Faculty of Computer Science and Information Technology

DATA ANALYSIS IN TRENDING YOUTUBE VIDEOS FOR CATEGORY PEOPLE AND BLOG

TUNKU KHAIRI BIN TUNKU HANIZD

67962

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2021

DATA ANALYSIS IN TRENDING YOUTUBE VIDEOS FOR CATEGORY PEOPLE AND BLOG

TUNKU KHAIRI BIN TUNKU HANIZD

This project is submitted in partial fulfilment of the requirements for the Degree of Bachelor of Computer Science and Information Technology with Honors.

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2021

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE DATA ANALYSIS IN TRENDING YOUTUBE VIDEOS FOR CATEGORY PEOPLE AND BLOG

ACADEMIC SESSION: 2018/2019

____

(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (√)

☑ CONFIDENTIAL (Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)

□ RESTRICTED (Contains restricted information as dictated by the body or organization where the research was conducted)

□ UNRESTRICTED

Validated by

Khairi
(AUTHOR'S SIGNATURE)

Permanent Address:

209, JALAN ANGGERIK 3/1, SUNGAI BULOH COUNTRY RESORT
47000 SUNGAI BULOH, SELANGOR DARUL EHSAN

Date: 28/6/2022

(SUPERVISOR'S SIGNATURE)

Date: ______________________

Note * Thesis refers to PhD, Master, and Bachelor Degree
** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

DECLARATION

I hereby declare that this project is my original work. I have not copied any other student's work or from any other sources except where due reference or acknowledgment is not made explicitly in the text, nor has any part been written for me by another person.

Khairi

…………………………………….

TUNKU KHAIRI BIN TUNKU HANIZD

Information System
Faculty Computer Science and Information Technology
Universiti Malaysia Sarawak 28 June 2022

Acknowledgment

First and foremost, I would like to praise Allah S.W.T, for providing good health to complete the final year project. I pray that every day I will be blessed by Allah S.W.T the ease of my tasks and always in good health.

Secondly, I want to express my gratitude to my parents, Tunku Hanizd bin Tunku Daud and Norazah bt. Mohammad Ali @ Nordin for their never-ending support, guidance, and prayers throughout my journey of completing my degree studies.

Not to forget, I would like to express my gratitude to Dr. Mohammad bin Hossin for his guidance and encouragement on the purpose of assisting me to complete Final Year Project 1. I greatly appreciate the time commitment you have given me.

In addition, I would like to thank my friends for providing me with motivation and entertainment on my sad days.

Finally, I want to express my deepest gratitude to myself for never giving up despite all challenges encountered and still staying positive even in unfortunate events.

Contents

List of Figure

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Background

YouTube was founded by Chad Hurley, Steve Chen, and Jawed Karim in 2005. YouTube originally was created as a platform to post videos of one's desire so people can publicly watch from any part of the world. Over time, YouTube become dominant as one of the largest free video-sharing websites that hosts billions of views per day. Because of huge traffic ongoing every day, YouTube enables to invent their new special program which is called "YouTube 's Partner Program". Alongside Google's AdSense from its parent company, which is Google, YouTube has provided a career opportunity for content creators called "YouTube content creators". The content creators earn their living from ads revenue, sponsors, or paid reviews included in their uploaded videos.

YouTube as a medium of expression provides public statistics for each uploaded video which are upload date, number of views, number of likes, and dislikes. YouTube also provides a comment section for sentimental engagement of the videos. YouTube determines the high engagement videos called "*trending videos"* by indicating the level of popularity of the videos based on the high number of views, likes, and positive comments for each video.

Although YouTube is a platform where content creators could freely upload their videos based on YouTube's guidelines, it is difficult to have high interactions in a single video.

YouTube categorizes the videos by which the topic is covered in the video. One of the categories is *“People and Blogs”*. The category covers videos that are related to people’s lifestyles, news about people, promotions, reviews, blogs, and topics that correlate with people. This project aims to study the factors that can give positive impacts on videos in this category to make attract the viewers' interest to interact with the videos using data analysis.

1.2 Problem Statement

According to an Alexa report, YouTube has become one of the most preferred digital video platforms that intercept more than 30 million visitors in a day. The largest portion of viewers comes from the USA with 16.4% of traffic followed by India (9.2%) and Japan (4.8%) respectively. Because of high traffic and more videos being added every day, unveiling the viewership pattern is considered a complex process for a normal content creator with no Data Science background.

YouTube has massive datasets that are categorized as Big Data. If the attributes of the trending videos are thoroughly studied and analyzed using data analytics, content creators could have the opportunity to greatly increase the engagement quality of their videos thus improving the marketing strategies of their videos.

To understand the causal of high interaction videos, in-depth analysis is required.

1.3 Scope of The Project

The data analysis covers trending videos from the YouTube platform. Due to the massive datasets that YouTube possesses, it is time-consuming and potentially can be misleading when interpreting the data. Therefore, this project will focus on the trending videos from the US region. This is because the US contributes the largest percentage of YouTube traffic with 16.4% visitors per day. This project also focuses on trending videos in the category "People and Blog".

