/watch?v=kkXPmHIci4Y”** where we can easily separate the video id from the URL using the string functions.

After receiving the video id, I implemented a method provided by YouTube API service and passed the id into that to receive simple statistical details of the video which includes total view-count, average rating, number of likes and dislikes. After that I checked that if the video has more than a hundred thousand views using the “If” condition on the view-count, and then get the detailed statistical information or the insight information by calling a method from the other class “InsightViews.Java” which I created to scrape the data from the insight link. Once the crawler gets the information described above it opens a MySQL connection and adds the information into the database using SQL queries. All these functions were inside a method called “VisitLinks”.

Once I reached this step, I just had to repeat this process by retrieving the video id`s from the database and follow the same process from beginning. So I created a recursive method which will access the database and retrieve the video id`s starting from the last to the first added in the data table. Then I passed the video id`s one by one to the first method “VisitLinks” to repeat the process for the first batch of videos added in the database. But this query will end once it reaches the top of the table, so to continue this for new videos I called this method within itself to continue the crawling process repeatedly until stopped by the user. This approach is based on **Breadth First Search** algorithm were the node which is added in the last of the list of nodes is accessed first. This has one advantage and one dis-advantage

***Advantage:*** If the user temporarily halts the program or it is stopped because by physical errors, it will always continue its search from the last added video in the table. This will save crawler from redundancy of crawling the same videos from the beginning over and over every time the program is executed.

***Dis-advantage:***The program will reach the last (new) nodes only when it reaches the first node from the last crawl of videos, this will consume a little time as it will again crawl the videos which were crawled previously. But this also has a little advantage, since it will crawl the links on the top of the table again, it will also update the information in the database if there are any changes in the statistics or the insight data of the video.

*Example:* N videos are added in the first crawl and then crawler will access each video inside the database beginning from the Nth video added and then N-1, N-2 …………till it reaches the first video. During this time X more videos are added to the database. Now the total videos in the database is N+X; now crawler will again begin the crawl from (N+X)th video then (N+X)-1, (N+X)-2, …… till it reaches back to 1st video in the database again and repeating the same procedure for the rest of the additions.

Overall this approach has more advantages as the crawler will have bigger reach over the nodes (videos) and the data inside will be accurate and update with little expense of time.

There are a total of 3 classes made in the crawler; first is the “LinksExtract.java” which includes the main method and calls the other two classes. The second class is “InsightViews.java” which retrieves the insight information of the video. Here I used the buffered reader to read the insight document and added every line inside a string array list. Then I scraped the statistical information from the document and passed it to the variables associated with it (which were called by the main method).

The third class retrieves the Uploader, Upload Date and the category of the video. I created the third class after 2 days of my crawling as I also wanted to add these attributes of the video in the database. So before I continued crawling, I modified the third class to update the database with this new information about all the videos previously added. After this was completed, I then changed this class and called it from the class containing the main method to retrieve these attribute`s values.

InsightViews.Java

UploaderDetails.Java

LinksExtract.Java

MySQL Database

**YouTube CRAWLER**

Fig. 3.3(a): Diagram showing the classes (blue) and their relationship inside the crawler.

After I stopped the crawler, I created a method in a new class which calls the database and transfer all the data from database to a MS-Excel workbook on the desktop. This method was used to search for different combination of data with different SQL queries and the final result was saved in “.xls” file on desktop. This was very helpful as it helped me analyse different combination of data in the Excel file.

The Crawler used 70 hours to crawl over 64000 videos and add 31,300 to the database. Although I ran the crawler for a week but during that period I had to pause crawling when my laptop overheated.

***Libraries Used:***

The program used many different Java libraries: YouTube API, JSoup and MySQL connector.

***Database Design:***

Database was designed on the basis of information to be added inside. The table 3.3(b) describes the Meta-Data of the table describing the structure.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **KEY** | **DEFAULT** | **EXTRA** |
| id | BIGINT(13) | NO | PRIMARY | NULL | auto\_increment |
| title | VARCHAR(60) | YES |  | NULL |  |
| vidid | VARCHAR(12) | YES | UNIQUE | NULL |  |
| totalViews | BIGINT(14) | YES |  | NULL |  |
| fbviews | BIGINT(9) | YES |  | NULL |  |
| mobviews | BIGINT(9) | YES |  | NULL |  |
| twviews | BIGINT(9) | YES |  | NULL |  |
| avgRating | FLOAT(12) | YES |  | NULL |  |
| nLikes | BIGINT(7) | YES |  | NULL |  |
| nDisLikes | BIGINT(7) | YES |  | NULL |  |
| uploader | VARCHAR(30) | YES |  | NULL |  |
| category | VARCHAR(30) | YES |  | NULL |  |
| uploadDate | VARCHAR(30) | YES |  | NULL |  |

Table 3.3(b): Table representing the Meta-Data of the table “links” in the database.

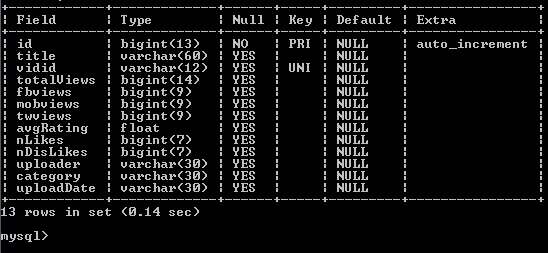


Fig. 3.3 (c) Figure showing the SQL query describing the table structure (same as table 3.3(a)).

