Online Video Recommendation System : Youtube Video Dataset

Anonymous

ABSTRACT

Recommender Systems by definition happen to be a type of information filtering system which is more focused on predicting on what a user might prefer with respect to an item which could be either videos,news articles and so forth.

The application that we intend to design is based on YouTube video recommender systems. In order to implement the application we utilize the YouTube data set and the YouTube API .The data set consists of 10 attributes and 3170 records. The description of the data set happens to be with respect to the video id, the uploader's user-name, the age attribute pertains to the number of days lapsed between the day video was uploaded up until February 15th 2007 and a few more. The YouTube API provides a platform to design customized systems and in our case it happens to be a Video Recommender System. The data set consists of 3170 records pertaining to the information with respect to the videos watched by the respective user specified in the uploader's username attribute. The description of the dataset is shown on Table 1. We intend to implement the visualization feature through the following attributes namely the views and rate. The views would give us a better perspective on the number of views with respect to the video and rate pertains to the user likability of the video and through this we would be able to provide a basis for the recommender system. Apart from the YouTube API we intend to use Weka which happens to be a data mining tool. Weka would assist us in visualizing data for this application in addition to providing a platform for mining. Through that we would be able to comprehend certain patterns which would give us a better outlook on the user preferences. For instance when a new user profile is created the recommender system would recommend videos with the maximum number of views at that instant of time. This way we intend to build a recommender system which would be based on user's preferences pertaining to the number of views and the likability ratio.

1. INTRODUCTION

In general, recommender systems can be categorized into two sections namely Collaborative Recommendations and Content-Based Recommendations. Collaborative approach pertains to building a system based on the user's past references in collaboration with similar decisions made by other users and this approach is used as a basis in order to recommend or suggest items based on the user's interest. Content Based approach focuses on the characteristics of an item rather than user's interests. A few applications which are based on collaborative and content based approaches are Last.fm, Pandora Radio, Facebook Ad suggestion, Netflix, YouTube and so forth.

We happen to refer these two papers [2] [3] in order to get a better perspective on building an Online Recommendation System utilizing the YouTube platform. As discussed in the first phase of our project we intend to build a video recommender system based on the user's preferences and likability ratio. In order to comprehend the preferences of an individual user we would follow the collaborative approach where in a video with the most number of views and that is pretty evident in the views attribute of the data set and in order to retrieve the videos based on the likability ratio we consider the ratings attribute which would give us a better idea of which of the videos have been rated highly. In the current phase of the project we discuss the data cleaning and preparation part with respect to the data set we happen to possess and how we contemplate on utilizing the YouTube analytics and data API.

2. MOTIVATION

Given the way social networking has been making its presence felt everywhere apart from posting pictures people would want to interact with one another by sharing videos and this paves way to friends commenting on each other's video by presenting their views by liking it and by letting know their preferences . This was our motivation for the project.The advent of smartphones and tablets has shown us that one can click shoot post images and videos almost instantly anywhere giving us access to one's preferences.It also tries to show social networking in a new light where in it isn't restricted to just posting images or updating statuses.This provides us a new platform to interact and measure user preferences.This provides us a better ways to communicate with users providing different services.

3. DESIGN AND IMPLEMENTATION

In this project we have used the YouTube Analytics API which facilitates in retrieving the insight data along with the YouTube Data API. Coming to the YouTube Analytics API which is termed as YouTube Insight is an analytics and report engine which provides viewing statistics, popularity metrics and demographic information for videos and channels. A video entry consists of a link to insight data if the authenticated user retrieving the entry owns the video. A profile entry consists of a link to insight data for the channel if the authenticated user is retrieving his or her own profile.

Since the data is formatted in the html form the link attribute is taken into consideration with the URL pointing to the file saved in the .csv format. The file is nothing but the report that we happen to use as a data source in our project. Each report happens to consist of worldwide video related data for a seven day period. The report includes "userstarttime" and "userendtime" values which signifies the time stamp ordering for the videos viewed.

The video demographics are stored in the VIDEOID demographics locations1.csv file which consists of following fields:

1. The video id - a unique identifier which is used to designate each YouTube video followed by the title which signifies the title of the video, the age group which signifies the generic age group viewing or subscribing to the particular video, the percentage would give us an estimate as to how many in the specified age group are actually viewing the video.
2. The views demographics are stored in the VIDEOID DATERANGE REGION views1.csv file which consists of the following fields

3. The date field signifies the day the video views occurred

4. The region field signifies the country where the video views occurred.

5. The views field signifies the number of times the video was viewed.

6. The popularity field signifies the metric that measures the popularity of the video as measured by views.

7. The rating 1 field signifies that the video was the least liked by the users. If this field has a larger value that signifies the video was the least liked.

8. The rating 5 field signifies that the video was the most liked meaning it has a higher rating amongst users with large number of users having viewed and liked it the most.

