ACADEMY OF ECONOMIC STUDIES,
FACULTY OF CYBERNETICS, STATISTICS
AND ECONOMIC INFORMATICS

BACHELOR'S THESIS

YouTube consumer behavior analysis application

Scientific coordinators:

Lect. univ. dr. *Toma Andrei*

Conf. univ. dr. *Aldea Anamaria*

Student:

Niculescu Ana-Maria

Bucharest

2017

CONTENTS

INTRODUCTION

The paper aims to describe YouTube E-motion application, a web application which performs sentiment analysis on YouTube comments related to controversial technologies using machine learning techniques and displays the results on a web page. It also proves that an analysis of the polarity of comments in order to identify their trends can provide a clear picture of the emergence of new technologies influence on potential user sentiments.

YouTube is one of the most influential media platforms in the modern society, a society which is highly dominated by internet speed and freedom of expression.

Starting with the moment of its emergence, in 2005, and until the present moment, in which YouTube is purchased by Google, the platform has continuously offered freedom of expression to any person who owns a channel and uploads videos, as well as to the persons who are watching the videos and communicate their opinions through likes, dislikes, comments, and most important, number of views. Having as a valid example YouTube phenomena, such as the hit Gagnam Style, by Psy, which, according to The Washington Post, is counting 2.85 billion views, we can easily conclude that a big part of the modern society's preferences and tendencies are mirrored by the reaction to such videos. Not only can YouTube serve as an entertainment medium, but also as a source of information and educational platform, given the fact that there are numerous channels which are constantly transmitting the latest news, offering trainings, inviting interested users to debate upon various subjects or revealing new innovations.

Given these circumstances, YouTube offers access to massive datasets which, when properly handled, can reveal public opinions and can determine the success of a product.

First of all, sentiment analysis on YouTube data can help businesses and organizations draw relevant conclusions about the consumer behavior and facilitate relevant predictions related to their actions and therefore acquire market intelligence.

Secondly, the data provided by YouTube comments represents a matter of interest for individuals who may buy various types of products or use services given the fact that they can scan and treat the comments as reviews and use the opinions of others in order to decide whether or not to trust the information presented in a certain video.

Finally, another application of this kind of analysis can be found in ad placement on media content such that an ad can be placed if one manifests a good opinion related to the content of a video, or an ad from a competitor can be placed if one expresses a bad opinion on a video.[1]

Taking into consideration subjects related to controversial technologies, such as Mars colonization, big data analysis or artificial intelligence, the need for such a tool increases given the fact that the evolution of these paradigm shifters raises ethical questions among potential future users.

By analyzing the sentiment of comments and their trending patterns, the application can offer a clear picture of overall user sentiment over a particular span of time. The application aims is to calculate and display the polarity of each comment using various classification techniques, to cluster videos based on the words used on comments and to perform an analysis based on word frequency.

[1] (Sentiment Analysis and Opinion Mining, 2012)

Related Work

Several researchers and natural language processing enthusiasts have performed sentiment analysis on social networks, such as YouTube or Twitter. Among these projects, the closest related to the presented application is a platform developed by Siersdorfer, which analyzed more than 6 million comments mined from 67000 videos in order to realize the identification of a possible relationship between comments, ratings, views and topic categories.[2]

Sentiment analysis performed on Internet Movie Database (IMDb) has also been performed, based on 2053 movie reviews. The work of Pang, Lee and Vaithyanathan depicts the fact that standard machine learning techniques outperform in the end manual classification techniques that require human intervention.

Another work in this field is made by Bollen et al. The researchers performed an analysis on Twitter feeds of users in order to predict daily changes to the closing values represented by Dow Jones Industrial Average (DJIA) Index, with a reported accuracy of 86.7%.[3]

In their article, Kilian Thiel, Tobias Kotter, Rosaria Silipo and Phil Winters, describe the development of a similar application, having as main purpose the creation of user profiles on Slashdot based on their comments. Slashdot is a website that mostly posts news related to technological subjects and allows readers to express their opinions through comments.[4]

There is also YouTube Analytics, which is a performance monitor for a channel with up-to-date metrics and reports. It provides a large amount of data available in different reports, like the Watch time, Traffic sources, and Demographics reports in the mean average percentage error.

Unlike the above mentioned projects, the application focuses on opinion mining regarding 8 specific categories of controversial technologies, which became more sustainable over time and promise to change our perspectives with respect to transportation, energy consumption or even offer a possible alternative to living on Earth. These categories of technologies are: electric cars, solar energy, artificial intelligence, Mars colonization, cloud computing, big data analysis and bionic limbs.

[2] (Sentiment Analysis and Opinion Mining, 2012)
[3] (Sentiment Analysis and Opinion Mining, 2012)
[4] (Kilian Thiel, 2016)

Methodology

In order to achieve the purpose of the application, the following steps have been followed:

First of all, the connection to YouTube has been established through an **API key**, in order to request the necessary data, represented by video details, such as id, title, number of views, number of likes and number of dislikes. Data and metadata about the comments related to each video has been extracted: comment text, publication date, author and votes up. The data obtained through the **HTTP request** has been stored in a **Mongo database** containing eight collections corresponding to the above mentioned subjects of interest.

Secondly, the data gathered has been prepared in order to be analyzed, taking into account limitations such as meaningless words or words in other languages than English. After the comment text has been refined and added to proper data structures, different types of analysis have been performed. Machine learning techniques have been used in order to put into perspective user sentiments regarding each of the selected subjects. The application uses **supervised learning techniques** as well **as unsupervised learning techniques**, aiming to finally display a report regarding public opinion on single page application.

Finally, the report is displayed on an interactive web page. The detailed approach regarding each one of these steps will be presented in the following chapters.

Paper structure

In the following pages, different aspects regarding the structure of the project will be detailed

The first chapter covers theoretical aspects of the project represented by the introduction of the technologies used in order to develop the application alongside with the argumentation of choice.

The second chapter focuses on the presentation of the machine learning techniques used in order to perform sentiment analysis and general statistics on YouTube data as well as the techniques used in order to extract data and store it in a local database.

The third chapter presents a better overview of the user interface functionality, by introducing screenshots the application.

Finally, the last chapter will serve as a conclusion and will open the discussion regarding further implementations and possible improvements.

From a philosophical point of view, a challenge within machine learning is to build intelligent machine, and a big part of machine intelligence is due to understanding human language. For a long period of time, the achievement of this purpose was perceived as being too difficult. However, over the past years Natural Language Processing technologies have grown, and the possibility of human language understanding has increased.[5]

[5] (Learning to Generate Reviews and Discovering Sentiment, 2017)

1. TECHNOLOGIES USED

In the current chapter, the technologies used for the implementation of a Single Page Application with a micro-server back-end alongside the tools used in order to perform data analysis, static persistence of data and version control are presented.

1.1 Single Page Application

A Single Page Application or SPA is a web application which provides a user interface similar to a desktop application. In such an application, the necessary resources are loaded dynamically as a response to the actions of the user. The interaction with an SPA requires the communication with a web server.[6]

In a web application, the client side (HTML, CSS, Angular) takes care of the presentation of information, while the server side (Python) takes care of information access (from Mongo Database) and processing (using machine learning techniques).

1.2 Python 3

Python is a high-level programming language, developed by Guido van Rossum and released for the first time in 1991. Its history dates back to late 1980's, when Guido van Rossum started the implementation at Centrum Wiskunde & Informarica in Netherlends of the programming language that was soon to be called the successor of ABC programming language.

According to Wikipedia, in October 2000, Python 2.0 was released, adding new features, such as garbage collector and Unicode support. Starting with this release, the development process began to become open to community and more transparent.[7]

In December 2008 Python 3 was introduced, representing a backwards-incompatible release.[8] This means that Python 2 code may not function properly under Python 3.

Python's greatest advantage is code readability, being often described as "executable pseudo-code", having a syntax that helps programmers to make use of various concepts in fewer lines of code, in comparison to other programming languages, such as Java or C++,

[6] https://en.wikipedia.org/wiki/Single-page_application
[7] https://en.wikipedia.org/wiki/Python_(programming_language)
[8] https://en.wikipedia.org/wiki/Python_(programming_language)

Python enables rapid development. Therefore, Python is concise and also very easy to read.[9]

Being a dynamic language, it increases the speed of development, by allowing the introduction of new attributes to objects on the fly, and allowing the dynamic typing of variables.

Python can be regarded from three points of view, given the fact that it is a multi-paradigm programming language. It supports different styles of programming:

1) Object oriented programming
2) Procedural programming
3) Functional programming.[10]

Taking into consideration the fact that a large part of the current application makes use of machine learning techniques, and the other part focuses on single page application development, this aspect of Python is especially important. Machine-learning algorithms can have a high degree of variation and in different contexts it is better to approach different paradigms.

Besides, Python introduces a wide standard library, including modules for numerical processing, web connectivity and also graphical programming. [11] The set of tools can be easily extended with libraries dedicated to natural language processing (nltk), data analysis (sci-kit learn, pandas, numpy), graphical representation (PIL, matplotlib), database connectivity (pymongo) and integrates frameworks for web development, such as Flask and Django. Thus, Python easily provides access to an integrated framework for multiple application development purposes.

Another advantage of this programming language is the fact that it is free, multiplatform and has a large community of developers ready to offer support regarding different areas of expertise related to Python programming.

The above mentioned facts represent the reasons why Python was chosen for the development of YouTube E-motion application.

[9] (Segaran, 2007)
[10] (Segaran, 2007)
[11] (Segaran, 2007)

Data analysis tools

1.3 Natural Language Processing Toolkit

According to Steven Bird, Ewan Klein and Edward Loper, Natural Language Processing represents in wide sense the computer manipulation of human language.[12]

Natural Language Processing toolkit, or NLTK is a Python infrastructure, specialized in the elaboration of Natural Language Processing programs. It was developed in 2001, at the University of Pennsylvania, and was continuously expanded by contributors over time. Now it serves as the main tool for different research projects.[13]

This toolkit provides access to over 50 lexical resources and corpora alongside with a large variety of text processing libraries for tasks like: tokenization, parsing, semantic reasoning, tagging and classification.[14] Some of the modules specialized in performing these type of tasks are: nltk.corpus , nltk.tokenize, nlt.stem, nltk.collocations, nltk.tag, nltk.classify, nltk.metrix and nltk.probablity.

According to Steven Bird, the main goals of nltk are the following:

- Modularity: made of components which can be independently used
- Extensibility: alternative implementations can easily be included
- Consistency: consistent data structures and intuitive method names
- Simplicity: hides the underlying complexity and provides practical means of performing NLP tasks.

However, one of the possible disadvantages of this tool is that becaus it uses complex algorithms in order to perform meaningful tasks, it is not very well optimized to have runtime performance.[15]

Given the fact that the comments extracted from YouTube platform represent human language data, nltk provides the ideal set of functions which ease the elaboration of specific analyses related to the project purpose.

[12] (Steven Bird, 2009)
[13] (Steven Bird, 2009)
[14] http://www.nltk.org/
[15] (Steven Bird, 2009)

1.4 Scikit-learn

Scikit-learn is a free Python library released initially in 2007.The library is designed to perform supervised and unsupervised machine-learning tasks, such as: clustering, regression or classification, being also able to operate alongside other Python scientific libraries, like SciPy and NumPy.[16]

According an article published in 2011 in the Journal of Machine Learning Research, the project vision is characterized by:[17]

- Code quality, providing solid implementations of complex algorithms
- BSD licensing, imposing few restrictions on the use and redistribution of software[18]
- Bare-bone design and API: framework code is avoided
- Community-driven development: the code is open-source
- Documentation: provides a large amount of documentation resources alongside real-world applications

Scikit-learn offers a large variety of machine learning algorithms, enabling comparisons between different methods used for a certain applications.

Taking into account the functionality of YouTube E-motion application, this tool is mainly used in order to perform sentiment analysis on comment for polarity detection. This methodology will be detailed in the following chapters.