1.4 Aim and Objectives

The purpose of this project is to discover how videos in the category of "People and Blogs" reached a high engagement. This project intends to uncover the hidden pattern by answering various questions about the trending YouTube videos using co-related analysis of Exploratory Data Analysis and Sentiment Analysis.

In particular, the objectives are:

- To analyze the patterns of high-interaction videos on YouTube using Exploratory Data Analysis
- To determine the topic consisted in the videos by using the Unsupervised Learning method.
- To discover optimal settings needed for the videos to achieve high interactions.

1.5 Brief Methodology

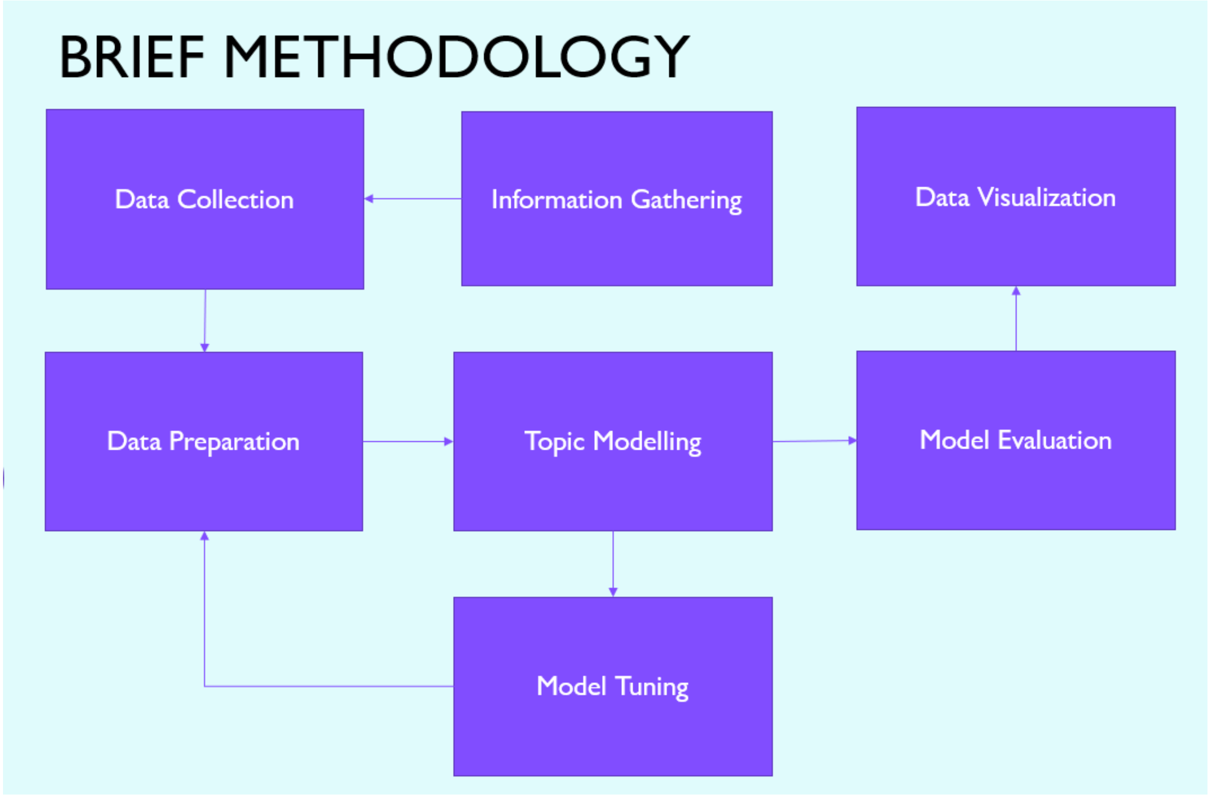


Figure 1.1:Brief Methodology of Project

Figure 1.1 shows the overview of the project workflow using a flowchart. In chapter 3, each process in the project workflow is discussed thoroughly.

1.6 Significance of Project

The project is conducted to utilize YouTube as a platform to further improvise digital marketing and help content creators storyboard their future videos. The project is also beneficial for content creators especially smaller channels with fewer viewers to grow their audience. It is done by discovering optimal settings necessary to make their videos trending and always relevant on YouTube. The project also is conducted to reveal hidden viewership patterns.

1.7 Project Schedule

Please refer to Appendix A and Appendix B for the Project schedule.

1.8 Expected Outcome

At the end of this project, the analysis is expected to find the correlation and impactful attributes of trending videos. The analysis is also expected to find viewership patterns of trending videos for the category “People and Blog”.

1.9 Project Outline

Chapter 1: Introduction uses to describe the background of the project by defining the elements of the research which are YouTube and its impact, the problem statements, scope, aim, and objectives, brief methodology, the significance of the project, project schedule, expected outcome and outline of the project.

Chapter 2: Literature Review describes the related topics discussed in previous papers on YouTube research, methodologies involved, validations, evaluations as well as visualization. This chapter also discusses the direction of the proposed project based on previous findings.

Chapter 3: Methodology describes how the project is conducted in detail. This chapter discusses information gathering, data collection, data pre-processing, feature extraction, data division, modeling process, evaluation, and visualization of the findings.