* 1. Software and Hardware Used

I used the following software for writing my crawler, storing the information and creating a backup after a certain time-interval

* Netbeans IDE 7.1.2
* Java JDK 1.7 – To write my crawler in Java
* MySQL database – Free and Easy to use, best for small scale projects
* GitHub – Backup project in repository using Git revision control system.
* Microsoft SkyDrive – Back Up data on another source.
* Laptop – Core i3 Processor with 3Gb RAM
* OS – Windows 7 Ultimate 64 Bit

Chapter – IV

**Statistics and Analysis**

**Statistics of the Crawled Data**

The YouTube crawler ran for six days crawling over 62,799 videos out of which 31,302 videos were added in the database. Out of the total videos crawled, 12,280 videos did not have any statistics available. 3,850 videos had Facebook view count available and 18,892 videos had been viewed from a mobile device. Twitter view count was only available for only 56 videos out of the total crawled videos which constitute 0.18% of the total data. The videos which had no statistics available are excluded from the Facebook, Twitter and Mobile Device view analyses as these videos have no information available to public. YouTube allow users to hide the statistics of their uploaded videos from the public view. The other exception with the crawled data is that the videos which did not had any Facebook view-count available might not necessarily be lacking views from Facebook but those numbers must be very insignificant that they were not available in the statistics section of the video and can only be seen by the user of the channel. The statistics of every video are updated once in a week on YouTube servers. This information comes in detailed statistics known as YouTube ‘insight data’. With the help of YouTube API users can only retrieve the insight data of only those videos which are owned by them, thus restricting them to view the insight data from other user’s channel using the API of course. But the insight data of other user`s videos are available to view on the video page, if and only the user of the particular video has allowed them to be publicly available. They can be viewed by clicking a small button on the video page, which can be seen under the video player next to the total view-count as shown in the figure 5(a).

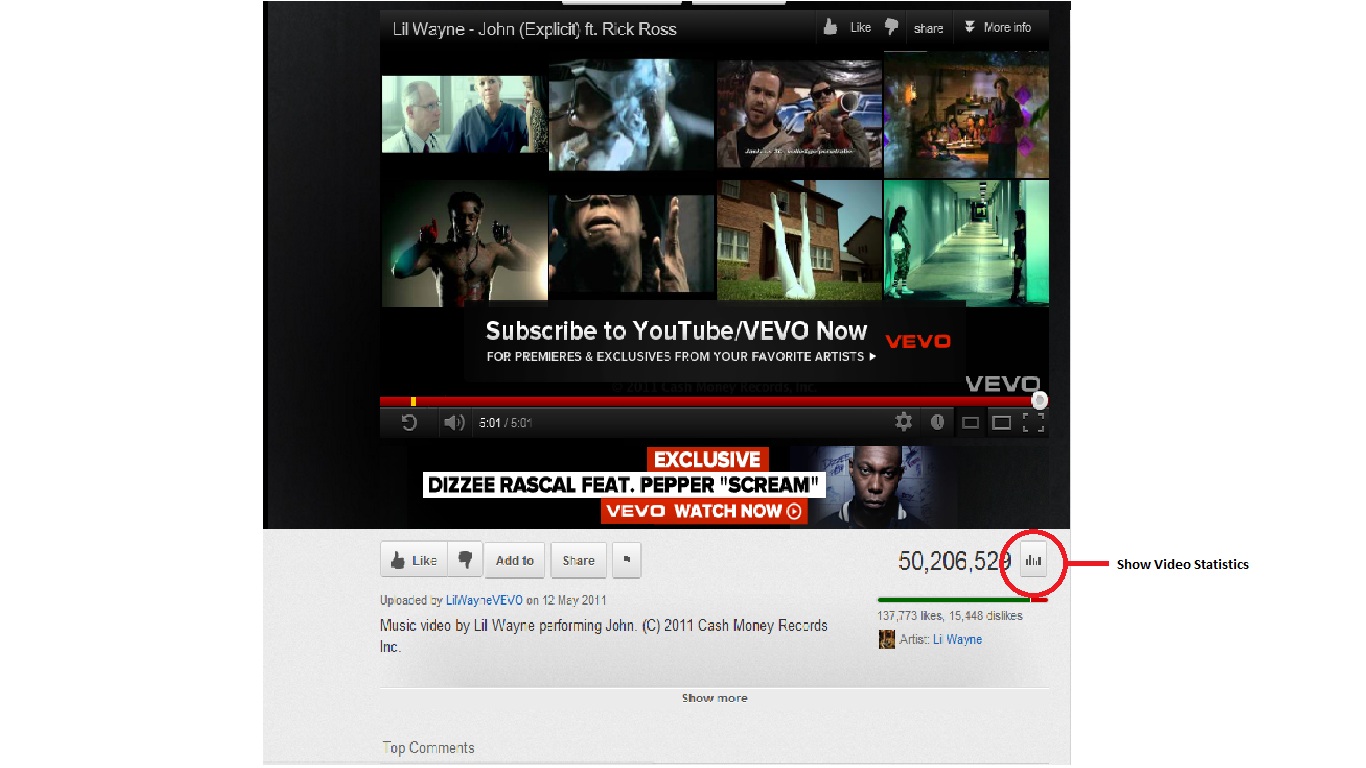


Fig. 5(a): Image showing the button where user can click to see the detailed statistics about the video.