1.5 YouTube Data API

According to Wikipedia, an Application Programming Interface (API) represents a group of tools that can be used in order to build application software. From a Web Development perspective, the API defines a set of HTTP request messages which expects to receive in return data in an Extensible Markup Language (XML) format or JavaScript Object Notation (JSON) format.[19]

[16] https://en.wikipedia.org/wiki/Scikit-learn
[17] (Scikit-learn: Machine Learning in Python, 2011)
[18] https://en.wikipedia.org/wiki/BSD_licenses
[19] https://en.wikipedia.org/wiki/Application_programming_interface

The YouTube Data API is allowing the incorporation of functions normally performed on YouTube platform into another application. This way, information regarding YouTube videos, channels or playlists can be extracted. The data is represented as JSON objects. In this case, data related to YouTube videos (number of views, number of likes, number of dislikes, title and id) and comments (comment text, publication date) has been requested.

The application uses oauth2client library in order to follow the OAuth 2.0. authentication protocol.[20]

Web development tools

1.6 Flask

Flask represents a micro web framework developed by Armin Ronacher in Python and initially released in 2010. [21]

Flask gained the "micro" attribute given the fact that the platform intends to keep the core of the framework simple yet extensible. As a general rule, Flask doesn't contain form validation or database abstraction layer, leaving these kind of choices at the latitude of the developer. This aspect makes Flask a highly customizable utility, enabling the developer to make design decision best suited for a particular project.[22]

Taking into account its prototyping speed and flexibility, Flask Framework was chosen as a web-development framework for YouTube E-motion application.

1.7 HyperText Markup Language (HTML)

HTML, or HyperText Markup Language is the mark-up language that is used to describe the structure of web pages semantically, and uses tags in order to indicate the way information is rendered. The browser is capable of interpreting the tags by receiving the HTML document from a web server or from the localhost and represents them in the form of multimedia web pages.[23]

[20] https://developers.google.com/youtube/v3/docs/
[21] https://en.wikipedia.org/wiki/Flask_(web_framework)
[22] http://flask.pocoo.org/docs/0.12/
[23] https://en.wikipedia.org/wiki/HTML

HTML allows multimedia content or other objects to be embedded into the web page by making use of tags, marked by angle brackets. It provides the ways to create structured documents because it denotes semantics for text, like paragraphs, lists, headings or quotes.

HTML has also the ability to embed programs written in scripting languages, which can influence the behavior and the way content is rendered in a page.

The HTML standards are held by World Wide Web Consortium (W3C), an organization which has as main purpose the development of guidelines and protocols which ensure the evolution of the Web on a long term scale.[24]

In October 2014, HTML 5 was released, introducing new elements, new input types, new attribute syntax and supporting a large variety of media elements, such as sound content, video content and external application containers. It also allows drawing graphics on the fly using canvas or drawing scalable vector graphics (SVG).[25]

1.8 Cascading Style Sheets (CSS)

Cascading Style Sheets is a mechanism used in order to style the elements which are rendered in a web page. It also accomplishes the purpose of separating structure (described by HTML) from presentation.

The main advantage of CSS is the fact that it decreases the amount of work, because it is able to control simultaneously the layout of multiple web pages and it can be reused.[26]

CSS uses rule sets, which are formed by a selector that points to the targeted HTML element, and a declaration block containing key-value pairs, represented by property name and value. Some of the properties which can be customized are: position, float, display, color and clear.

[24] https://en.wikipedia.org/wiki/World_Wide_Web_Consortium
[25] https://www.w3schools.com/ht0)ml/html5_new_elements.asp
[26] https://www.w3schools.com/css/css_intro.asp

1.9 Bootstrap

Bootstrap represents an open-source web framework built for front-end, having as main purpose the design of web applications and websites. It was initially released in 2011, being originally named Twitter BluePrint and was meant to serve as a Twitter framework.[27]

The templates provided by Bootstrap are HTML and CSS based for various elements, such as forms, typography or navigation.

Bootstrap contains major types of components:

- Style sheets, providing modern appearance for HTML elements
- Re-usable components: CSS classes applied to some HTML elements
- JavaScript components, offering additional user interface elements

1.10 AngularJS

AngularJS represents a JavaScript-based front-end web application framework which is open-source. It was initially released in 2010. Its main target is to simplify the development and also the testing of web applications and cross-platform mobile applications, which may follow model-view-controller (MVC) and model-view-model (MVVM) architectures. [28]

AngularJS framework is used in order to extend the traditional HTML by presenting dynamic content on a page with the help of two-way-data-binding mechanism, which enables the synchronization of views and models.

AngularJS intends to:

- Separate DOM manipulation from the application logic
- Separate the client side from the server side of the application, allowing parallel development
- Provides structure to the application development process

[27] https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)
[28] https://en.wikipedia.org/wiki/AngularJS

1.11 Data Persistency: MongoDB

MongoDB was first released in 2009 and it is an open-source NoSQL database, being able to provide high availability, high performance and automatic scaling.

Unlike relational (SQL) databases, which use tabular relations, NoSQL databases ensure simplicity of design and ease the “horizontal” scaling to clusters of machine (which means adding more machines into a pool of resources) , given the fact that NoSQL databases store data differently than relational databases. The trade related to this database approach is the fact that consistency is compromised in favor of speed and availability.[29]

MongoDB stores the records in a structure similar to JSON objects, whose field values may include other documents or document arrays. Using document structure provides the following advantages:[30]

- Polymorphism supported by a dynamic schema
- The need for joins is reduced by using embedded documents
- The objects correspond to native data types used in programming languages

The documents are stored in collections-which correspond to tables in relational databases- and the collections are stored in Mongo databases.[31]

Taking into account the fact that the current project makes use of large sets of data, which have to be accessed as fast and efficiently as possible, MongoDB has been considered to be the optimal approach.

In order to ensure the interaction of Python scripts with the database, pymongo distribution was used.

1.12 PyCharm IDE

PyCharm is an integrated Development Environment produced by JetBrains and first released in 2010, which is used for development of applications in Python. It provides a large variety of utilities, such as integrated unit tester, code analysis, debugger and integration of web development frameworks.[32]

[29] https://en.wikipedia.org/wiki/NoSQL
[30] https://docs.mongodb.com/manual/introduction/
[31] https://docs.mongodb.com/manual/core/databases-and-collections/
[32] https://en.wikipedia.org/wiki/PyCharm

PyCharm incorporates an intelligent code editor, which introduces error detection and on the fly code fixes, eases the navigation of the files within the project and provides fast and safe refractor features which support project-wide changes.

The IDE also represents a unified UI for working with version control systems, such as Git, enabling the configuration of automatic deployment to a remote host or virtual machines. [33]

In addition to Python, the IDE offers firs-class support for various development frameworks, such as Django, Flask or Pyramid and specific template languages (JavaScript, TypeScript, CoffeeScript, AngularJS, Node.js and HTML/CSS).

1.13 Git

Git is an open-source system dedicated to distributed version control. Git encourages the existence of a big number of local branches that don't depend on each other, their manipulation being very fast. [34] The main advantage of distributed systems is the fact that the entire repository (project) is cloned on the local machine and the changes are uploaded on the server when the programmer decides. The distributed model also presents an increased speed and ensures the possibility of working offline. Other notable advantages of Git are:

- Branching and merging are easy and fast, meaning that the programmer can sandbox the features and ideas until he decides to upload them on server
- The workflow presents a high flexibility
- Integrity of data is assured given the fact that Git uses SHA1 trees and data corruption sources can be easily detected, therefore Git serves as a back-up solution

[33] https://www.jetbrains.com/pycharm/features/
[34] https://git-scm.com/about

2. APPLICATION DEVELOPMENT APPROACH

2.1 Data gathering

The starting point in the development of YouTube E-Motion application is data gathering. In order to extract YouTube data, YouTube API tool was used to send HTTP requests to YouTube and to receive information. According to the blog post named *Mining YouTube using Python*, posted by Kunal Jain on www.analyticsvidhya.com, the following steps are required in order to achieve this goal in a Python environment:[35]

The first step in order to gain access to data was to **request an API key from Google Developer Console**, by activating an account on this platform. After that, a new project and an API key were created and activated.

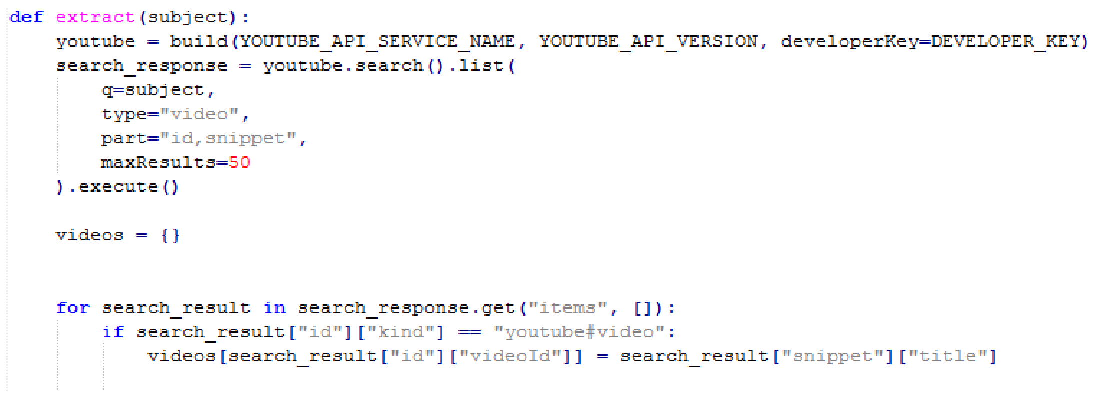
In order to ensure the interaction of Python with the YouTube API, google-python-api-client and ouauth2client libraries have been imported. The results are further read into a nested dictionary, or associative array. It represents a data type characterized by an unordered set of key-value pairs, the keys being always unique.[36] The second step was setting up the YouTube search parameters. This implies the introduction of a developer key variable, corresponding to the one generated by Google API, and other variables which retain the values corresponding to username, password, API Service Name and API version that will be further used in order to make the request.

The third step was to perform YouTube search. I order to do so, the *extract* function was created. The function takes as argument a keyword which serves as a filter for YouTube requests. Keywords related to artificial intelligence, 3D printing, cloud computing, big data, Mars colonization, electric cars and solar energy have been passed as arguments.

---

[35] https://www.analyticsvidhya.com/blog/2014/09/mining-youtube-python-social-media-analysis/

[36] https://docs.python.org/2/tutorial/datastructures.html#dictionaries

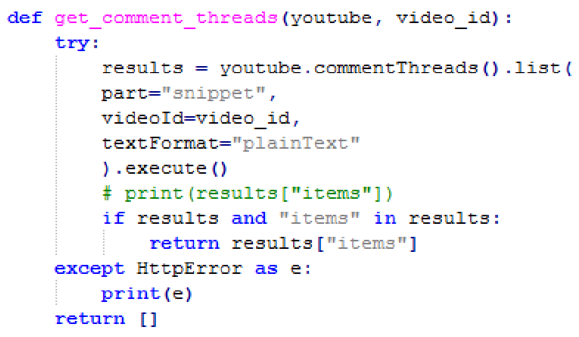


```
def extract(subject):
    youtube = build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION, developerKey=DEVELOPER_KEY)
    search_response = youtube.search().list(
        q=subject,
        type="video",
        part="id,snippet",
        maxResults=50
    ).execute()

    videos = {}


    for search_result in search_response.get("items", []):
        if search_result["id"]["kind"] == "youtube#video":
            videos[search_result["id"]["videoId"]] = search_result["snippet"]["title"]
```

Figure 2.1.1: Definition of extract function

The code from Figure 2.1.1 represents the definition of *extract* function. The build method imported from google-api-python-client library takes care of the YouTube API connection allowing to later create a list containing these results. The list method retrieves results that match the specified query term, which in our case is the subject that is passed as an argument. Then each result is added to the appropriate list. Taking into account the YouTube API limit, data about maximum 50 videos can be requested at a time. By using this method we only extract general data about the videos, such as video id, video title, number of views, number of likes, number of dislikes and the number of comments.

In order to **extract data related to comments** for each video, the function *get_comment_threads* is used. The function is taking as parameters the results from the build method mentioned earlier and also the id of the video for which the comment extraction is intended. The **try catch mechanism** handles the case in which either the video has deactivated comments or the case in which the video has no comments, therefore there is no information to be extracted.