Chapter 4: Implementation discusses the detailed methodology used in the project according to the phases mentioned in the project overview.

Chapter 5: Analysis and Results Discussion consists of discussion findings based on analysis and model performance discussion.

Chapter 6: Conclusions and Future Works addresses the project achievement, contributions, limitations, and project future works.

CHAPTER 2: LITERATURE REVIEW

The study of users' behaviors on YouTube has been an interesting topic in the research world since YouTube started to rapidly grow into one of the largest video-sharing platforms on the internet. This literature review discusses the concept of relevant topics with the proposed project. Then the literature review covers the comparison of how related previous research is conducted. Finally, this chapter discusses the direction of the proposed project and the overview of the proposed project.

2.1 YouTube: A Miracle of One Video

According to Gohar Feroz Khan and Sokha Vong, YouTube is one of the most successful video-based communication mediums to express feelings, communicating with friends, and advertise business messages. YouTube was founded by Chad Hurley, Steve Chen, and Jawed Karim in 2005. Snickars, P., & Vonderau, P. (2009) stated that the 2 co-founders, Chad Hurley and Steven Chen successfully persuaded Google to invest $1.65 billion in stocks for the most-talked-about Web acquisition in October 2006. YouTube met its turning point to become one of the most preferred digital video platforms with a video featuring its third co-founder, Jawed Karim titled "Me at The Zoo". Little to his know, this spontaneous video has indicated inconspicuously that YouTube has the potential to become a proprietary platform with a new built-up online community (Snickars, P., & Vonderau, P., 2009).

2.3 The Phenomenon of Content Creation & Its Online Community

The ability to distribute content has attracted its community to upload various kinds of videos on YouTube. In the summer of 2006, YouTube already has 13 million visitors in its traffic with hundreds of million videos uploaded on the platform. Mattias Holmbom (2015) said that the videos uploaded on YouTube have endless variations of cultures and interests. He added the localization of YouTube in 75 countries acquired has sparked the ability of monetization. YouTube created a partnership program with media companies to run ads alongside videos, splitting revenues with its partner. Today as a user-driven platform, content creators have become the face of YouTube. The era of online entrepreneurship has continued to grow up until today and the main contributor of this era is called "*content creators*".

To content creators, YouTube is more than a personal hub to upload videos (Holmbom, M., 2015). Alongside the partnership program by YouTube, these content creators devoted their multiple hours of life every single day to creating content and uploading it on YouTube. Although developing a YouTube channel has a potential reward for its creator, it is difficult to have an audience big enough to provide feasible income through a content creation career path. According to Mattias Holmbom, this is due to the high saturation of YouTube channels, let alone YouTube videos. The rivalry for attention on this platform is so high that it may look impossible to some minds.

According to Rowley's findings (2004) in his writing, he stated that developing a YouTube channel has a similar concept to online branding. He added that most, if not all, practical steps for developing a YouTube channel are connected to online branding. By utilizing the analytic tools provided by YouTube via its partnership program, content creators can analyze the level of attractiveness of their channel and the videos they uploaded using metadata inserted in each video. Content creation also holds the value that their videos should be plausible content for advertisers.

2.4 Social Network and Video Blogger (Vlogger) Community

YouTube as part of social media, is used to convey the creators' personality and behaviors they intend to display. In November 2010, Biel, J. I., and Gatica-Perez, D. stated that these videos in the most basic format, are usually defined as conversational videos that serve as a tool for communication and interaction alongside serving as a living documentary of the creators. Serves as a unique medium for self-presentation and interpersonal perception that surpasses the use of text and still photos, conversational video blogs, or it is shorter-term, vlogs have become a unique category of video on YouTube (Biel, J. I., Aran, O., and Gatica-Perez, D., July 2011). Luers (2007) stated that vlog has several types of genres which are diary, experimental, documentary, and mash-up. According to Wauster (2010), YouTube hosts the largest number of video blogs with approximately 35% followed by Blip. tv (14%) and Vimeo (9%) 2010.

Mitchell (1969) describes a social network as "a specific set of linkages among a defined set of persons, with the additional property that the characteristics of these linkages as a whole may be used to interpret the social behavior of the persons involved".

2.5 YouTube Advertising

With the rise of smartphones, social media has become universally accessible, making it a significant platform. As a social site, YouTube is no exception. It lets users find new songs, artists, and funny videos, for example. As a result of the increased use of YouTube, it has become an important platform for businesses to reach their target audiences. Unlike traditional commercials, YouTube endorsement marketing, also known as native advertising, is a type of marketing in which adverts are effortlessly integrated into the video content (Katrina Wu, 2016).

There is a study that stated the effectiveness of YouTube advertising. Advertisers who run video ads on YouTube have increased their spending by more than 40% every year, while the top 100 advertisers on YouTube have increased their spending by more than 60% per year (Biographon, 2019).

According to Katrina Wu (2016), YouTube endorsement marketing is divided into 3 forms. The first form is when a content creator collaborates with a sponsor to generate videos, this is known as direct sponsorship. The second form called affiliated links is those in which the content creator receives a commission from sales that are attributed to them. The third form is free product sampling which companies send things to content creators for free in exchange for them to use in a video whether to review or simply advertise explicitly or even implicitly according to the consent of both parties; content creators and companies.