More detailed information can be seen after clicking the button.

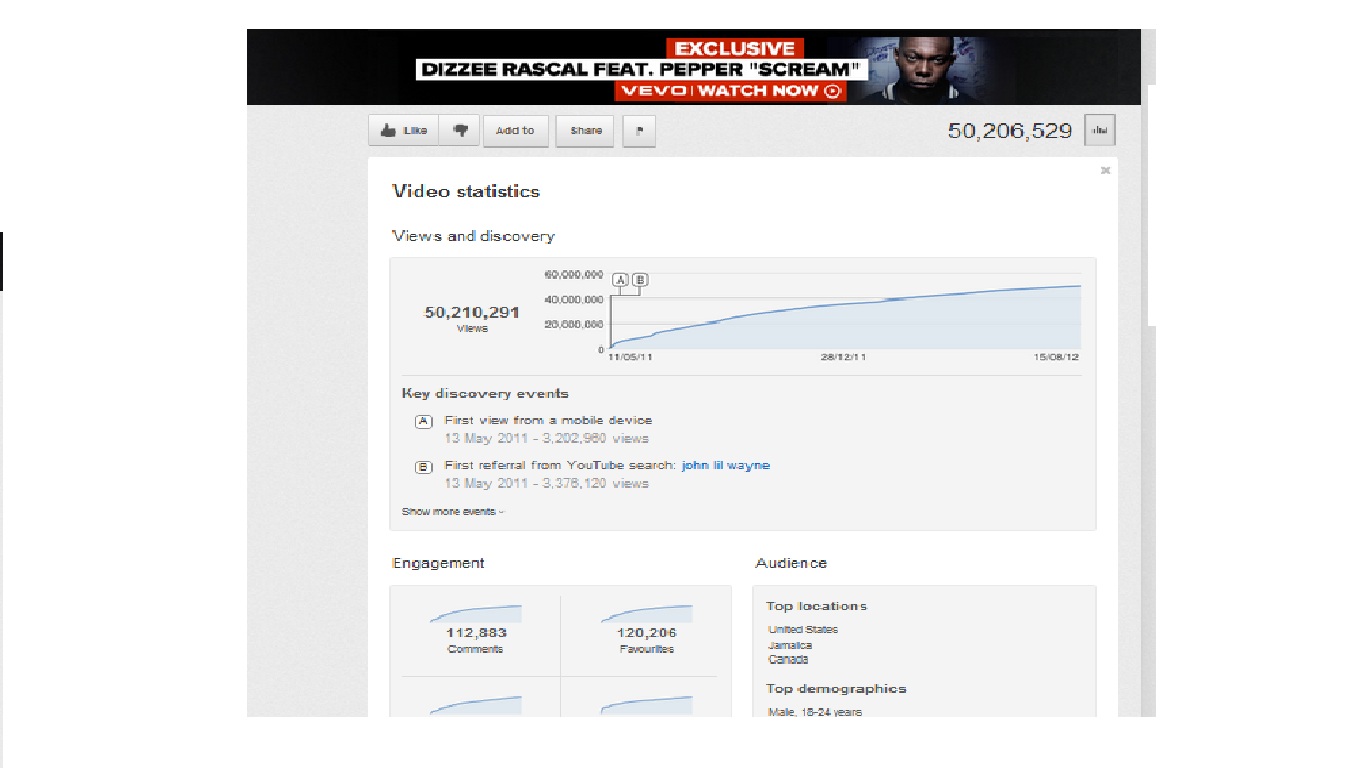


Fig. 5(b): An example image showing the detailed statistical information of a YouTube video.

This information is not presented in the HTML when viewed in the source code viewer of the browser as YouTube restricts users to scrape this data. So it is impossible to retrieve the insight data of other users from the API and scrapping it from the web page.

But as we know that the insight report can still be seen, there is another way around to get this data and scrape it. After researching a little more, I used the Google Chrome Browser`s Tools->Developer Tools to see what action takes place in the background when we click the show statistics button. I was able to retrieve the insight link which gets the insight information: “[**http://www.youtube.com/insight\_ajax?action\_get\_statistics\_and\_data=1&v=(VIDEO\_ID)**](http://www.youtube.com/insight_ajax?action_get_statistics_and_data=1&v=(VIDEO_ID))**”** from the resources section of the Developer tool tab. This link was hidden inside the following information in the resources tab (shown in highlighted text):

("yt.www.watch.actions.stats",function(a){qn(a)&&(ln(),X**("/insight\_ajax",{**format:"XML",method:"GET",o**:{action\_get\_statistics\_and\_data**:1,v:L("VIDEO\_ID")},j:function(a,c){nn(c.html\_content);vard=G("stats-opt-out-chbox");d&&S(d,"change",function(){Ym(!d.checked)})},A:on}))});t("yt.www.watch.actions.unlike",function(){kn();var a=1==en();gn(a?2:1)});

The above code can was taken from the Google Resources tab shown below.