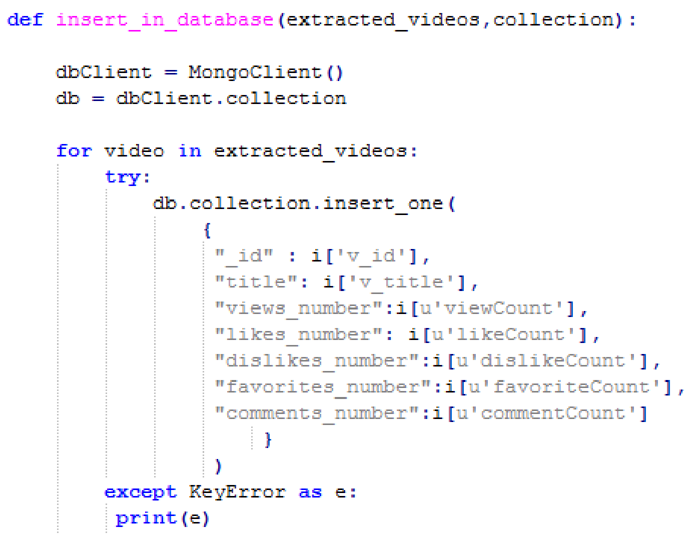
```
def get_comment_threads(youtube, video_id):
    try:
        results = youtube.commentThreads().list(
        part="snippet",
        videoId=video_id,
        textFormat="plainText"
        ).execute()
        # print(results["items"])
        if results and "items" in results:
            return results["items"]
    except HttpError as e:
        print(e)
    return []
```

Figure 2.1.2: Definition of get_comment_threads function

2.2 Database Interaction

2.2.1 Inserting data

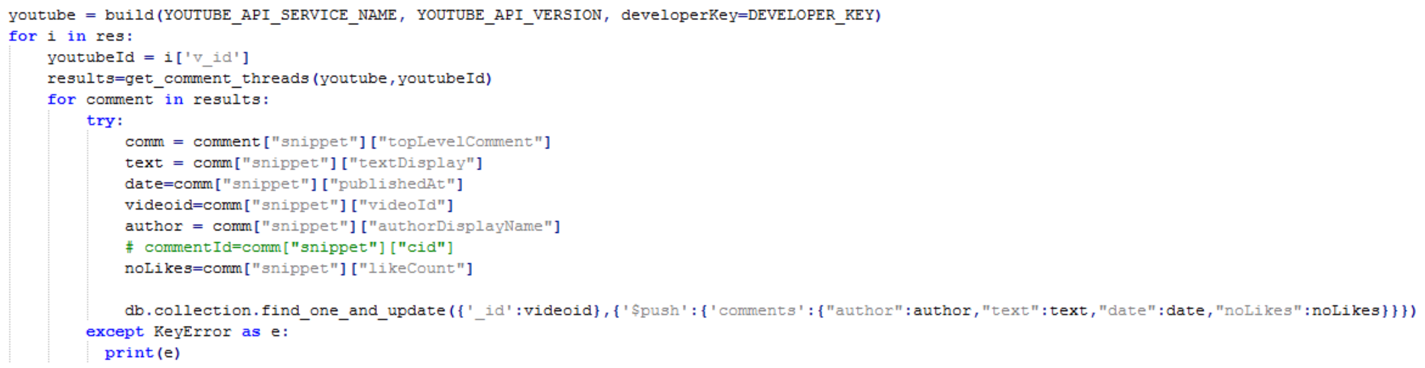
The database used for this application is called *LatestTechnologies* and consists of eight collections named after the subjects of interest mentioned above. Each collection has the same structure, so the insert method represents a template for all collections.



```
def insert_in_database(extracted_videos,collection):

    dbClient = MongoClient()
    db = dbClient.collection

    for video in extracted_videos:
        try:
            db.collection.insert_one(
                {
                 "_id" : i['v_id'],
                 "title": i['v_title'],
                 "views_number":i[u'viewCount'],
                 "likes_number": i[u'likeCount'],
                 "dislikes_number":i[u'dislikeCount'],
                 "favorites_number":i[u'favoriteCount'],
                 "comments_number":i[u'commentCount']
                     }
                )
        except KeyError as e:
         print(e)
```

Figure 2.2.1: Description of insert_in_database function

In order to update the collection with data about comments, the function iterates through all records and calls the *get_comment_threads* function. When a returned value for the comments contains a video id identical to the one belonging to a ceratin video in the collection, data about the comments is inserted into the matching document.



```
youtube = build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION, developerKey=DEVELOPER_KEY)
for i in res:
    youtubeId = i['v_id']
    results=get_comment_threads(youtube,youtubeId)
    for comment in results:
        try:
            comm = comment["snippet"]["topLevelComment"]
            text = comm["snippet"]["textDisplay"]
            date=comm["snippet"]["publishedAt"]
            videoid=comm["snippet"]["videoId"]
            author = comm["snippet"]["authorDisplayName"]
            # commentId=comm["snippet"]["cid"]
            noLikes=comm["snippet"]["likeCount"]

            db.collection.find_one_and_update({'_id':videoid},{'$push':{'comments':{"author":author,"text":text,"date":date,"noLikes":noLikes}}})
        except KeyError as e:
         print(e)
```

Figure 2.2.2: Comment update in database

2.3 Data refining

Taking into account the fact that the data used in order to develop the application is acquired from Web, it is impossible to control the language in which YouTube users manifest their opinions, the characters -accented characters may be used, as well as characters outside the Latin alphabet- and also the uncensored expressions.

However, the uncensored expressions have not been filtered out, because they may highly influence the meaning of comments (for example the degree of negativity).

In order to perform different types of analysis, especially regarding natural language processing of comments, we have to first take into account various limitations that natural language can impose on data analysis. This limitations may represent the main reason why Python may produce confusing output working with a non-refined data set.

A possible limitation is represented by stopwords[37].As they are described in the book written by Steven Bird et al, stopwords represent high frequency words, for example “and”, ”a”, ”the”, “to”, that sometimes are better to be filtered out of the data set before performing further processing. Given the fact that stopwords don’t have a great lexical content they may cause problems related to frequency distribution of words in a comment.

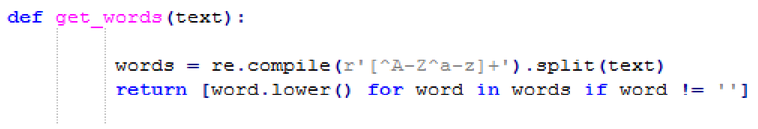
In order to eliminate the stopwords, stopwords corpus from nltk was imported and a variable representing the set of stopwords has been created.

Another limitation is the spoken language. Although nltk includes so-called Swadesh wordlists, which contain approximately 200 common words in languages other than English, as well as other similar tools, in order to make sure that the analysis is correct, we will only use English comments. Besides, the training dataset used for polarity detection contains only English comments and reviews. In order to filter only English words, a dictionary was imported and each word was checked to see if it belongs to the English dictionary.

The script *refine_comments* contains functions which retrieve the comment text according to further necessities of data analysis approaches. The functions query the database and return the comments in the form of word list, making use of *get_words* method that splits the text formed by all comments according to a regular expression in order to separate the words.

[37] (Steven Bird, 2009)

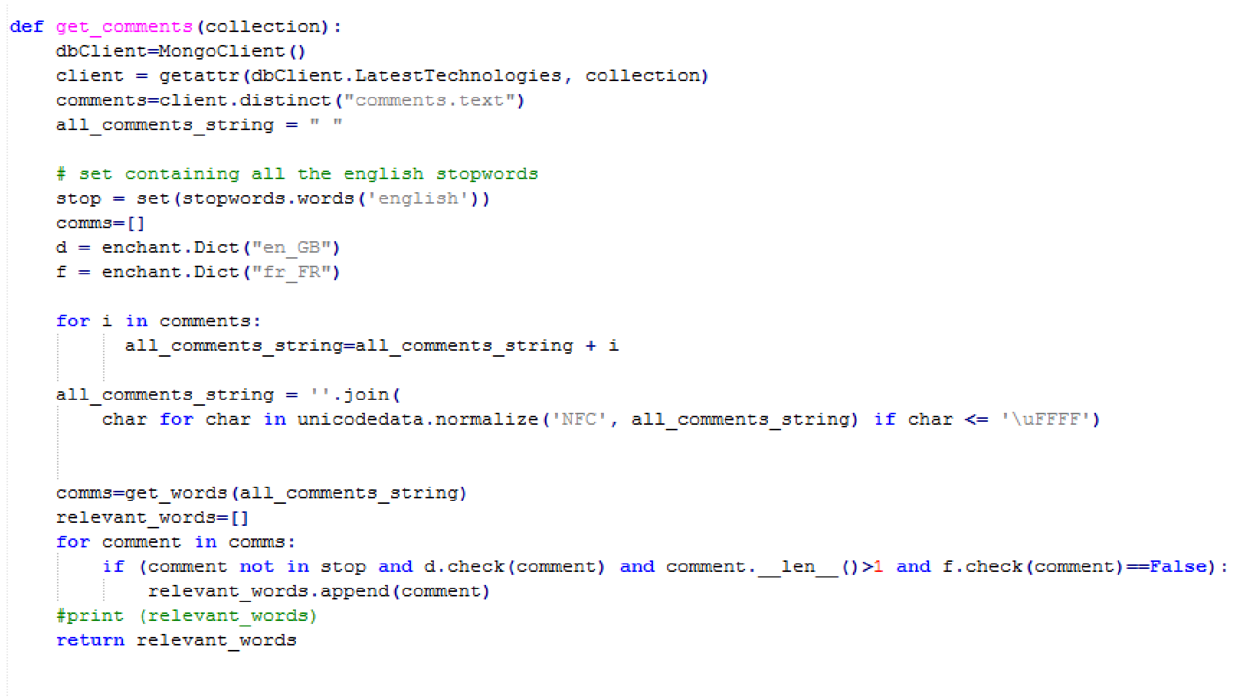
The definition of the *get_words* function is presented in Figure 2.3.1:



```
def get_words(text):

        words = re.compile(r'[^A-Z^a-z]+').split(text)
        return [word.lower() for word in words if word != '']
```

Figure 2.3.1 Definition if get_words function

The function which is querying the database, makes use *of get_words* and filters the words according to the above mentioned criteria is *get_comments* function, presented below:

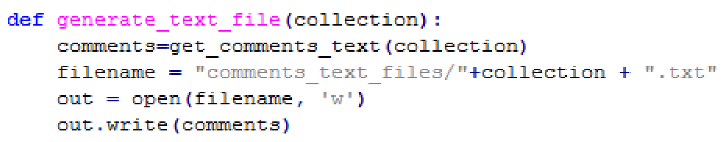


```
def get_comments(collection):
    dbClient=MongoClient()
    client = getattr(dbClient.LatestTechnologies, collection)
    comments=client.distinct("comments.text")
    all_comments_string = " "

    # set containing all the english stopwords
    stop = set(stopwords.words('english'))
    comms=[]
    d = enchant.Dict("en_GB")
    f = enchant.Dict("fr_FR")

    for i in comments:
        all_comments_string=all_comments_string + i

    all_comments_string = ''.join(
        char for char in unicodedata.normalize('NFC', all_comments_string) if char <= '\uFFFF')


    comms=get_words(all_comments_string)
    relevant_words=[]
    for comment in comms:
        if (comment not in stop and d.check(comment) and comment.__len__()>1 and f.check(comment)==False):
            relevant_words.append(comment)
    #print (relevant_words)
    return relevant_words
```

Figure 2.3.2: Definition of get_comments function

The function *get_comment_text* does the same thing *as get_comments* with the difference that *get_comment_text* returns the text of the comments from the collection passed as parameter under the form of a string.