Dugyu Firait (2019) has conducted a study about the value that YouTube advertising and its effect on purchase intention for consumers. In his study, he finds that participants aged 40 and over believe YouTube advertisements to be more informative than those aged 18 to 29. Participants aged 30-39 believe YouTube commercials should be more interesting and trendier than those of other ages. In his findings, he concluded that the value of YouTube adverts has a beneficial impact on consumers’ purchasing intention.

2.6 YouTube Algorithm

YouTube uses the retrieval method as a feature for its video recommendation system. Essentially, YouTube delivers automatic suggestions to influence or assist users’ decision-making processes. According to Bendersky, M., Garcia-Pueyo, L., Harmsen, J., Josifovski, V.,

& Lepikhin, D. (2014), the algorithm used by the YouTube system generates a sorted list of relevant videos for the viewer to watch in response to the video users is now watching. The collaborative filtering analysis is used on YouTube as a foundation to tailor the recommendation system by adding information about related previously watched by the user or can be based just on collective patterns of users watching videos. Most users are likely presented by the co-view videos suggested by the algorithm with the videos that previously has been watched by the user with the same watching behaviors as them.

2.7 Latent Dirichlet Allocation (LDA) in Topic Modeling

Topic modeling is an unsupervised machine learning method of extracting themes as mathematical objects from a corpus of documents. According to Carina Jacobi et al. (2015), topic models are computer algorithms that work by using the distribution of words in a collection of documents to find latent patterns of word occurrence. Equal to other topic modeling algorithms, the Latent Dirichlet Allocation (LDA) algorithm is an unsupervised learning technique that creates topics based on patterns of words that co-occur when analyzing words in documents.

2.8 Review of Related Works

In this section, related work of YouTube research is reviewed which is divided into data collection, data preparation and analysis findings and evaluation, and lastly, visualization.

2.8.1 Data Collection

Several articles that discussed digital research stated that they scraped code-based data by querying the platforms' API for the data collection process referring to standards stated by Rogers (2013) discussed in his article about Digital Method.

In Choudhury, S., and Breslin, J. G. (2010) research on "*User sentiment detection: a YouTube use case"* they extracted data corpus on the most popular and relevant videos from five main categories and ten subcategories, including politics and news, science and technology, travel, music, movies, sports, gaming, people, and blogs. They gathered the 2,000 most popular videos in each category.

In the research of understanding the characteristics of internet short video sharing, Cheng, X., Dale, C., & Liu, J. (2007) scrapped data based on habits and social networks as they are of particular interest to that topic. They used a mix of the YouTube API and scrapes of YouTube video web pages to crawl the YouTube site for three months and collect information on its videos.

2.8.2 Data Preparation and Analysis

2.8.2.1 Data Preparation

In Severyn, A., Uryupina, O., Plank, B., Moschitti, A., & Filippova, K.'s research for opinion mining on YouTube (2014), they improvised the data preparation process by instead of using the traditional bag-of-word processing, they added new characteristic with features from a sentiment lexicon and features that quantify the negation in the comment.

For the research “User sentiment detection: a YouTube use case” by Choudhury, S. and Breslin, J. G. (2010), They did some simple pre-processing of the text material after gathering the video data and the related comments. Stop-word removal and term stemming were applied to the comments. They stemmed the terms using Porter stemming and then used SentiWordnet to detect sentiment polarity.

In Rinaldi, E., and Musdholifah‘s article about opinion mining on Indonesian comments on youtube videos using FVEC-SVM (2017), they executed the preparation step in 8 phases. Emoji Removal, Slash Removal, Punctuation Removal, Slang Word Fixing, POS-tagging, Tokenizing, Tupling, and Class Selection are the 8 phases in the pre-processing stage. Frequently, video comments have included emojis that are unrelated to the message. As a result, in this study, the Unicode number was used to delete the emoji from the comment. They used the FVEC approach and TF-IDF approach for the feature extraction process. The FVEC method entails creating unigram and bigram words, counting negations, and calculating the cosine similarity between the comments and the title. All generated unigram and bigram words are converted into TF-IDF vectors. To extract the number of negation terms in the comment, negative counting is used. This characteristic specifies whether the document is positive or negative in polarity. The number of negation words, TF-IDF of unigram and bigram words, and cosine similarity are the next features extracted from the comment.

For the research written about an exploratory investigation of music on YouTube by Airoldi, M., Beraldo, D., and Gandini, A. (2016), In the data preparation process, the data was acquired using the YouTube Data API v. 2.0. The data gathering procedure was divided into two parts. In the first step, we were able to acquire a generic list of videos associated with music content by querying the API for the keyword 'music' and setting the language option to English. The broad reach of the keyword allows for a wide range of musical genres to be covered, but because the search query is written in English, the results should be limited to that language. This initial batch has 500 videos since YouTube API v. 2.0 only permitted a maximum of 500 items to be extracted from one keyword. They crawled the algorithm of the YouTube-related video in the second phase to create a more consistent sample and a relational dataset for network research. They performed API requests to collect 25 related videos for each video obtained in the first step, increasing their data variance with new videos and links between them.