**

Fig. 5 (c): Image showing the Chrome browsers Developer tool to view the code generated by the Show button on YouTube.

When using the insight link inside the browser, it automatically downloads an XML file which contains the HTML tags with data, opened and closed by the XML tags as shown in the following code example.

**<root>**

**<html\_content>**

**<![CDATA[**

<div class="watch-actions-stats"> <div class="stats-header"> <h1> Video statistics </h1> </div> <p>

…………………………………………………………Some HTML Tags and Data……………………………...............

<span>First embedded on:</span> <span class="extra"> <a rel="nofollow" href="http://facebook.com" dir="ltr">**facebook.com**</a> </span> </p> <p class="sub-data">**12 Jul 2012 - 10,009 views** </p></dd> </dl>

…………………………………………………………Some HTML Tags and Data……………………………...............

**]]>**

**</html\_content>**

**<return\_code>**

**<![CDATA[ 0 ]]>**

**</return\_code>**

**</root>**

Fig. 5(d): An example representing the code from the file downloaded from the insight link.

Within this content all the insight information was available which could be seen on the video page. Removing the XML tags from the above content and saving it as a HTML file displayed the following information which is shown on the video page.

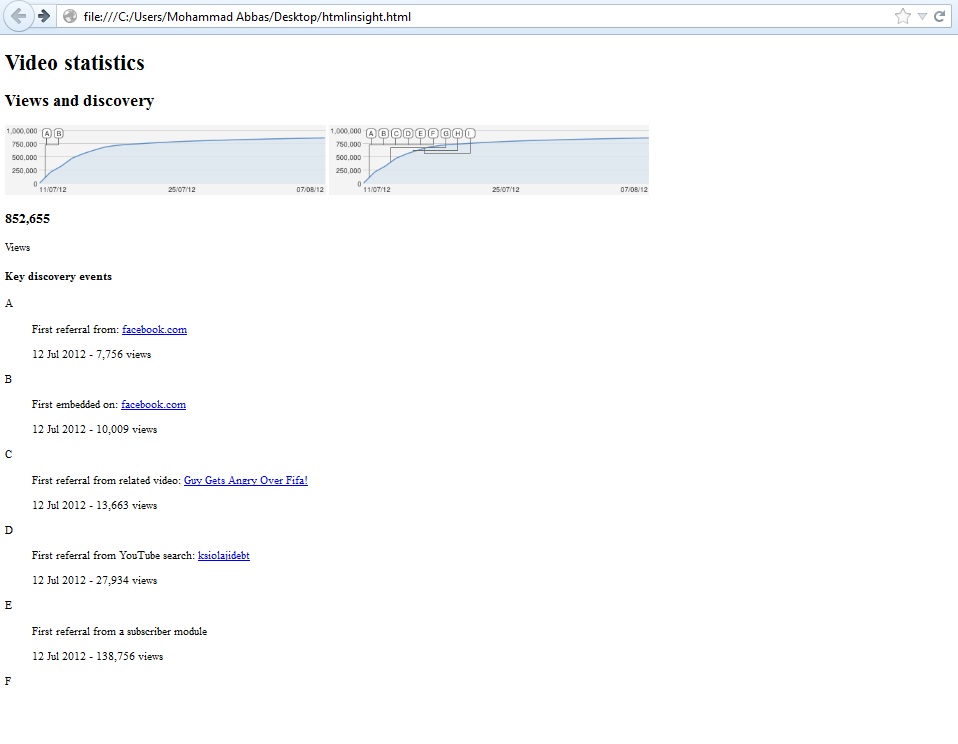


Fig. 5(e): Image showing the statistical information of the downloaded page from the Insight link. The XML tags are removed from the file and saved it as HTML.

After getting the data from the insight link it was possible to parse the XML page in Java and scrape data within the HTML document. These details are already discussed in the “Crawler” chapter.

**Analysis of the Crawled Data**

Crawling all videos and retrieving complete information from YouTube would certainly take several months as YouTube is a swarm of millions of videos. Due to the small nature of this project we take a small dataset of videos to examine the patterns of viral and most popular videos which had total views over a hundred thousand and the effect of Facebook and Twitter on their popularity.

As the database had 31,300 videos in the database, the distribution of videos belonging to each category is shown in the graph 5(f).

Fig. 5(f)

*Formula Used:*

*Percentage of Videos by category = (Total Videos in the Category/Total Number of Videos in Database) X 100*

The maximum numbers of videos crawled are from “Music” and “Entertainment” genres while the least number of videos in the database are from “Trailers”, “Shows” and “Unspecified Category”.

1. *Video Categories and Ratings*

YouTube allows its users to tag their videos under 27 different genres in general and 19 different genres within the movie-genre. The reach of my crawler ranged up to 17 different genres and one unspecified genre where it was unable to get the information about the genre of the video. There are only 40 videos which were stored in the unspecified category. The Average Rating for a video ranges from 1 to 5 and is calculated by this formula:

*Average Rating = ((Number of Likes \* 5) + Number of Dislikes)/Total Number of Ratings*

*Total Number of Ratings (Raters) = Number of Likes + Number of Dislikes*

*Average rating for each category = (Total Ratings of videos in the category/Total number of videos) X 100*

Fig. 5.1(a)

The graph 5.1(a) describes the Average rating of the videos in different categories of the entire dataset. “Trailers” and “Music” categories were the ones that were most rated with an average rating of approximately 4.8 whereas News had the least average rating of 3.9. The third most highly rated category is “Shows” with an average rating of approximately 4.7. Categories in between “Gaming” and “Travel” had an average rating ranging between 4 and 4.5. According to the above scale “Trailers”, ”Movies” and “Shows” seems to be the most popular category, but to support this assumption we also need to take a look at the number of raters for each category as a video can have 10 likes and 1 dislike to have more average rating while the other which has more raters will become insignificant although having more views.