The function *generate_text_file* makes use of the *get_comment_text* for dumping the comments for the specified collection in a text file having the same name as the collection. The definition *of generate_text_file* method is in Figure 2.3.3:



```
def generate_text_file(collection):
    comments=get_comments_text(collection)
    filename = "comments_text_files/"+collection + ".txt"
    out = open(filename, 'w')
    out.write(comments)
```

Figure 2.3.3 Definition of generate_text_file functio

2.4 Data Analysis

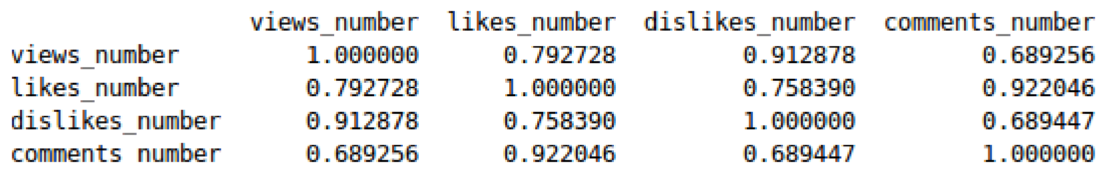
2.4.1 Descriptive statistics on video information

For each category of videos, a set of descriptive statistics has been computed to offer a better overview of the videos presented in each collection. Having this in mind, the function *compute_descriptive_stiatistics*, which takes as argument the collection name, was created. The function makes use of Pandas library in order to compute the value of indicators such as: number of records, mean, standard deviation, minimum value of views and maximum values of views. [38]

2.4.2 Correlation matrix

In order to determine whether or not the number of views, number of likes, number of dislikes and number of comments influence each other, the correlation matrix for each collection has been computed using Pandas. Figure 2.5.1 shows the output of the correlation function applied to Artificial Intelligence collection. As it can be noticed in the figure, the correlation between views number and likes number is relatively high (approximately 79%), while the correlation between views number and dislikes number is even higher (approximately 91%), indicating that there may exist a negative perception of this subject.

The fact that there is a positive correlation between all four indicators can be also noted.



```
                 views_number  likes_number  dislikes_number  comments_number
views_number         1.000000      0.792728         0.912878         0.689256
likes_number         0.792728      1.000000         0.758390         0.922046
dislikes_number      0.912878      0.758390         1.000000         0.689447
comments_number      0.689256      0.922046         0.689447         1.000000
```

Figure 2.5.1: Correlation matrix of Artificial Intelligence collection

[38] https://chrisalbon.com/python/pandas_dataframe_descriptive_stats.html

2.4.3 Statistics on comments

In order to gather more information regarding the language used by YouTube users while expressing their opinions, several statistics on comments have been performed with the help of NLTK.

1) Plotting the word frequencies

The frequency distribution of words is computed with the help of nltk.probability library and is used to count the number of appearances of each word in the comments corresponding to one of the eight subjects. Technically, the frequency distribution is defined as a function whose purpose is to map for each sample, in our case comment text, the number of times a word occurred.[39] By plotting the frequencies of the most encountered 50 words we can find out the most informative words and also get an insight of the overall reaction to the subject. As an example, if the most used words are negative, we can conclude that in general, the public presents a negative reaction towards that subject.

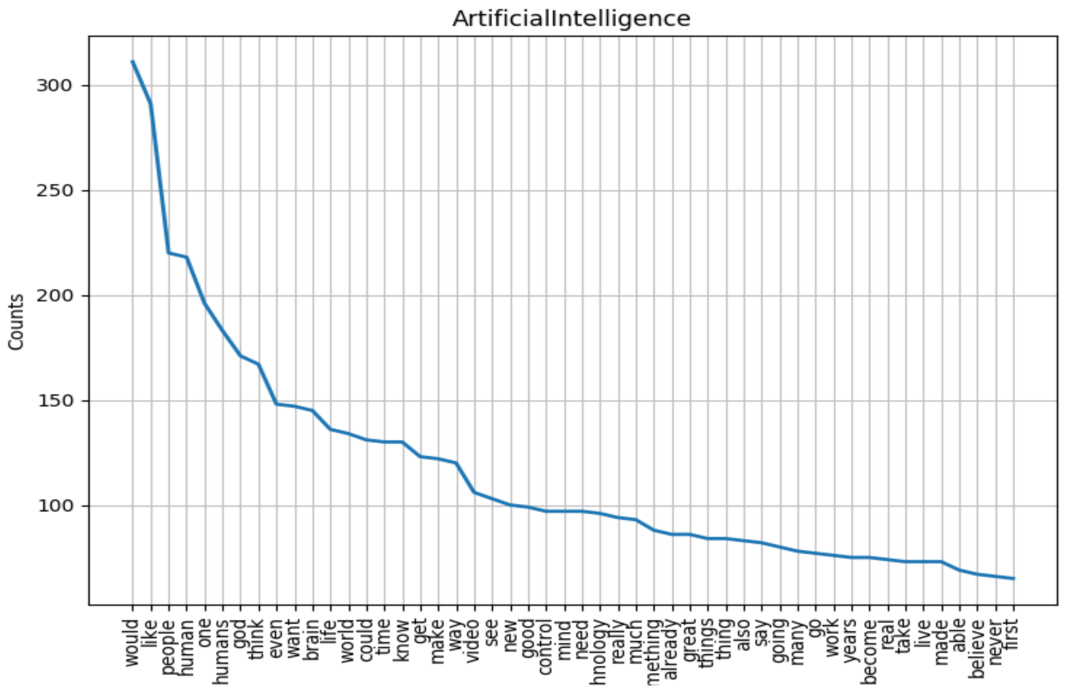


Figure 2.6.1: Frequency distribution plot for Artificial Intelligence

[39] http://www.nltk.org/_modules/nltk/probability.html

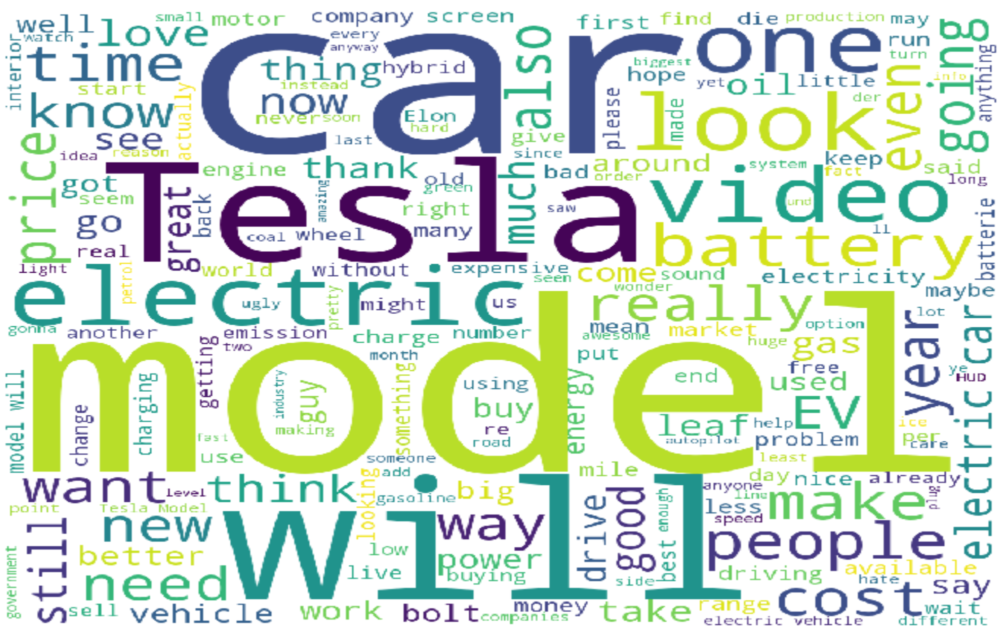


Figure 2.6.2: Word cloud of Electric cars collection based on frequency distribution of words

Figure 2.6.1 represents the frequency distribution plot of words used in comments related to artificial intelligence. As it can be noticed, words like “human”, “people”, ”brain” or “life” are among the most 50 used words in comments.

Figure 2.6.2 represents a more artistic way of displaying the most frequent words used in comments related to electric cars with the help of a word cloud.

2) Plotting cumulative frequency

The cumulative frequency puts into perspective the frequency distribution plot, showing how much of the entire text formed by comments is taken up by the most frequent 50 words. [40] The cumulative frequency plot is computed using *plot_cumulative_frequency* function. A plot representing the output of the function applied to “SolarEnergy” collection is presented below.

[40] (Steven Bird, 2009)

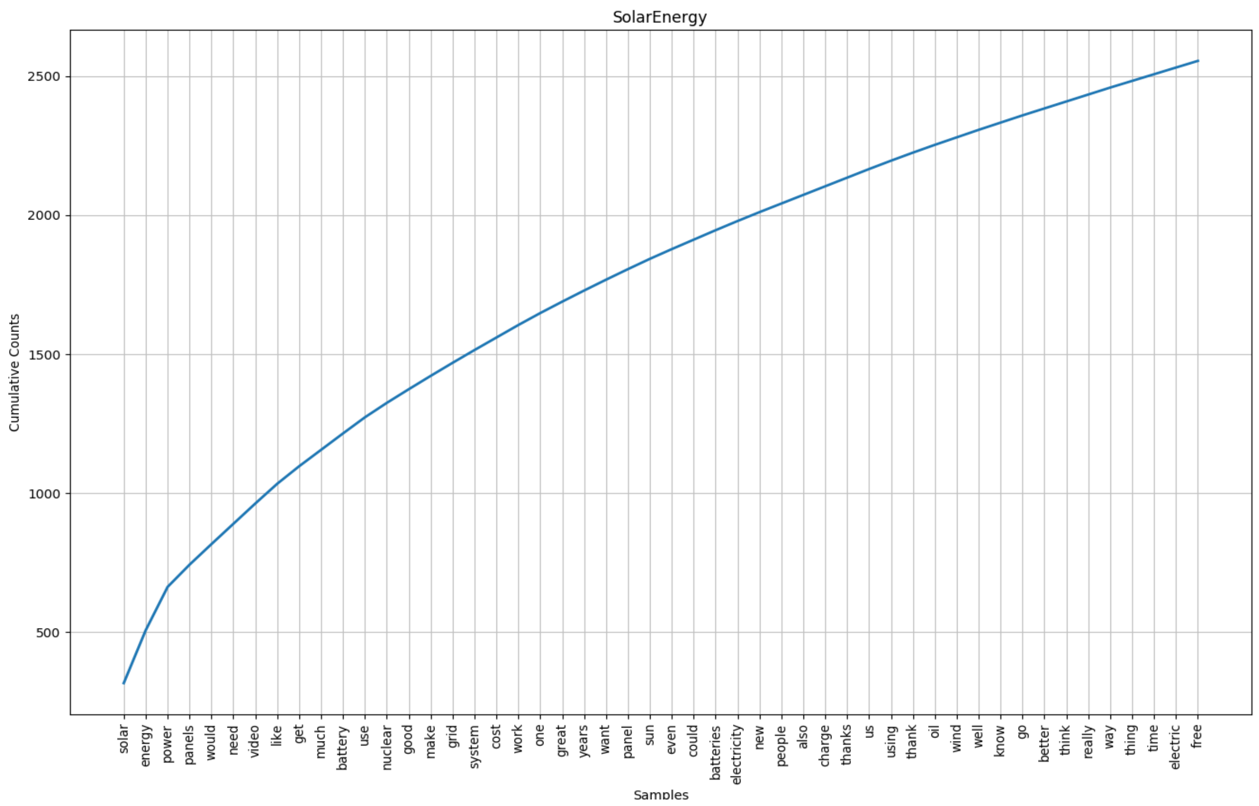


Figure 2.6.3: Cumulative frequency plot for comments related to solar energy

As it can be observed in the graph presented in Figure 2.6.3, the biggest number of the words which form the comments related to solar energy, is represented by the word "solar" (about 250 appearances). Also, it can be noticed that 2500 of the words used in all the comments corresponding to solar energy are actually the most frequent 50 words.

3) Dispersion plot

The dispersion plot is used to determine the location of the most frequent words in the text, in other words, to present density of a particular word in a portion of the text. In order to draw more relevant conclusions, the comments were chronologically ordered before making the necessary computations for creating the dispersion plot.

4) Average number of words per comment

The average number of words per comment is used in order to reflect how many words are used in average in order to express an opinion. Taking into account the fact that the video subjects encourage debates, this indicator can also offer an insight on how well documented the opinions are.

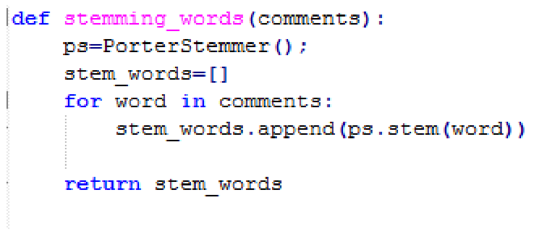
5) Collocations frequency

A collocation is a meaningful sequence of words that appear grouped very often. For example, “red wine” represents a collocation, but “the wine” is not. Another distinctive sign of a collocation is the fact that they cannot be substituted with synonyms.