9. The values in the rating field 2 - rating field 4 signify the likability ratio amongst users.

3.1 Data cleaning and preparation

As the first step towards our project, we have cleaned the data. The dataset obtained had certain inconsistencies which had to be cleaned before we started to mine through the dataset. In regards to that, we used Google Refine to remove any inconsistent data that were present in the dataset. For example, there were several values that were not present under the related IDs field. The reason behind this could either mean that the particular video does not have related videos to it or it could also mean that the value is missing. Thus, such values were filtered and replaced as "MISSING VALUES". Also, there were certain records that did not contain the userID which was again replaced by MISSING-VALUE. This was done by sorting that particular column and then replacing all the blank records with MISSING-VALUES. Also, there were certain empty rows which were eliminated for consistency. One of the other issues that we found was, while sorting the data based on the age of the video, we discovered a discrepancy, wherein the age was 0 but it had 203 views. This is practically not possible. Also, we noticed that there were no videos related to that particular video. As we did not find any use out of this record, we removed that row from the table.

We also merged the tables of different dates as we thought the dataset from one day was too small to mine through and find patterns in it. We also noticed another flaw after merging the tables, some of them had around ten related video IDs associated with each video while some had eleven. Thus, in order to even it out we removed the eleventh column for consistent results. Also, certain fields like comments had negative values in it which is not allowed. In this case, we did not remove the value, but rather changed the negative value to positive value assuming that it would have been a mistake while the data was entered into the table.

4. CURRENT WORK

As stated in the project specification we intend to build an online recommender system where in we happen to refer the youtube data set which consists of the usernames of uploader, the video id, the age, the category, length, views, rate, rating, comments and related ids attributes and so forth .Since we utilize the youtube data api for our current phase of project we did come across various code samples as stated in the youtube api site and got a chance to utilize the snippet present in the gdata-java-client.googlecode site[1] which has access to a variety of feeds it happens to list a bunch of options namely retrieving the standard youtube feeds , searching a particular feed , searching a feed based on categories and keywords , retrieving a list of a particular user's uploaded videos along with one's favorite videos, along with responses comments a list with a user's playlist ,list of user's subscriptions and the details of a user's profile . After choosing one of the options we retrieve the input by keying in the username or the uploader's name through standard input what we happen to implement through our project. The stored list of uploader's name listed in the data set is passed as the input through to the Client program mentioned above to retrieve the list of favorite videos based on the uploader's name.

Each of the options described in the list happens to have a customized function or method related to it. The Youtube client java file (which we have named as getRecommendtion) reads through the list of usernames that has been segregated from the original dataset. This is fed into the java file and the API is used to retrieve the list of videos that has been marked as favorite by the user. Later the keywords in the title of the videos are alone extracted out.

Table 1: Dataset Description

Attribute	Description : Datatype
VideoID	Unique Identifier for a video: String
Uploader	The username of the uploader: Integer
Age	The number of days the video has been on Youtube:Integer
Category	The category under which the video falls into: String
Lenght	The duration of the video: Integer
Views	Number of views for the video: Integer
Rate	The rating the video has got:Float
Rating	The number of ratings the video has got:Integer
Comments	Number of comments for the video: Integer
Related IDs	Videos related to the particular video: String

We have a list of keywords which we have stored in a separate file which was gathered by analyzing various search feeds of different users. We then compare the video name with the list of keywords that we have. Later, we used the matched keyword as an input to get the search feeds that can be got for that particular search keyword.The manner in which this is accomplished is through the use of the data apis customized for youtube as stated above with specific error handling mechanisms just in case we need them to handle exceptions. The result of this search feed is given in the decreasing order of the views for that video. Thus, we suggest the top five videos that we get for the particular keyword to the user.

5. LESSONS LEARNT

Lessons learnt were having to deal with the google/youtube apis.Figuring out how to use them and where to start.It was a good learning experience with respect to having to deal with the feeds since the video recomender system retrieves outputs from certain feeds.Also with the data set that we had which consisted of usernames as one of the fields not all of the users had a playlist they just had a youtube account and that way we had a hard time dealing with the users.These were a few things we happened to learn during the course of our project

6. FUTURE WORK

As of now the video recommender system that we have built only works as a non-gui application more as a console or a command line interface application so in the future we would want to build a gui for this application since this is a video recommender system and this would provide a better interface for users to interact.We would want to come up with a better recommendation algorithm.Also come up with better mechanisms when it comes to recommending users the videos than the ones presented in the project specification.One other important factor that we considered was probably recommending users videos based on the user's location.Since location sensors play a huge part in social networking category.

7. REFERENCES

[1] Gdata-java-client. `http://gdata-java-client.googlecode.com//`.

[2] J. Davidson, B. Liebald, J. Liu, P. Nandy, and T. V. Vleet. The youtube video recommendation system. *ACM Trans. Program. Lang. Syst.*, pages 293–296, November 2010.

[3] R. Zhou, S. Khemmarat, and L. Gao. The impact on youtube video recommendation system on video views. *ACM Trans. Program. Lang. Syst.*, pages 404–410, September 2010.