The total number of likes and dislikes for all the categories were 230,938,984. The graph 5.1(b) describes the percentage of ratings for each category (total number of raters for each category).

*Formula Used:*

*Total Raters = Number of Likes + Number of Dislikes*

*Raters Percentage= (Total Raters in the category/ Total Raters of all categories) X 100*

Fig. 5.1(b): Graph representing different YouTube Categories and their total raters in percentage.

It is quite clear that “Music” has the maximum number of raters while the other categories have less raters compared to it. The numbers of raters for “Trailers” are very less compared to other categories because of the less number of videos crawled in this category, so it will not be possible to measure its popularity in this case. Keeping in mind that only those videos were crawled which had more than a hundred thousand views, “Music” seems to be most popular when we measure it using ratings and view counts.

1. *Facebook Views and Categories*

In this section we will take a look at the number of views for each video category directed from Facebook. Out of 31,300 videos in the database, 3,850 videos had Facebook view-count available and the total view-count from those videos is approximately 496.9 million. The graph 5.2(a) represents the statistics of the Facebook view-count compared to the categories.

*Formula Used:*

*Facebook View-Count percentage by category = (Facebook Views from the category/Facebook Views from all the categories) X 100*

Fig. 5.2(a): Graph showing the total Facebook views in percentage for each category with respect to the total Facebook view-count of all categories combined.

It is very intriguing to know that with only around three thousand videos; approximately 497 million views are directed from Facebook for the available categories and approximately 325 million views are from music videos, which is quite large and keeping the fact in mind that these are only a small set of videos which had over a hundred thousand views. In this aspect we infer that “Music” has more Facebook views compared to other categories like “Comedy”, “Entertainment” and “Gaming”. To prove this we need to take a look at more detailed information about the number of videos we have in database. Within this total segment of videos in the database with Facebook view-count, 163 are from “Car” category, 657 from “Comedy”, 526 from “Entertainment”, 151 from “Film”, 134 from “Gaming”, 967 from “Music”, 346 from “People”, 152 from “Pets”, 138 from “Science” and 292 from “Science”. The remaining categories had less than a 100 videos in the database with Facebook view-count. More detailed information can be seen in Fig 5.2(b).

*Formula Used:*

*Videos with Facebook view-count in percentage (Category based) = Videos with Facebook view-count in the category/Total Videos with Facebook view-count) X 100*

Fig. 5.2(b): Graph representing the number of videos in database which have Facebook view-count corresponding to different categories.

If we compare the values in graphs 5.2(a) and 5.2(b) we can get the average Facebook views of a video in each category. For this we use the formula:

***Average Facebook Views of a video in a Category = Total Facebook Views in the Category / Total number of videos with Facebook view-count in the Category***

From the above formula we calculated the values and obtained graph 5.2(c).

Fig. 5.2(c): Graph showing the average Facebook view-count of a video in different categories.

It can be seen that an average Facebook view for a Music video has near about 340 thousand views whereas rest of the categories do not have more than 100 thousand views. Therefore we deduce that “Music” category is not only the most popular, but they are shared more on Facebook and are likely to be viewed more when shared on it. “Travel” is the next most shared and viewed video category on Facebook with an average view for each video is more than a 100 thousand. The remaining categories like “Entertainment”, “Pets” and “News” etc. have average views in between 50 and 100 thousand.

I search for popular music videos myself and found out that **Vevo** which has different multiple YouTube channels with name of different artist and is currently the most popular uploader. Vevo does have its own website where users can view videos of popular artists. Vevo also has a Facebook Page where they share the links of the latest videos they upload on YouTube and it has millions of subscribers on Facebook as well. More detailed information about it can be viewed on Wikipedia.

1. *Facebook View-Count Versus Total View-Count*

In this part we will examine the contribution of Facebook in the total view-count of the videos. We will discover the extent to which Facebook helps the videos to become viral or popular. For this we take the total Facebook view-count and the total view-count of the videos which have Facebook view-count available in our database. Then we take out the percentage of Facebook view-count by the formula:

***Facebook view-count percentage of the category = (Total Facebook Views of the Category/Total Views of the category) X 100***

Graph 5.3(a) represents the Facebook View percentage of the total view-count of individual categories.

Fig. 5.3(a): Graph showing the percentage of Facebook Views out of Total view-count with respect to each category.