In order to obtain a list of collocation from the text, we must first extract the bigrams from the text, which are simple pair of words. After creating a list of bigrams, we obtain the collocation by extracting the most frequent bigrams. [41]

6) Stemming words

Stems represent root words whose use may offer different variations. However the meaning of the text may not be changed according to these variations exposing the analysis to the risk of redundancy.[42] In order to eliminate this redundancy and obtain the most used root words, *stemming_words* function has been created. The body of the function is presented below:



```
def stemming_words(comments):
    ps=PorterStemmer();
    stem_words=[]
    for word in comments:
        stem_words.append(ps.stem(word))

    return stem_words
```

Figure 2.6.4: Definition of stemming_words function

As it can be noticed, the function receives as parameter a list of words that represent text comments text and returns a word list representing the stems-root words- using PorterStemmer function deom the nltk.stem library.

[41] (Steven Bird, 2009)
[42] https://pythonprogramming.net/stemming-nltk-tutorial/

2.5 Machine learning techniques

Machine learning represents a subfield of computer science that offers computers the capacity to learn without being programmed explicitly. This field evolved from computational learning theory in artificial intelligence and the study of pattern recognition. The field is divided into three categories, depending on the type of feedback used in order to develop the learning algorithms.[43] These categories are:

- Supervised learning: the machine receives inputs as an example and their desired outputs and uses algorithms in order to learn a general rule that is able to map the given inputs to outputs.
- Unsupervised learning: the machine finds structure and patterns in the input by its own/
- Reinforcement learning: the machine relies on rewards and punishments as a feed-back from the environment in order to learn how to make decisions which bring a better outcome with each computation

The presented application used supervised learning as well as unsupervised learning algorithms in order to analyze and draw conclusions from the comments.

The development of machine learning algorithms is possible given the fact that almost all nonrandom data presents patterns, allowing the program to make generalizations and to determine the most important characteristics of data. Machine learning algorithms present a high dependence on statistics. [44]

One of the principal advantages of machine learning algorithms usage is the fact that machines can process data much faster and sometimes more accurately than human beings. Therefore, a large amount of data can be analyzed in a shorter amount of time.

However, according to Programming Collective Intelligence authors, machine learning presents some weaknesses. One of the possible drawbacks of machine learning is the fact that computer programs lack the cultural knowledge and subjective experience humans have, this is why these kind of programs can only make decisions based only on a limited amount of data.

In addition, these methods can fall into overgeneralizing trap, given the fact that strong generalizations can be based on a limited number of observations which may not be entirely accurate.[45]

[43] https://en.wikipedia.org/wiki/Machine_learning
[44] (Segaran, 2007)
[45] (Segaran, 2007)

Nonetheless, as long as new information enters the system the machine capacities improve, these limitations may be overcome over time.

Other real-life applications of machine learning algorithms, besides natural language processing, are: biotechnology, financial fraud detection, machine vision, product marketing, supply chain optimization, stock market analysis and national security.[46]

In the following pages, the machine learning algorithms used in order to process the comments will be detailed.

2.5.1 Unsupervised learning

As mentioned earlier, unsupervised learning is a machine learning approach where the algorithms don't rely on examples of correct answers in order to make further decisions.[47] Given the fact that unsupervised learning algorithms rely on activities like reconstruction, generation or density estimation, useful representation for a specific task may prove to be laborious. This is why much work has been put into designing architectures meant to enforce useful representation learning.[48]

An unsupervised learning approach is clustering. Clustering algorithms take the data and identify different groups that exist within it.[49]

Hierarchical clustering: traditional approach and sci-kit learn approach

One approach of clustering is represented by hierarchical clustering, which, according to the description made by Toby Segaran in his book, has the purpose of building a hierarchy of groups by merging in a continuous manner the two most similar groups. The flow of this kind of algorithm is represented in Figure 2.7.1:

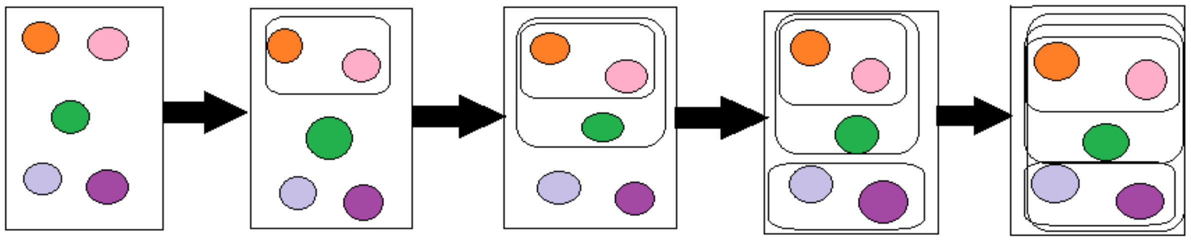


Figure 2.7.1: Hierarchical Clustering Process

[46] (Segaran, 2007)
[47] (Segaran, 2007)
[48] (Learning to Generate Reviews and Discovering Sentiment, 2017)
[49] (Segaran, 2007)

As it can be noticed in the above figure, in the first steps the groups are represented individually and then the closest groups are merged. The output of the hierarchical clustering is best represented using a dendrogram that draws the closest nodes in a tree-like manner, as the one presented in Figure 1.7.2:

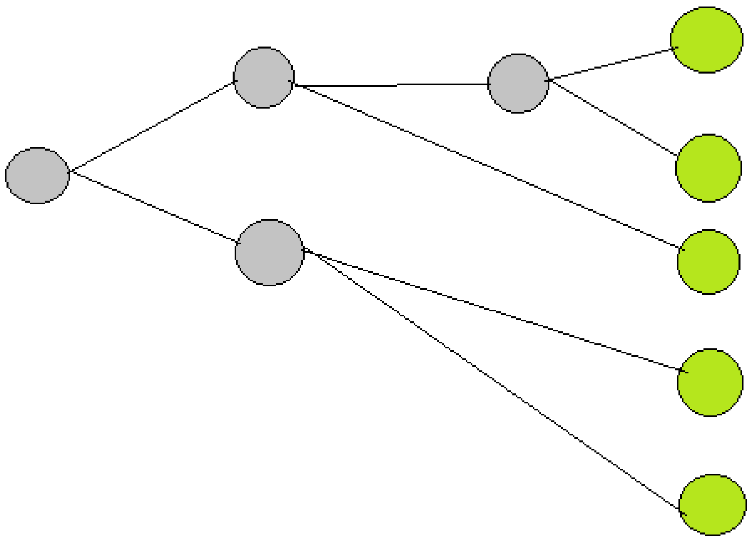
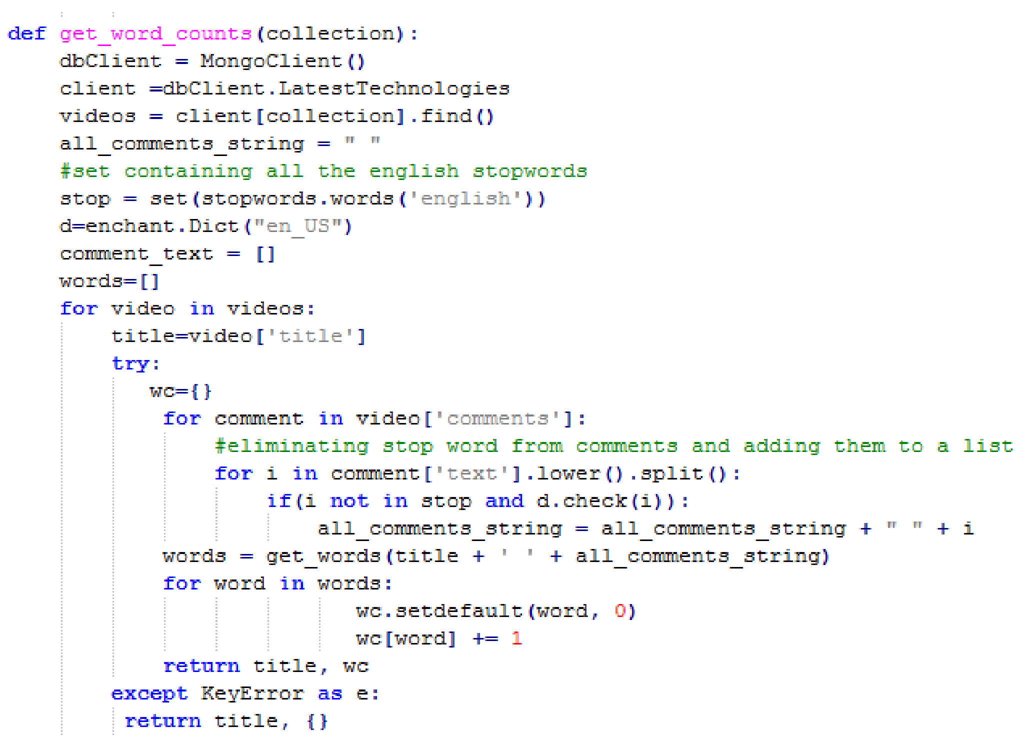


Figure 2.7.2: Dendrogram representation

This chapter presents the way in which videos belonging to a certain subject of interest group together in order to form clusters, based on the number of times the most frequent words typed by YouTube users are recorded in each video's comments. According to Toby Segaran, hierarchical clustering requires the steps described in the following paragraphs.

The first step is to read in a proper manner the data used for clustering, in our case, the comments and video titles. In order to do so, we compute **a frequency table of the words**, in which the column headers represent the words, the line headers represent the video title, and the intersection reflects the number of times a word appears in that video's comments.

By clustering YouTube videos based on word frequencies, we can determine the existence of video groups that provoke reactions manifested in similar ways.



```
def get_word_counts(collection):
    dbClient = MongoClient()
    client =dbClient.LatestTechnologies
    videos = client[collection].find()
    all_comments_string = " "
    #set containing all the english stopwords
    stop = set(stopwords.words('english'))
    d=enchant.Dict("en_US")
    comment_text = []
    words=[]
    for video in videos:
        title=video['title']
        try:
           wc={}
            for comment in video['comments']:
                #eliminating stop word from comments and adding them to a list
                for i in comment['text'].lower().split():
                    if(i not in stop and d.check(i)):
                        all_comments_string = all_comments_string + " " + i
            words = get_words(title + ' ' + all_comments_string)
            for word in words:
                           wc.setdefault(word, 0)
                           wc[word] += 1
            return title, wc
        except KeyError as e:
         return title, {}
```

Figure 2.7.3: Definition of get_word_counts function

The role of *get_word_counts* function is to get the title of every video and all the words used in that videos comments alongside the number of their occurrences. In order to do so, the function connects to the database and extracts data corresponding to the collection name passed as parameter. Then the function uses *get_words* as described above in order to create a list of words corresponding to each video title. After that, the function iterates through the word list and creates a dictionary containing the words as keys and the number of their appearances as values. Finally, the title of video and the corresponding dictionary are returned.

After creating the proper dataset, the **next step is to find out the closeness between videos**. The closeness between two videos is calculated using the Pearson correlation coefficient, which has the following formula:

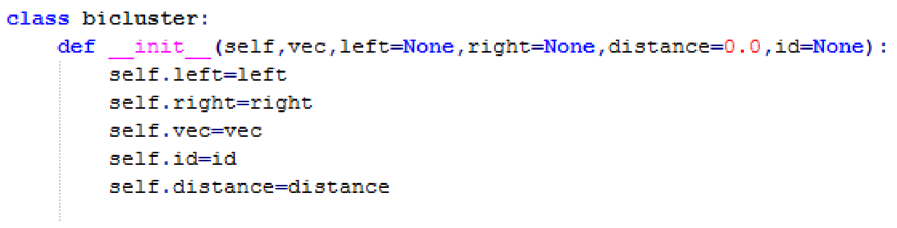
$$\rho_{xy} = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

Where:

- Cov is the covariance
- σ_x is the standard deviation of x
- σ_y is the standard deviation of y

The closer the Pearson correlation is to 1, the stronger the link between two videos is. If the Pearson correlation value is closer to 0, then there is no relationship between two videos.

