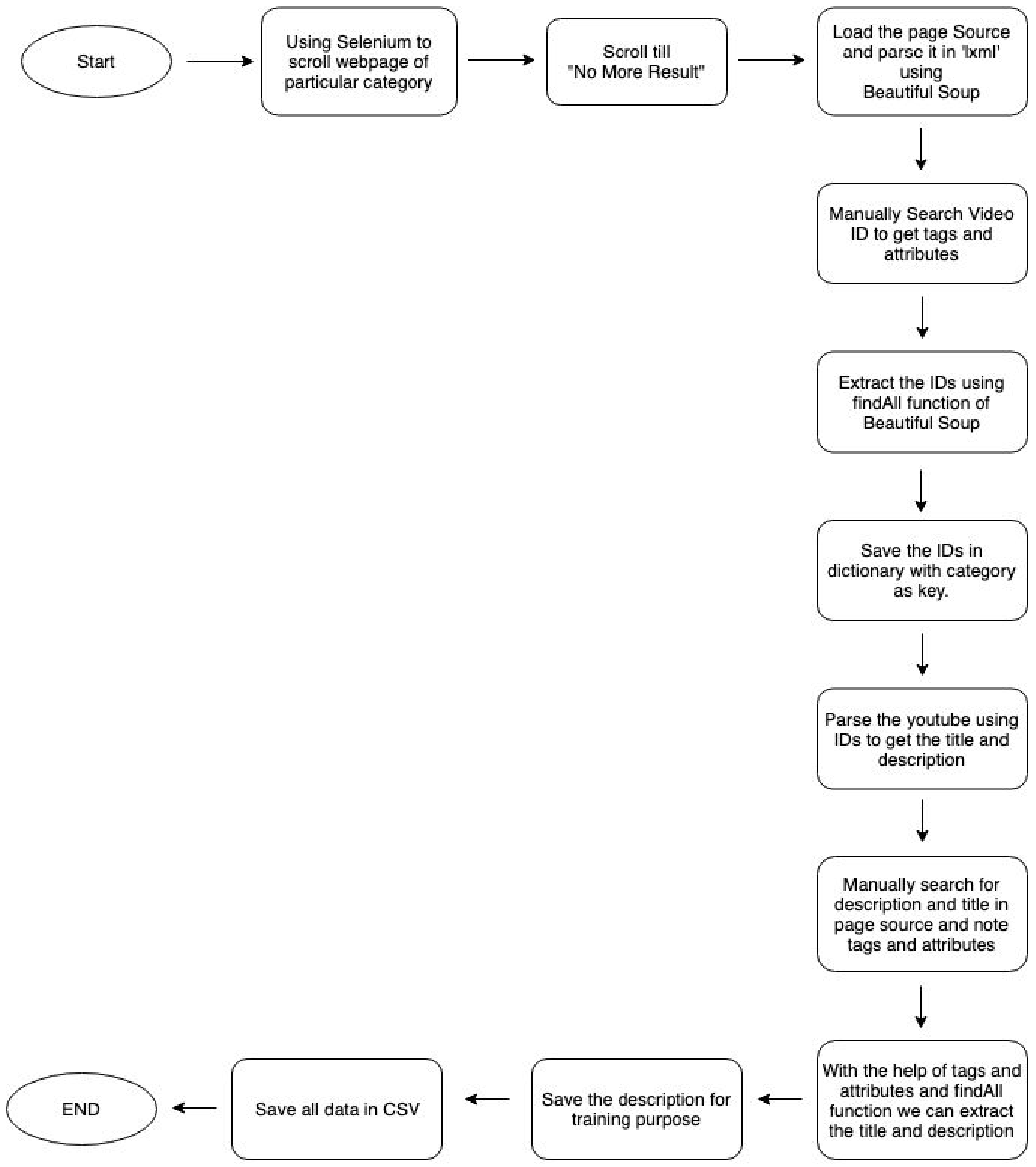
REPORT

1. Scrapping

First of all, before scrapping the website we have to understand the nature of the site, since youtube is an infinite scrolling website so our first approach was to decode the XAR and try to



Start
Using Selenium to scroll webpage of particular category
Scroll till "No More Result"
Load the page Source and parse it in 'lxml' using Beautiful Soup
Manually Search Video ID to get tags and attributes
Extract the IDs using findAll function of Beautiful Soup
Save the IDs in dictionary with category as key.
Parse the youtube using IDs to get the title and description
Manually search for description and title in page source and note tags and attributes
With the help of tags and attributes and findAll function we can extract the title and description
Save the description for training purpose
Save all data in CSV
END

Scraping youtube using Selenium and Beautiful Soup connect the links but youtube is using the tokens which generate at the time of loading the website and finding those tokens is a bit tricky so we are using Selenium for this project,

Selenium is a tool which can be very useful in scraping the infinite scroll pages, it actually opens the website using a web driver and has the ability to scroll in X and Y direction.

To scroll the pages in the Y direction, we need to find the height of the full page, which can be found by the documentElement.scrollHeight.

This method scrolls the website to the end and waits for a few seconds to load. when the website loaded completely then it again finds the height of full webpage and scrolls the website to the end of Untitled Diagram (2).pngthe page which loads more pages. This process repeats until the counter ends.

At some point in the loading, “no more result” come and further scrolling would be not possible.

Now we have all the page data, we can extract it as page source and use beautiful soup to dig deeper into this code.

To extract data from this code we used the findAll function of beautiful soup, it takes the tags or attributes to locate the specific line and then with then, we can extract the data using various operations.

To find the tags and attributes we find the keyword in page source using ctrl+F manually, From this, we only extracted the ids and saved them in a file for further use,

Now we again parse the youtube but this time we load the video page because the full description is available only of the individual video page

Now again we have to find the tags and attributes and then with the help of findALL function of Beautiful Soup we can find the specific line where our desired data is enclosed,

Now we save the description so that we can use it for training puprose.

In the end, we save it to the csv file.

2. Pre-processing

Description that we get from these youtube videos are full of irrelevant keywords but those keywords are very common so I created a stop words specific for these keywords which helps to eliminate them,

specific_stopwords=['subscribe','channel','http','https','like','dislike','please','click','notification','bell','credits','credit','link','support','donate','thanks','thank you','all','twitter','youtube','facebook','github','linkedin','insta','instagram','buy','must','watch','episode','part','series','lesson','download','video']

Apart from this, we can use the stemming, tokenizing and part of speech tagging.

3. Graphical reports are in the .png format and in the data folder itself.

4. My approach was to scrape the youtube from the scratch, I didn't use any google or youtube API to get any data, I use basic method to search for the attribute and tags and then get the data, This method gives me a freedom to tweak things easily. Now for this, I used Selenium and my main target was to get the Vid Ids because if I have the IDs then I can scrape the video page individually.

So once I got the Ids, I proceed to get the description and save each description separately under the category folder for further analysis but I have done my job using CSV file only.

Then with the help of pandas, I accessed the CSV file and take out the data and preprocessed it to use in the program.

Finally, the scraping part has a lot more room for improvement and can load a lot more files if we dig a little bit deeper.