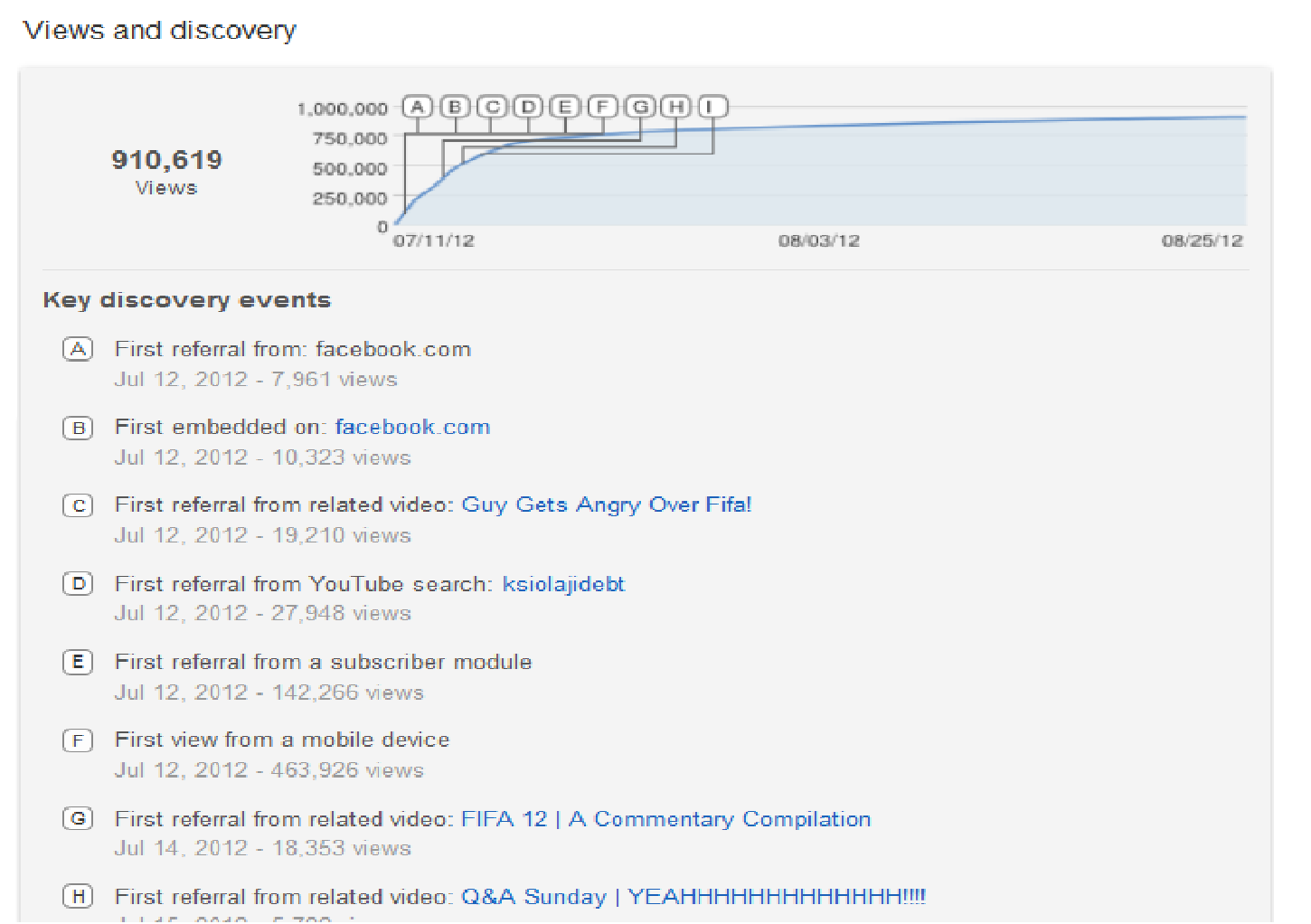
The table shows more detailed information about the Facebook Views vs. Total Views. With a total of 3849 videos have approximately 497 million Facebook views against a total of 10.2 billion total view counts for all the individual categories.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **No. of Videos** | **Total Views from Facebook** | **Total View-Count** | **Category** | | 163 | 6539048 | 192639230 | Cars | | 657 | 37311704 | 896154785 | Comedy | | 83 | 3692018 | 69007968 | Education | | 526 | 41936469 | 1073944625 | Entertainment | | 151 | 9010906 | 201156363 | Film | | 134 | 2953759 | 88303865 | Gaming | | 48 | 2182134 | 69916699 | Howto | | 964 | 325103335 | 6260304963 | Music | | 137 | 12248855 | 159921457 | News | | 21 | 1389262 | 25044977 | on-profits | | 346 | 19004991 | 426455352 | People | | 152 | 11219891 | 295657560 | Pets | | 138 | 4086430 | 109441968 | Science | | 1 | 6580 | 572226 | Shows | | 292 | 16443345 | 297865911 | Sport | | 1 | 26482 | 1771049 | Trailers | | 31 | 3599078 | 39074125 | Travel | | 4 | 177021 | 5994456 | unspecified | | **3849** | **496931308** | **10213227579** | **TOTAL** | |

Table 5.3: Table describing about the total number of videos with Facebook views, total Facebook view-count, total view-count and category.

From the graph 5.3(a) we observe that Facebook contributes at least 7% of the total view-count for the “News” category and around 9% for the “Travel” Category. “Music” which is the most popular category has nearly 5% of the contribution from Facebook with “Education”, “on-Profits” and “Sports” on the same level. Rest of the categories range in between 2-4% which is less compared to other categories.