In order to calculate the Pearson correlation coefficient, the function named *pearson_correlation* was defined. Given the fact that each cluster formed by this algorithm can be either a point in the tree or an endpoint associated to the video, the *bicluster* class, presented below, is created. Its purpose is to preserve the properties of each point.



```
class bicluster:
    def __init__(self,vec,left=None,right=None,distance=0.0,id=None):
        self.left=left
        self.right=right
        self.vec=vec
        self.id=id
        self.distance=distance
```

Figure 2.7.4: Definition of bicluster class

The third step is to write **the algorithm for hierarchical clustering**. The algorithm starts with the creation of a group of clusters that represent the videos taken individually. The main loop of the function looks for the two closest videos by taking every possible pair and computing the correlation coefficient. The best pair of videos is merged into a cluster. Then, the data characterizing the new cluster is computed as the average of the data that characterized the old clusters. The loop continues until there is one cluster left. The definition of the function *hierarchical_clustering*, which is applying this algorithm can be found in Appendix 1.[50]

Finally, in order to graphically represent the cluster by searching through it recursively, the *get_height*, *get_depth*,*draw_node* and *drawdendogram* functions have been created.

The *get_height* function checks if the current cluster is an end point. If so, the value 1 is returned. Otherwise, the height of that cluster is represented by the sum of the heights of its branches.

The *get_depth* function calculates the total error of the root node. Therefore a scaling factor is generated, taking into account how much error there is. The error depth of a node represents the maximum possible errors from each of its branches.

The *draw_node* function takes the heights of the child nodes belonging to the current cluster and calculates where they should be. Then, the function unites the nodes through lines. Longer lines reflect the fact that the two videos which form a certain cluster

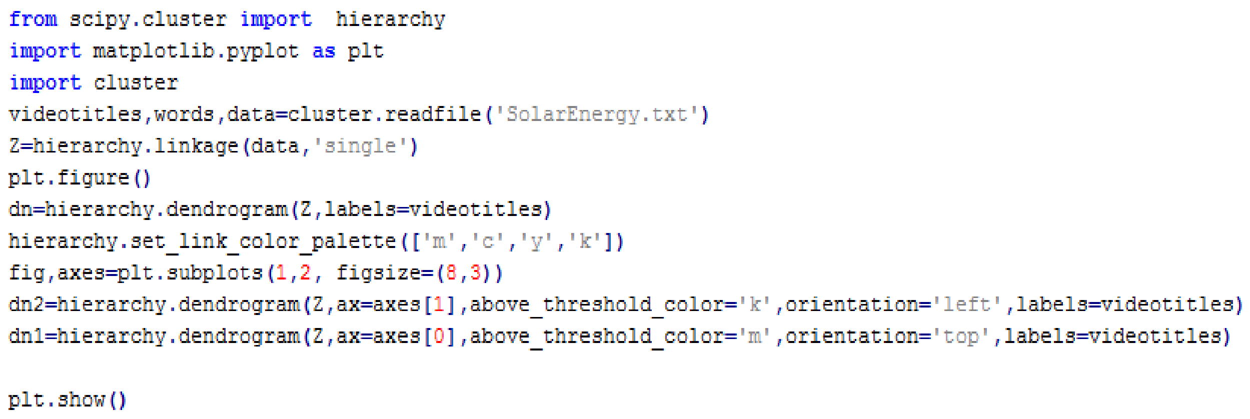
[50] Definition of hierarchical_clustering function, Appendix1

are not so similar taking into consideration the content of the comments, while the shorter lines notify the fact that they're almost identical.

Finally, the *draw_dendogram* function creates a new image representing the clusters. The scaling factor of the image is calculated by dividing the width by the total depth of the cluster.

An example of an output corresponding to the application of hierarchical clustering algorithm on videos falling into Solar Energy category can be found in Appendix 1.[51]

An alternative approach to hierarchical clustering classical algorithm is the usage of Sci-kit learn library, which hides the complexity of the algorithm and provides functions meant to create and draw the clusters in fewer lines of code, as presented in Figure 2.7.7.



```
from scipy.cluster import  hierarchy
import matplotlib.pyplot as plt
import cluster
videotitles,words,data=cluster.readfile('SolarEnergy.txt')
Z=hierarchy.linkage(data,'single')
plt.figure()
dn=hierarchy.dendrogram(Z,labels=videotitles)
hierarchy.set_link_color_palette(['m','c','y','k'])
fig,axes=plt.subplots(1,2, figsize=(8,3))
dn2=hierarchy.dendrogram(Z,ax=axes[1],above_threshold_color='k',orientation='left',labels=videotitles)
dn1=hierarchy.dendrogram(Z,ax=axes[0],above_threshold_color='m',orientation='top',labels=videotitles)

plt.show()
```

Figure 2.7.7: Sci-kit Learn hierarchical clustering

2.5.2 Supervised learning

As stated earlier, supervised learning represents a machine learning method in which the machine is given a part of the expected output to learn from, and learns progressively from that initial dataset.[52]

In YouTube E-motion application, a classifier is used in order to determine the polarity of comments, or, in other words, to perform sentiment analysis. Other purpose of this approach is to represent the evolution of polarity over time, and to draw a pie chart representing the proportion positive and negative words found through the comments of each

[51] Dendogram representing clusters from Solar energy category, Appendix1
[52] (Kilian Thiel, 2016)

video category. In order to do so, the steps described presented on pythonprogramming.net website were followed:

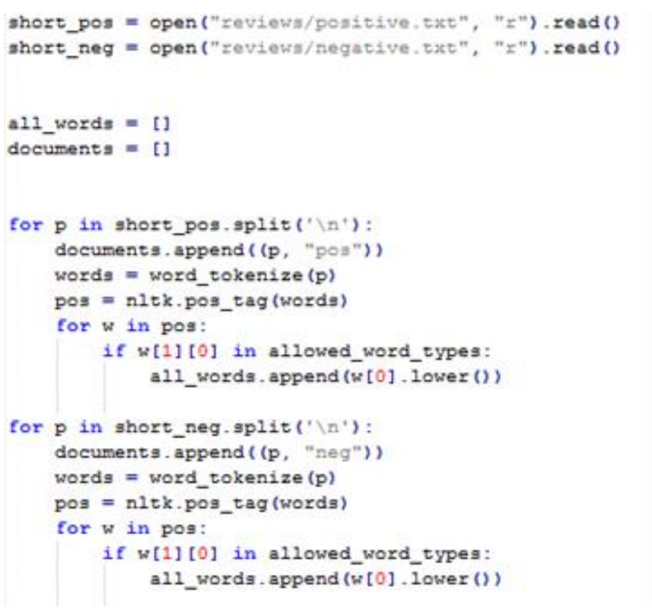
1. Preparation and loading of the dataset
2. Preprocessing and feature extraction
3. Sentiment classification
4. Classification evaluation
5. Visualization

Classification represents the task of giving the proper label to a given input, in our case to determine if the comments are negative or positive. Taking into account basic classification tasks, each input is considered to be separated from all the other inputs, the set of labels being defined in advance.[53]

In order to perform sentiment analysis, we will split the data in two: training set and testing set. The training set is represented by two files containing short movie reviews and manually selected comments related to each subject of interest. One file contains comments and reviews labeled as negative, while the other file contains comments and reviews labeled as positive.[54] These training sets were chosen given the fact that there is a high degree of similarity between the language used in YouTube comments and the language used in movie reviews.

In Figure 2.7.8 the way *documents* variable is created is presented. The variable is a list containing tuples corresponding to each review from the training set and its associated polarity.

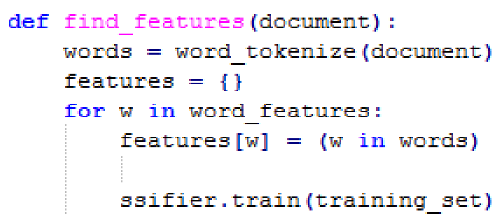
[53] (Steven Bird, 2009)
[54] https://pythonprogramming.net/new-data-set-training-nltk-tutorial/



```
short_pos = open("reviews/positive.txt", "r").read()
short_neg = open("reviews/negative.txt", "r").read()


all_words = []
documents = []


for p in short_pos.split('\n'):
    documents.append((p, "pos"))
    words = word_tokenize(p)
    pos = nltk.pos_tag(words)
    for w in pos:
        if w[1][0] in allowed_word_types:
            all_words.append(w[0].lower())

for p in short_neg.split('\n'):
    documents.append((p, "neg"))
    words = word_tokenize(p)
    pos = nltk.pos_tag(words)
    for w in pos:
        if w[1][0] in allowed_word_types:
            all_words.append(w[0].lower())
```

Figure 2.7.8: Data preparation script

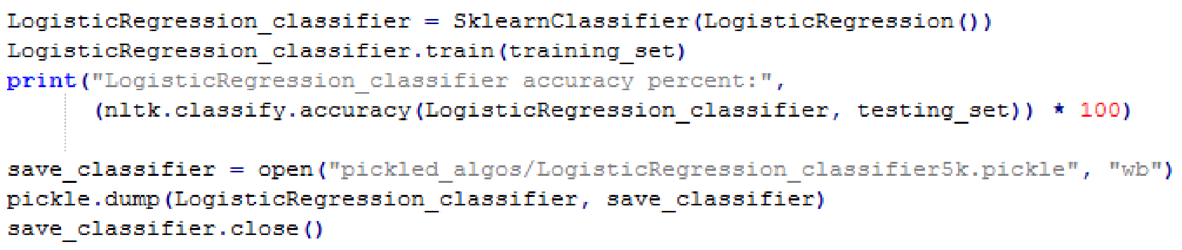
The next step is to adjust the feature finding function, tokenizing by word in the document, as presented in Figure 2.7.9



```
def find_features(document):
    words = word_tokenize(document)
    features = {}
    for w in word_features:
        features[w] = (w in words)

        ssifier.train(training_set)
```

Figure 2.7.9: Definition of find_features function

After passing the first two steps in the sentiment classification process, which were preparation of the dataset and feature extraction, we can go on and train a classifier. In order to do so, classifiers provided by Scikit-learn library which are incorporated by NLTK were used. This way Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Logistic Regression, SGD, SVC, LinearSVC and NuSVC classifiers were trained alongside their accuracy percent, in order to apply the most efficient classifier in the end. Figure 2.7.10 presents the initialization of Logistic Regression classifier. The other classifiers are initialized in similar ways.



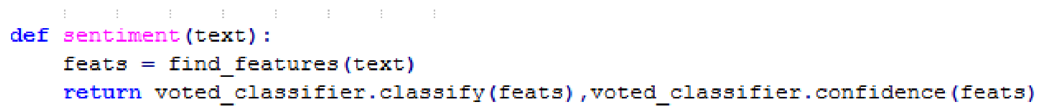
```
LogisticRegression_classifier = SklearnClassifier(LogisticRegression())
LogisticRegression_classifier.train(training_set)
print("LogisticRegression_classifier accuracy percent:",
      (nltk.classify.accuracy(LogisticRegression_classifier, testing_set)) * 100)

save_classifier = open("pickled_algos/LogisticRegression_classifier5k.pickle", "wb")
pickle.dump(LogisticRegression_classifier, save_classifier)
save_classifier.close()
```

Figure 2.7.10: Classifier training code

Given the fact that the training of the classifier is based on a large data set, a long period may be required in order for the algorithm to run, depending on computer performance. This problem is solved by serializing the classifiers, which are actually objects, in a static pickle file. The biggest advantage of this approach is that the classifier training becomes a one and done process. After pickling, data is read directly from a file, requiring substantially less time.[55]

After that, the feature sets extracted from the *featuresets* pickle are shuffled, and split in groups of 10000 features, one group representing the training set, while the other group represents the testing set. After all the saved classifiers are loaded from the pickle files, an object of type *VoteClassifier* is created. The object is constructed based on all the computed classifiers, and has two available methods: classify and confidence.

After the creation of the *VoteClassifier* object, the *sentiment* function is computed, as described in Figure 2.7.11



```
def sentiment(text):
    feats = find_features(text)
    return voted_classifier.classify(feats),voted_classifier.confidence(feats)
```

Figure 2.7.11: Definition of sentiment function

The function takes one parameter, represented by a text and breaks down the features with *the find_features* function described above. Then, the *VoteClassifier* object is used in order to not only return the classification of each comment, but also to return the confidence in that particular classification. In order to ensure the accuracy of the classification, only comments whose classification exceeds a 80% confidence degree are kept.