2.8.2.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis is used to visually summarise and highlight the important aspects of the data to be investigated. Although EDA may or may not utilize any statistical model, it provides insight into what the data holds beyond hypothesis testing and predictive modeling. Exploratory data analysis (EDA) is a valuable process developed in the modern era that aids in gaining familiarity with any given dataset before building a Machine Learning Algorithm or model.

Based on Heckert, N. A., Filliben, J. J., Croarkin, C. M., Hembree, B., Guthrie, W. F., Tobias, P., & Prinz, J. (2002), The EDA includes a variety of techniques for the following purposes: to gain a deeper understanding of the current dataset, to discover optimal factor settings, to develop penurious predictive models, to test fundamental assumptions, to identify outliers and anomalies in the dataset, to extract useful variables and information and to reveal hidden structures.

2.8.2.3 Sentiment Analysis

In the article about opinion mining on YouTube by Severyn, A., Uryupina, O., Plank, B., Moschitti, A., and Filippova, K. (2014), the opinion mining approach they used focuses on the creation of classifiers to predict comment type and polarity. Traditionally, such classifiers have relied on bag-of-words and other complex features. They defined a baseline feature vector model and a novel structural model based on kernel approaches. Instead of using traditional bag-of-word representation, they added features from a sentiment lexicon and characteristics that quantify the negation in the comment to the usual bag-of-words representation. The following feature groups are used by our model (FVEC) to encode each document:

i) Word n-gram - Over lower-cased word lemmas, they compute unigrams and bigrams, where binary values are utilized to signify the presence/absence of a given item.

ii) Lexicon – They used the MPQA Lexicon and the Hu and Liu Lexicon since both lexicon techniques are manually produced freely available sentiment lexicons. They utilize the number of terms identified in the comment that have positive and negative sentiment as a characteristic for each of the lexicons.

iii) Negation - allows a more thorough study of the element of negation contained in the comments.

iv) Video Concept - Analyse the cosine similarity between a comment and the video's title/description. The majority of the videos have a title and a brief description, which may be used to decode the topicality of each comment by looking at how closely they overlap.

For the structural model, they chose shallow structures with simpler and more durable components (STRUCT model). Their shallow tree structure specifically is a two-level syntactic hierarchy comprised of word lemmas (leaves) and part-of-speech tags, which are further divided into chunks.

In the modeling process, they use supervised methods, such as SVM, to perform OM. The goal is to train a model that can determine the sentiment and kind of each comment automatically. They use the one-vs-all technique to create a multiclass classifier for this. For each of the classes, a binary classifier is trained, and the predicted class is determined by selecting the class with the highest prediction score.

Meanwhile, in another article by Rinaldi, E., & Musdholifah, A. (2017, November) about opinion mining on Indonesian comments of youtube videos using FVEC-SVM, the features of cleaned comments are extracted using FVEC, such as TF-IDF of n-gram words, negation counting, and cosine similarity. Finally, SVM is used to classify the comment based on its features. However, the Extraction Features process does not include the utilization of the lexicon.

2.8.3 Findings and Evaluation

Based on YouTube data that was collected for 205 days and included 40 950 videos in Khanam, S., Tanweer, S., & Khalid, S. S. (2021) research, the normalization of all the numerical attributes is conducted by implying Robust Z-score normalization to ensure that any potential outliers do not affect the normalization.

The following trends and patterns were discovered in their findings based on the data features:

- Views: According to their research, 91 percent of trending videos have less than 5 million views, while 71% have 1.5 million views.
- Uniqueness: Only 6351 videos out of a total of 40 950 are unique, as some videos are trending for more than one day.
- Comments: They discovered that 93 percent of trending videos had less than 25 000 comments, and 67 percent have less than 4000 comments after studying the comments.
- For 84 percent of the videos, the 'like' count is 100 000, while for 69 percent of the videos, it is 40 000.
- Most common word: trending videos with 1 million or more views have titles that are between 35 and 55 characters long, with the most common terms being 'Trailer,' 'Official Video,' 'Audio,' and 'New.'
- Data correlation: We discovered a high positive link between the trending video's likes and views, as well as a somewhat lesser correlation between the number of comments and dislikes.

In the article written by Airoldi, M., Beraldo, D., & Gandini, A. (2016) about the exploratory investigation of music on YouTube, they concluded that the relatedness between two music videos that may be regarded as crowd-generated music categories has been proven to be determinant in a large sample of YouTube music videos. They believe that the closely linked groupings of related music videos that emerge from a computational examination of the network's community structure could be thought of as rough miniatures of music categories as they arise "from the bottom up.". However, interestingly they added that there is a kind of situational dependence in which listeners ignore the stereotype observed in the research and chose their music preference and genre for purely aesthetic and stylistic purposes. They stated that 10% of their sample follows the logic mentioned above.

In an article about opinion mining on YouTube by Severyn, A., Uryupina, O., Plank, B., Moschitti, A., & Filippova, K. (2014), they stated that Because the STRUCT model can create structural patterns of sentiment, their STRUCT model is more accurate compared to FVEC model. There are some cases that which The FVEC bag-of-words model misclassifies positive evaluation because the model misinterpreted two positive expressions to outweigh a single negative expression. In contradiction to the STRUCT model, the structural model can accurately classify the comment as negative by identifying the product of interest and associating it with the negative expression via a structural feature.