The noticeable point here is that none of the categories have a view-count from Facebook which contributes more than 10% of the total view-count yet majority of the categories had an average Facebook view count of more than 50 thousand. This data only provides us the partial facts to understand the relationship between the Social Network. To find out the complete contribution of Social Networks for the viral videos on YouTube would also need to require the date at which the video was shared on Facebook. This will help us differentiate between the total view-count in between that period and after it was shared on Facebook. YouTube detailed statistics does provide this information but directly on graph (which is in an image format and cannot be scrapped for information) so it is not possible at the moment to get that piece of information. Date/Timeline plays an important role as the links shared on Facebook fade away from time to time as the walls of main page are filled with new update posts and messages every second. As some of the videos once uploaded on YouTube are immediately shared on Facebook so they get their starting views from Facebook and once they reach thousand views within few days or a week they become highlighted in the YouTube`s main page in the list of popular videos and people visiting the main page visit that video on a usual basis. Then the chain of high hits on that video increases and that video become viral, thus Facebook has an indirect role for the popularity of that video. This is true for some videos as I have inspected them manually. For example, in the given fig 5.3(b) we can see that the video was uploaded on 11/07/2012 and then embedded and shared on Facebook on 12/07/2012 which immediately for it total views of 18,000 from Facebook.

Fig. 5.3(b): Image showing the detailed statistical information of a video having views from various sources.

**(The video can be viewed on: http://www.youtube.com/watch?v=dxTNjexGkkw&feature=plcp)**

Perhaps getting the date on which the video was shared on Facebook will help us create a more thorough understanding about their relationship. We will discuss this aspect in the Chapter “Future Work”.

1. *Twitter View-count*

Unfortunately the crawler was not able to find many videos which had Twitter view count available. Although 700 videos are shared on twitter each minute (YouTube statistics), but the crawler was not able to crawl enough videos which had it. Out of 31,300 videos only 55 videos were scrapped which had Twitter view count. The crawler`s design was intended to search for viral videos specifically so those videos did not had the Twitter view count. This dataset is not enough to support any proofs and reasonable assertions for our research questions, still I would like to discuss about the data we have. Table below shows the total number of videos distributed among categories having Twitter view-count and describes the total twitter view count and number of videos having twitter views for each category.

|  |  |  |
| --- | --- | --- |
| **Total Views from Twitter** | **No. of Videos** | **Category** |
| 501068 | 3 | Comedy |
| 27374 | 2 | Education |
| 251619 | 13 | Entertainment |
| 45538 | 15 | Gaming |
| 32696 | 3 | Howto |
| 44245 | 6 | Music |
| 62948 | 2 | News |
| 37798 | 2 | People |
| 7490 | 1 | Science |
| 195085 | 7 | Sport |
| 18636 | 1 | Travel |

Table 5.4: Table showing total twitter views, no. of videos having twitter views and corresponding category.

As we can see we have a small set of videos 11 different categories and the majority of them belong to “Gaming” and ”Entertainment”. The most number of views were given by “Comedy” with only 3 videos which area round half a million.

At the moment we cannot say anything about the viral videos and popularity in relation to Twitter tweets but we can get more information in future to discuss this aspect.

1. *Mobile Device Views*

In the crawling of YouTube videos, I encountered another interesting thing about the viral videos; the view-count of these videos is also dependent on mobile devices. YouTube video services today are provided on almost every electronic gadget like mobile phones, gaming consoles, tablets and pads etc. People can view videos either on the browsers provided on the device or from a YouTube application specifically made to watch videos.

YouTube insight data provides the users of the video to watch the view count they are get from mobile devices. Public Users can also view the same information provided under the statistical details about the video.

For this study same method was applied to scrape the view-counts of Mobile Device views like the Facebook view-counts. A total of 18,891 videos were stored in the database which has mobile view count available from a total of 31,300 videos. After receiving the total view count, it was surprising to find out that almost every Category of video had majority of views from Mobile Device. The total mobile views for all the 18,891 videos are 4,559,127,368 which is approximately 4.56 billion and the overall view count of all these videos is 39565087299 which is near about 39.56 billion. The percentage of total views from the mobile device against the total view count is 11.52%, which is high as it denotes that a big percentage of population visit YouTube via mobile and other devices. Figure 5.5(a) represents the distribution of categories against total mobile device views of respective categories.

|  |  |
| --- | --- |
| Fig. 5.5(a): Graph showing the Categories against the total number of views from the mobile device. The scale interval of X-axis is 200 million between major gridlines and 50 million between minor gridlines.  Fig 5.5(b): Graph representing average view-count from a mobile device from each category. |  |

*Formula Used: Average view-count of a video = (total mobile device views in that category / total view count of that category) \* 100*

|  |  |  |  |
| --- | --- | --- | --- |
| **No. of Videos** | **Category** | **total views from mobile** | **total view-count** |
| 884 | Cars | 84568151 | 912552737 |
| 2443 | Comedy | 689463987 | 3767311331 |
| 483 | Education | 64113792 | 428549618 |
| 3238 | Entertainment | 915839765 | 4944558725 |
| 1170 | Film | 392973384 | 1977042834 |
| 2696 | Gaming | 365950576 | 1548653627 |
| 495 | Howto | 138555291 | 685036553 |
| 2217 | Music | 1092097469 | 19152785399 |
| 780 | News | 97104723 | 881502772 |
| 105 | on-profits | 12371802 | 86324698 |
| 1502 | People | 299611412 | 2019177495 |
| 508 | Pets | 111920432 | 898536816 |
| 963 | Science | 98677303 | 658163272 |
| 27 | Shows | 7193658 | 37399163 |
| 1178 | Sport | 150255690 | 1295617218 |
| 2 | Trailers | 734430 | 6975646 |
| 172 | Travel | 32445390 | 241262077 |

Table 5.5: Table representing the total no. of videos with the total mobile view count and overall view count with respect to each category.