[55] https://pythonprogramming.net/sentiment-analysis-module-nltk-tutorial/

3. WEB PAGE DESCRIPTION

As stated earlier, YouTube E-Motion is a single page application which aims to present the chosen categories and to display the analysis results in an interactive manner. In order to do so, the application uses HTML, CSS and Bootstrap for the structure and design of the page, Flask for micro-server connection and access of the resources corresponding to each category and finally AngularJS for routing and templating.

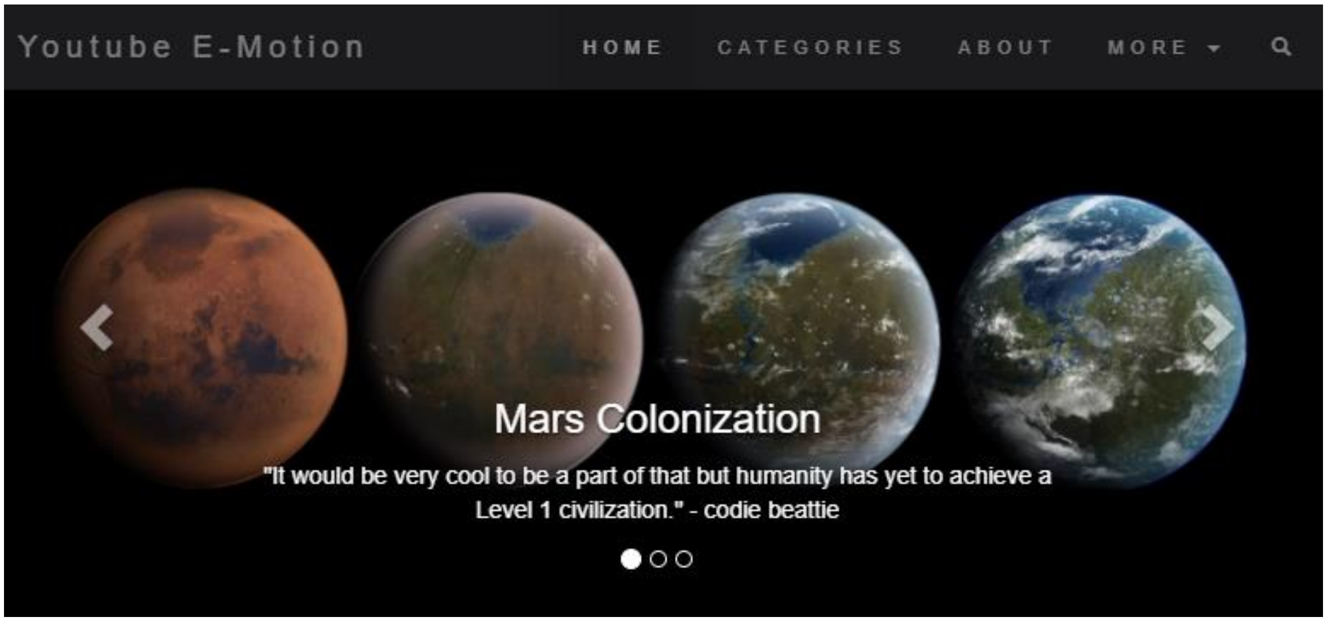

Youtube E-Motion
HOME
CATEGORIES
ABOUT
MORE
Mars Colonization
"It would be very cool to be a part of that but humanity has yet to achieve a Level 1 civilization." - codie beattie

Figure 3.1: YouTube E-motion web page

The page has three main sections, as presented in the navigation bar situated on the top of the page: “Home”, “Categories” and “About”, which correspond to the main divisions of the starting page. The link named “More” directs to more detailed information regarding each of the eight categories.

The “Home” link directs to the first division of the page, which is a carrousel containing representative photos for the chosen subjects, as well as a comments posted on YouTube by random users.

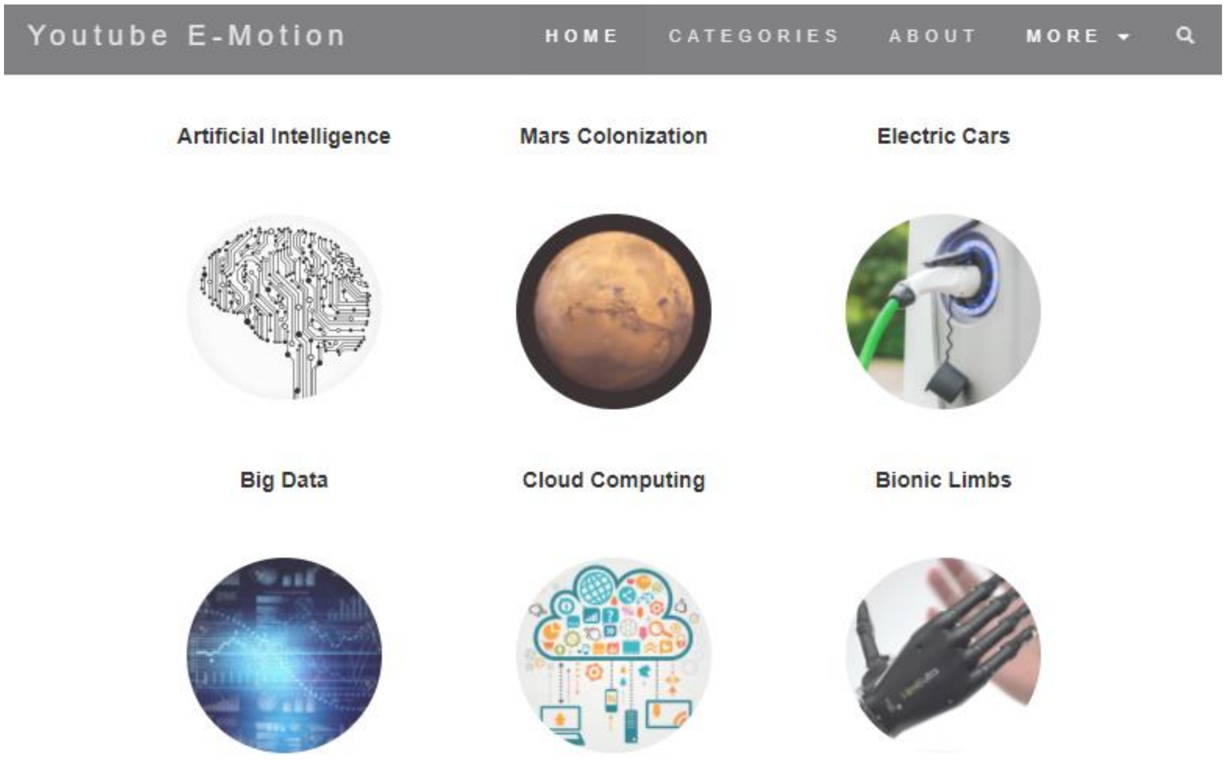


Figure 3.2: YouTube E-motion web page categories

The next division of the page contains a menu that allows the user to pick a subject he or she is interested in. (Figure 3.2) .When an item is selected, a report about the videos and video comments is generated, including graphs and images and descriptions for each analysis approach.

Finally, the “About” section describes briefly the application’s purpose.

CONCLUSIONS

As stated earlier, sentiment analysis performed with the help of machine learning techniques can be very useful in a society dominated by information flow and freedom of expression. Our nature invites us to express our opinions regarding various interests of subjects, products, events, or anything else part of our daily lives. The main purpose of opinion mining is to detect the polarity of such manifestations, which may contain positive or negative statements. The rising of the social media over the past ten years has also offered sentiment analysis an increasing importance and opinion mining has rather become a rule than an exception. As a matter of fact, this is a reason why a lot of advances in the natural language processing field are made.

The content generated by every user day-by-day represents a treasure for many companies because social media is the easiest way to gain access to the customer opinions and receive real-time feedback.

However, sentiment analysis has its challenges. First of all, the volume of data is increasing fast and because it is mostly formed of text, or unstructured data, it becomes more and more difficult to process it. Secondly, a machine can get easily confused because people express themselves by using abbreviations, emoticons, or colloquial language. The larger the amount of data introduced in the system, more information the program gets, the more accurate are its conclusions.

Even so, the machine is sometimes unable to detect the border between literal meaning and sarcasm.

Further implementations

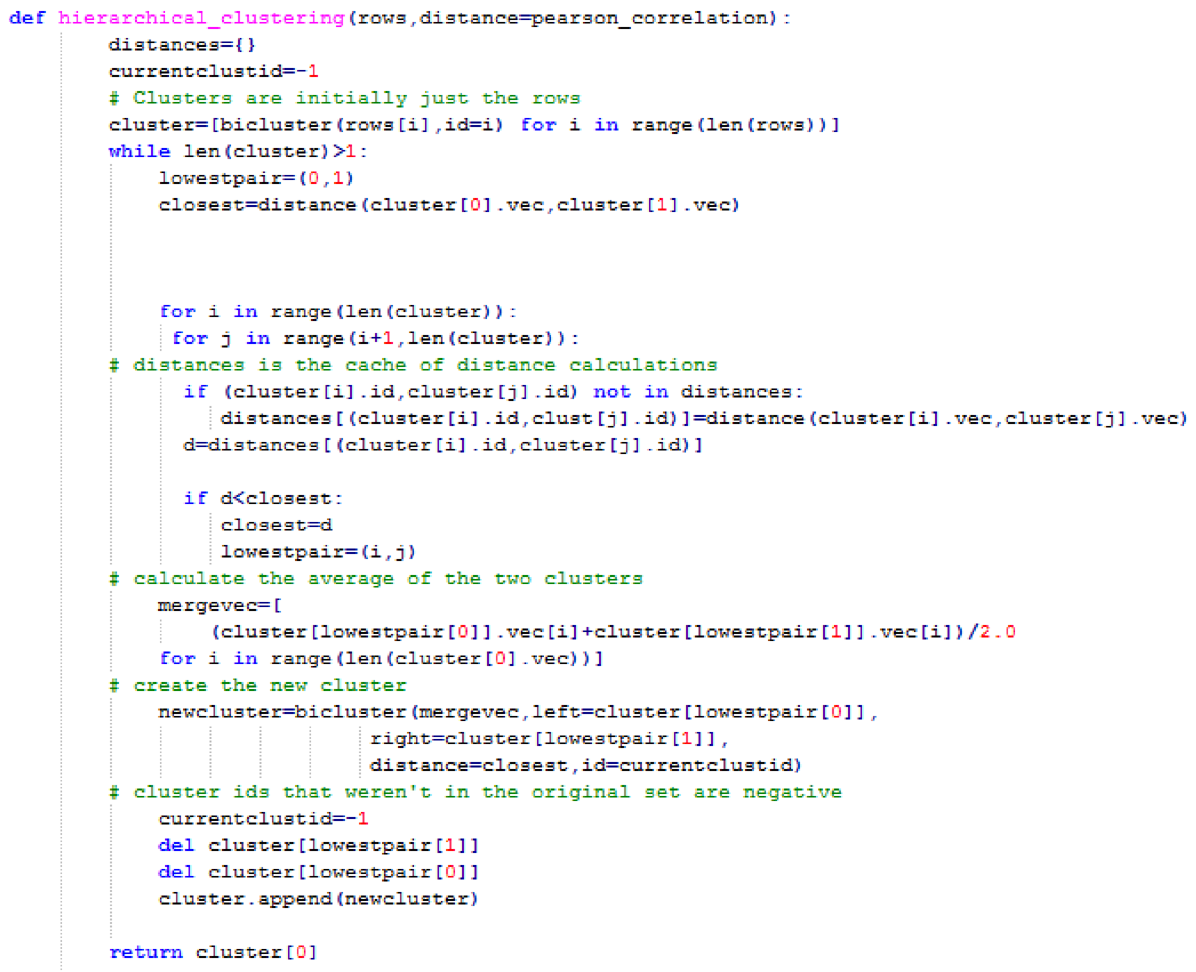
YouTube E-motion application opens the door to a large variety of further implementations.

First of all, the methodology used in order to perform sentiment analysis can be applied as well to other languages. Even though we live in a globalized world, in which a large proportion of internet users speak English, there are also entities who may benefit from sentiment analysis in languages like Romanian.

Secondly, based on the proportion of comment polarity over time, a prediction model can be found and applied in order to determine the tendencies and the future preferences of uses, such that companies can adapt to previsioned behaviors of clients.