Another article about opinion mining on YouTube comment by Rinaldi, E., & Musdholifah, A. (2017, November), used a 10-fold cross-validation method to evaluate each kernel function in the FVEC-SVM, they stated that FVEC-SVM that uses linear kernel function has the highest accuracy with 63.99%. They concluded that FVEC-SVM outperformed other kernel functions.

2.8.4 Visualization

In this section, observation of the visualization is used to provide better research insights. These visualizations are presented to help readers understand the context of the findings.

2.8.4.1 Using Histograms

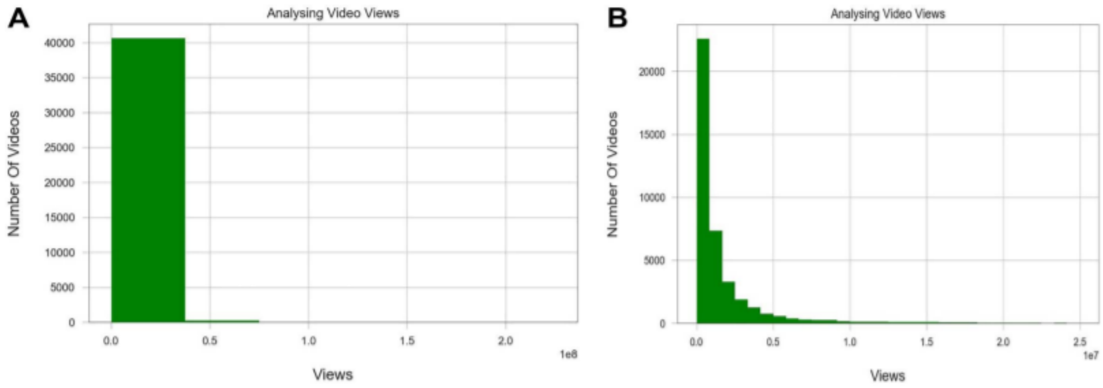