From Graph 5(a) we can see that “Music” was the most viewed category on the mobile devices followed by “Entertainment”, “Comedy”, “Film” and “Gaming”. “Music” has approximately 1.09 billion views from mobile devices for only 2,217 videos (Table 5.5). This is a very large number of views for only a small number of videos signifying that majority of YouTube viewers use mobile devices to access YouTube. It also tells us that “Music” is the most viewed category from mobile devices, but to establish this fact we will take a look at the average mobile view-count for each category and compare them.

“Entertainment” has approximately 916 million mobile views for only 3,238 videos. “Comedy” has around 689 million views, where as “Film” and “Gaming” have 392 and 365 million views respectively (Table 5.5). Graph 5.5(b) represents the average views of a video from a mobile device for each category. “Music” has an average view of 492 thousand from the mobile devices whereas “Film” has an average view of 335 thousand. “Entertainment”, “How To” and “Comedy” have and average view of 290 thousand. “Trailers” is excluded from our analyses as the number of videos it has in the database are not enough for comparison. The remaining categories have an average view in between 100 and 200 thousand. Overall it can be concluded that all the categories had enough views from a mobile device and “Music” was the most popular category among them.

We move on to the final part where we will determine the overall impact of mobile device views on the total view-count of the videos. As we know the total view count from the mobile devices for all the videos are 4.56 billion and the total view-count of these videos is 39.56 billion and the overall percentage of mobile device views in total views is 11.52%. Now we take a look at the overall percentage of mobile device views for each individual category.

Fig 5.5[c]: Graph representing the Mobile Device view-count in percentage with respect to total view-count of individual category.

According to the graph 5.5[c], it can be seen that “Gaming” and “how to” videos are more watched on mobile devices with respect to their total view counts. Almost every category has a 10-20% of total view-count from mobile devices except “Cars” and “Music” which has a mobile view-count of 9% and 6% respectively. Music videos have less percentage here as they are viewed equally from other sources because of their popularity. Finally we can conclude that Mobile Devices have their own significance on the popularity of the videos as YouTube is not limited to PC`s and laptops.

*6. General Information about all the videos*

Here we take all the 31,300 videos in the database and examine the information they have about the total view counts based on different category. The total view count of all the videos is 78.59 billion views. The average view for each video in the database is 251 million views. The following information is given in the table 6.1.

|  |  |  |  |
| --- | --- | --- | --- |
| No. of Videos | Category | Total View Count | Average View |
| 1265 | Cars | 1366433416 | 108018452 |
| 3749 | Comedy | 7041406321 | 187820921 |
| 707 | Education | 927693830 | 131215535 |
| 5028 | Entertainment | 8794524676 | 174910992 |
| 2067 | Film | 3303480205 | 159820039 |
| 3417 | Gaming | 2123872810 | 62156067 |
| 661 | Howto | 908884244 | 137501398 |
| 6527 | Music | 43667064093 | 669021972 |
| 1025 | News | 1241741382 | 121145501 |
| 171 | on-profits | 164055961 | 95939158 |
| 2384 | People | 3652089303 | 153191665 |
| 683 | Pets | 1254975729 | 183744616 |
| 1279 | Science | 962622231 | 75263662 |
| 31 | Shows | 44783875 | 144464113 |
| 2006 | Sport | 2693241151 | 134259280 |
| 2 | Trailers | 6975646 | 348782300 |
| 258 | Travel | 404177631 | 156657997 |
| 40 | Unspecified | 32254859 | 80637148 |
| 31300 | **TOTAL** | **78590277363** | **251087148** |

Table 5.6 : Table representing all the videos in the database with their total and average view-count corresponding to each category.

Fig 5.6(a): Graph representing the total view-count of each video in the database with respect to their category. The scale interval on Y-axis is 5 billion (nE+10 represent values in billions. Eg: 5E+09 = 50X10^9= 5 billion).

*Formula Used (table 5.6):*

*Average View Count of a category = (Total view count/Total Number of videos in that category)*

* 6,527 videos from Music category have a huge total of 43 billion views. The average view count for videos in Music category has 669 million views approximately (Table 5.6) which is the highest among all the other categories.
* 5,028 videos from Entertainment category have a total of 8.79 billion views next to Music. The average view count for this category is 174.9 million views (Table 5.6).
* Comedy has the third highest total view count of 7.04 billion views. The average view count for this category is 187.8 million views (Table 5.6).
* Rest of the categories have less than 5 billion views.

From the above inferences we can state that Music is the most popular category and highly viewed from different sources such as Facebook and Mobile device. The average view count signifies that for each video there are at least 669 million views for music videos.