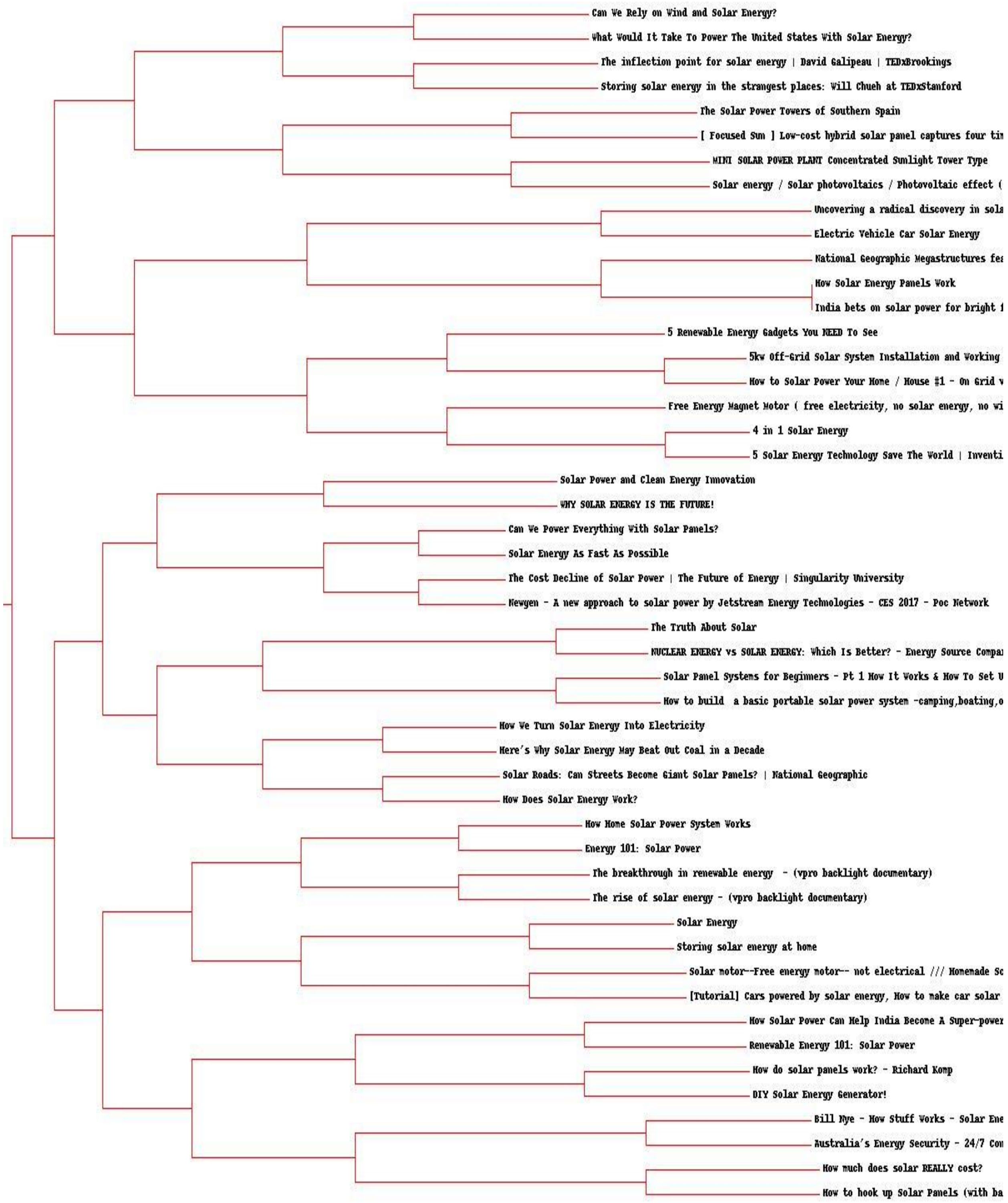
Finally, the application can work on a much larger scale than eight subjects of interest, performing a real-time analysis on a subject typed in by the user. This way, any existent subject on YouTube could be approached. However, the scale of such a project would increase a lot, requiring also a proportional increase in hardware performance, namely processing power.

APPENDIX1



```
def hierarchical_clustering(rows,distance=pearson_correlation):
        distances={}
        currentclustid=-1
        # Clusters are initially just the rows
        cluster=[bicluster(rows[i],id=i) for i in range(len(rows))]
        while len(cluster)>1:
            lowestpair=(0,1)
            closest=distance(cluster[0].vec,cluster[1].vec)



            for i in range(len(cluster)):
             for j in range(i+1,len(cluster)):
        # distances is the cache of distance calculations
              if (cluster[i].id,cluster[j].id) not in distances:
                 distances[(cluster[i].id,clust[j].id)]=distance(cluster[i].vec,cluster[j].vec)
              d=distances[(cluster[i].id,cluster[j].id)]

              if d<closest:
                 closest=d
                 lowestpair=(i,j)
        # calculate the average of the two clusters
            mergevec=[
                (cluster[lowestpair[0]].vec[i]+cluster[lowestpair[1]].vec[i])/2.0
            for i in range(len(cluster[0].vec))]
        # create the new cluster
            newcluster=bicluster(mergevec,left=cluster[lowestpair[0]],
                             right=cluster[lowestpair[1]],
                             distance=closest,id=currentclustid)
        # cluster ids that weren't in the original set are negative
            currentclustid-=1
            del cluster[lowestpair[1]]
            del cluster[lowestpair[0]]
            cluster.append(newcluster)

        return cluster[0]
```

Definition of *hierarchical_clustering* function


Can We Rely on Wind and Solar Energy?
What Would It Take To Power The United States With Solar Energy?
The inflection point for solar energy | David Galipeau | TEDxBrookings
Storing solar energy in the strangest places: Will Chueh at TEDxStanford
The Solar Power Towers of Southern Spain
[Focused Sun] Low-cost hybrid solar panel captures four tim
MINI SOLAR POWER PLANT Concentrated Sunlight Tower Type
Solar energy / Solar photovoltaics / Photovoltaic effect (
Uncovering a radical discovery in sola
Electric Vehicle Car Solar Energy
National Geographic Megastructures fea
How Solar Energy Panels Work
India bets on solar power for bright f
5 Renewable Energy Gadgets You NEED To See
5kw Off-Grid Solar System Installation and Working
How to Solar Power Your Home / House #1 - On Grid v
Free Energy Magnet Motor (free electricity, no solar energy, no wi
4 in 1 Solar Energy
5 Solar Energy Technology Save The World | Inventi
Solar Power and Clean Energy Innovation
WHY SOLAR ENERGY IS THE FUTURE!
Can We Power Everything With Solar Panels?
Solar Energy As Fast As Possible
The Cost Decline of Solar Power | The Future of Energy | Singularity University
Newgen - A new approach to solar power by Jetstream Energy Technologies - CES 2017 - Poc Network
The Truth About Solar
NUCLEAR ENERGY vs SOLAR ENERGY: Which Is Better? - Energy Source Compa
Solar Panel Systems for Beginners - Pt 1 How It Works & How To Set U
How to build a basic portable solar power system -camping,boating,o
How We Turn Solar Energy Into Electricity
Here's Why Solar Energy May Beat Out Coal in a Decade
Solar Roads: Can Streets Become Giant Solar Panels? | National Geographic
How Does Solar Energy Work?
How Home Solar Power System Works
Energy 101: Solar Power
The breakthrough in renewable energy - (vpro backlight documentary)
The rise of solar energy - (vpro backlight documentary)
Solar Energy
Storing solar energy at home
Solar motor--Free energy motor-- not electrical /// Homemade Sc
[Tutorial] Cars powered by solar energy, How to make car solar
How Solar Power Can Help India Become A Super-power
Renewable Energy 101: Solar Power
How do solar panels work? - Richard Komp
DIY Solar Energy Generator!
Bill Nye - How Stuff Works - Solar Ene
Australia's Energy Security - 24/7 Con
How much does solar REALLY cost?
How to hook up Solar Panels (with ba

Dendogram representing clusters formed in Solar Energy category

REFERENCES

1. **Alex Smola, S.V.N. Vishwanathan. 2008.** *Introduction to Machine Learning.* s.l. : Cambridge University press, 2008.
2. *Analyzing and Mining Comments and Comment Ratings on the Social Web.* **Stefan Siersdorfer, Sergiu Cehlaru, Jose San Pedro, Ismail Sengor Altingovde, Wolfgang Nejdl. 2014.** 2014.
3. *Introduction to Machine Learning The Wikipedia Guide.*

   Kilian Thiel, Tobias K¨otter,Rosaria Silip,Phil Winters. 2016. *Text Mining and Visualization: Case Studies Using Open-Source Tools.* s.l. : Chapman & Hall/CRC, 2016.
4. *Learning to Generate Reviews and Discovering Sentiment.* **Sutskever, Alec Radford Rafal Jozefowicz Ilya. 2017.** 2017.
5. *Polarity Trend Analysis of Public Sentiment on YouTube.* **Amar Krishnay, Joseph Zambreno and Sandeep Krishnan.**
6. *Scikit-learn: Machine Learning in Python.* **Fabian Pedregosa, Gael Varoquaux ,Alexandre Gramfort ,Vincent Michel. 2011.** 2011, Vol. Journal of Machine Learning Research.
7. **Segaran, Toby. 2007.** *Programming Collective Intelligence.* 2007.
8. *Sentiment Analysis and Opinion Mining.* **Liu, Bing. 2012.** 2012.
9. **Steven Bird, Ewan Klein, Edward Loper. 2009.** *Natural LanguageProcessing with Python.* s.l. : O'Reilly Media, 2009.
10. **Willi Richert, Luis Pedro Coelho. 2013.** *Building Machine Learning Systems with Python.* s.l. : 3 Packt Publishing, 2013.
11. (https://en.wikipedia.org/wiki/Single-page_application) -last accessed: June 2017
12. (https://en.wikipedia.org/wiki/Python_(programming_language)) last accessed: June 2017
13. (http://www.nltk.org/) last accessed: June 2017
14. (https://en.wikipedia.org/wiki/Scikit-learn)- last accessed: June 2017
15. (https://en.wikipedia.org/wiki/BSD_licenses)- last accessed: June 2017
16. (https://en.wikipedia.org/wiki/Application_programming_interface)- last accessed: June 2017
17. (https://developers.google.com/youtube/v3/docs/) last accessed: June 2017

18. (https://en.wikipedia.org/wiki/Flask_(web_framework)) last accessed: June 2017
19. (http://flask.pocoo.org/docs/0.12/) last accessed: June 2017
20. (https://en.wikipedia.org/wiki/HTML)- last accessed: June 2017
21. (https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) last accessed: June 2017
22. (https://www.w3schools.com/ht0)ml/html5_new_elements.asp) last accessed: June 2017
23. (https://www.w3schools.com/css/css_intro.asp) last accessed: June 2017
24. (https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)) last accessed: June 2017
25. (https://en.wikipedia.org/wiki/AngularJS) last accessed: June 2017
26. (https://en.wikipedia.org/wiki/NoSQL) last accessed: June 2017
27. (https://docs.mongodb.com/manual/introduction/) last accessed: June 2017
28. (https://docs.mongodb.com/manual/core/databases-and-collections/) last accessed: June 2017
29. (https://en.wikipedia.org/wiki/PyCharm) last accessed: June 2017
30. (https://www.jetbrains.com/pycharm/features/) last accessed: June 2017
31. (https://git-scm.com/about) last accessed: June 2017
32. (https://www.analyticsvidhya.com/blog/2014/09/mining-youtube-pythonsocial-media-analysis/) last accessed: June 2017
33. (https://docs.python.org/2/tutorial/datastructures.html#dictionaries) last accessed: June 2017
34. (https://chrisalbon.com/python/pandas_dataframe_descriptive_stats.html)last accessed: June 2017
35. (http://www.nltk.org/_modules/nltk/probability.html) last accessed: June 2017
36. (https://pythonprogramming.net/stemming-nltk-tutorial/) last accessed: June 2017
37. (https://en.wikipedia.org/wiki/Machine_learning) last accessed: June 2017
38. (https://pythonprogramming.net/new-data-set-training-nltk-tutorial/) last accessed: June 2017
39. https://pythonprogramming.net/sentiment-analysis-module-nltk-tutorial/)last accessed: June 2017