Figure 2.1: Views associated & data distribution with trending videos

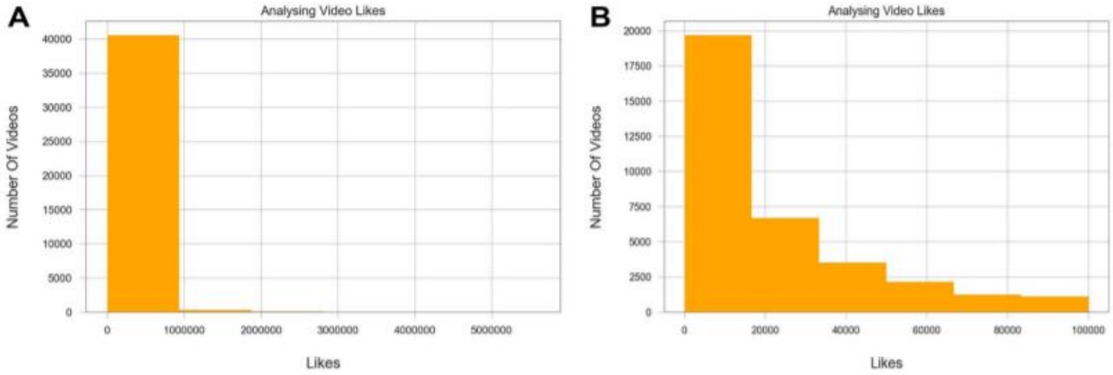


Figure 2.2: Video Likes Analysis and its Data Distribution

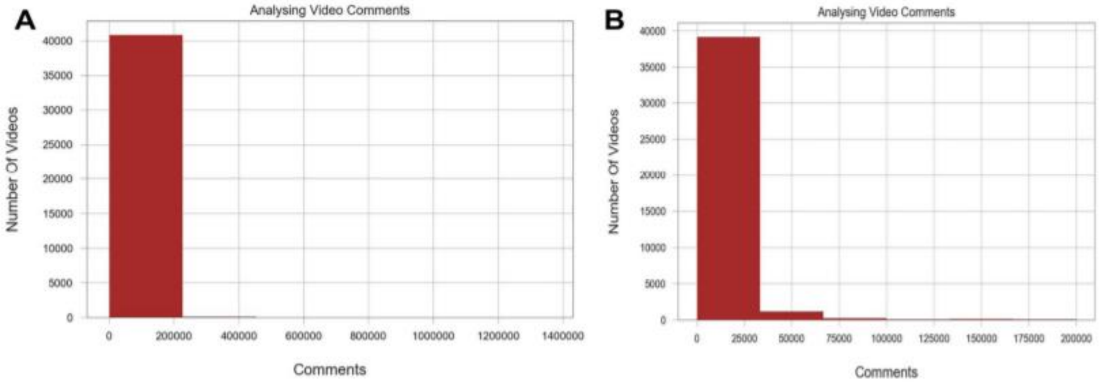


Figure 2.3: Video Comments Analysis and its Data Distribution

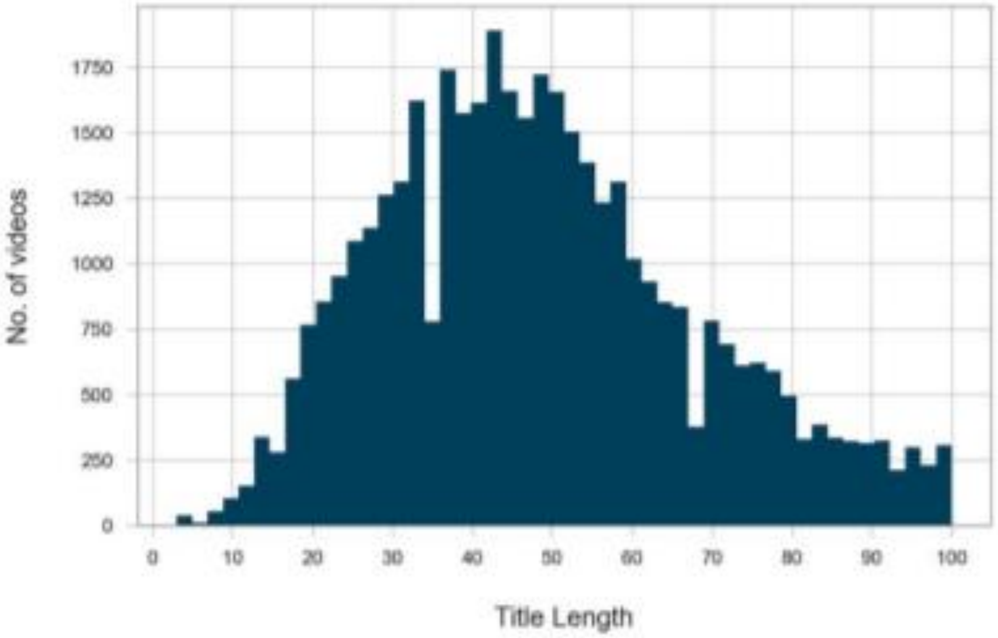


Figure 2.4: Data Distribution for Title Length

Figure 2.2, 2.3, and 2.5 shows the histogram visualizations that are used to show the distribution of variables in the dataset.

2.8.4.2 Using Scatter Plot

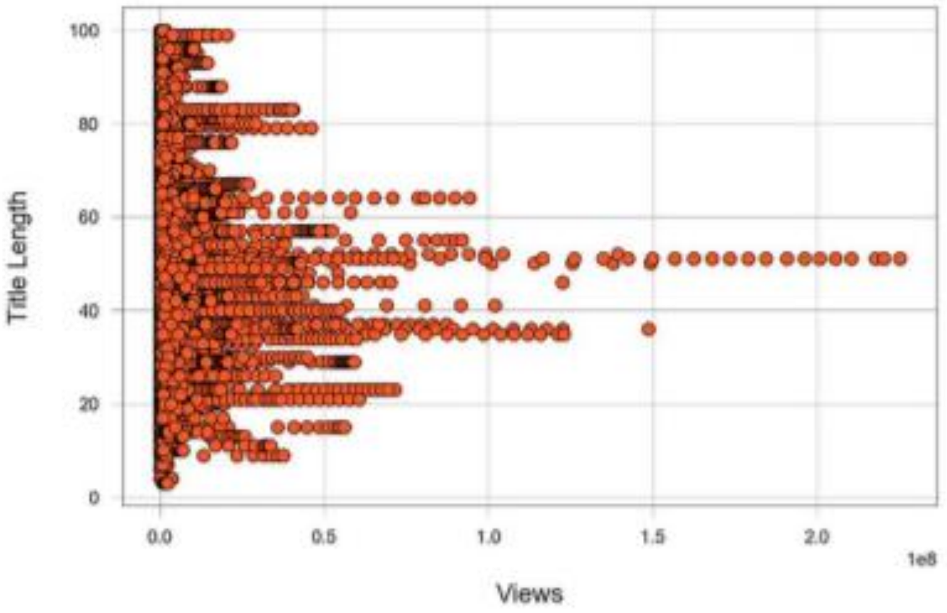
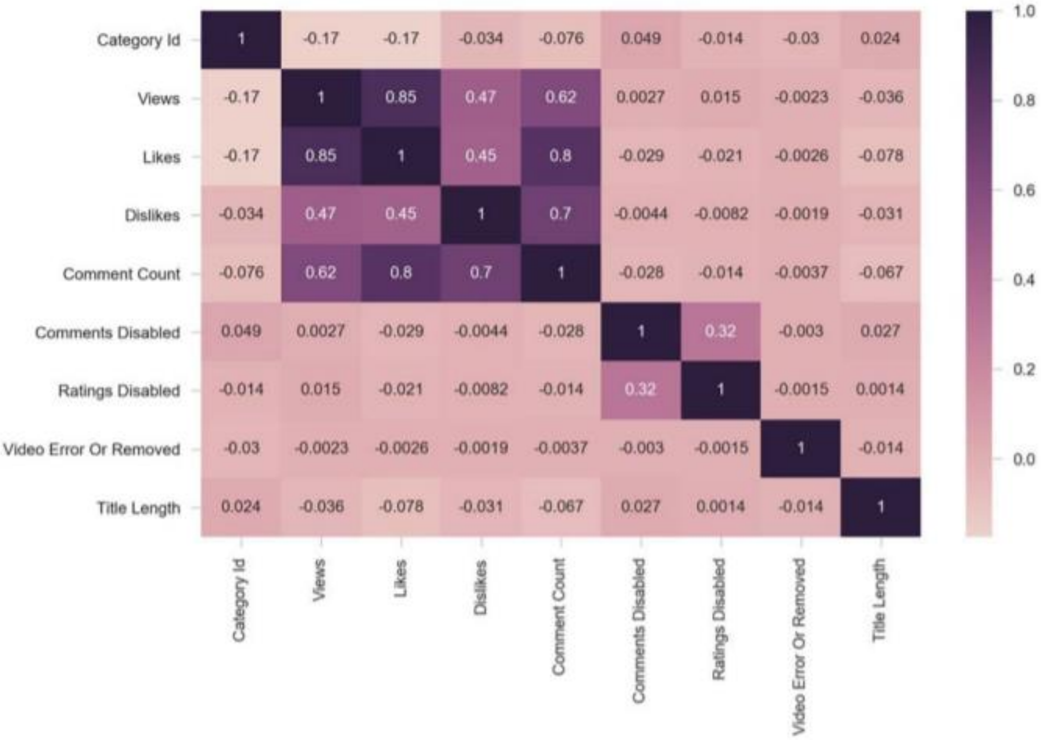


Figure 2.5: Scatter Plot for Title Length vs Views

Figure 2.5 shows that a Scatter plot is used to observe and show relationships between title length of video and views.

2.8.4.3 Using Heat Map



	Category Id	Views	Likes	Dislikes	Comment Count	Comments Disabled	Ratings Disabled	Video Error Or Removed	Title Length
Category Id	1	-0.17	-0.17	-0.034	-0.076	0.049	-0.014	-0.03	0.024
Views	-0.17	1	0.85	0.47	0.62	0.0027	0.015	-0.0023	-0.036
Likes	-0.17	0.85	1	0.45	0.8	-0.029	-0.021	-0.0026	-0.078
Dislikes	-0.034	0.47	0.45	1	0.7	-0.0044	-0.0082	-0.0019	-0.031
Comment Count	-0.076	0.62	0.8	0.7	1	-0.028	-0.014	-0.0037	-0.067
Comments Disabled	0.049	0.0027	-0.029	-0.0044	-0.028	1	0.32	-0.003	0.027
Ratings Disabled	-0.014	0.015	-0.021	-0.0082	-0.014	0.32	1	-0.0015	0.0014
Video Error Or Removed	-0.03	-0.0023	-0.0026	-0.0019	-0.0037	-0.003	-0.0015	1	-0.014
Title Length	0.024	-0.036	-0.078	-0.031	-0.067	0.027	0.0014	-0.014	1

Figure 2.6: Heatmap Correlation

Figure 2.6 is a Heat map visualization that is used to find the correlation between the various YouTube attributes present including views, likes, dislikes, rating, title, etc., and hence to perform a bivariate analysis.

2.8.4.4 Using WordCloud

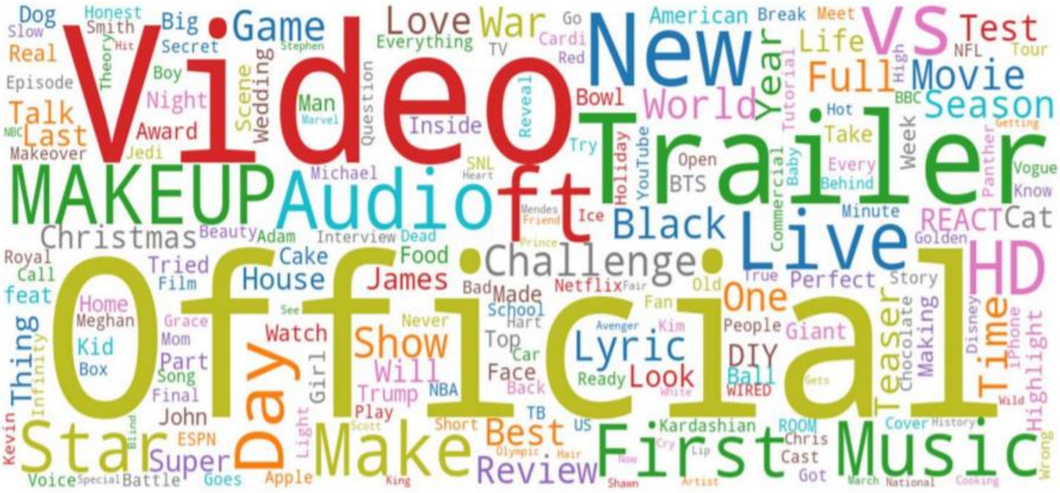


Figure 2.7: WordCloud for Top 30 most Common Words

Figure 2.7 shows the implementation of *WordCloud* to portray the Top 30 most common words used in the title of trending videos.

2.8.4.5 Using Bar Chart

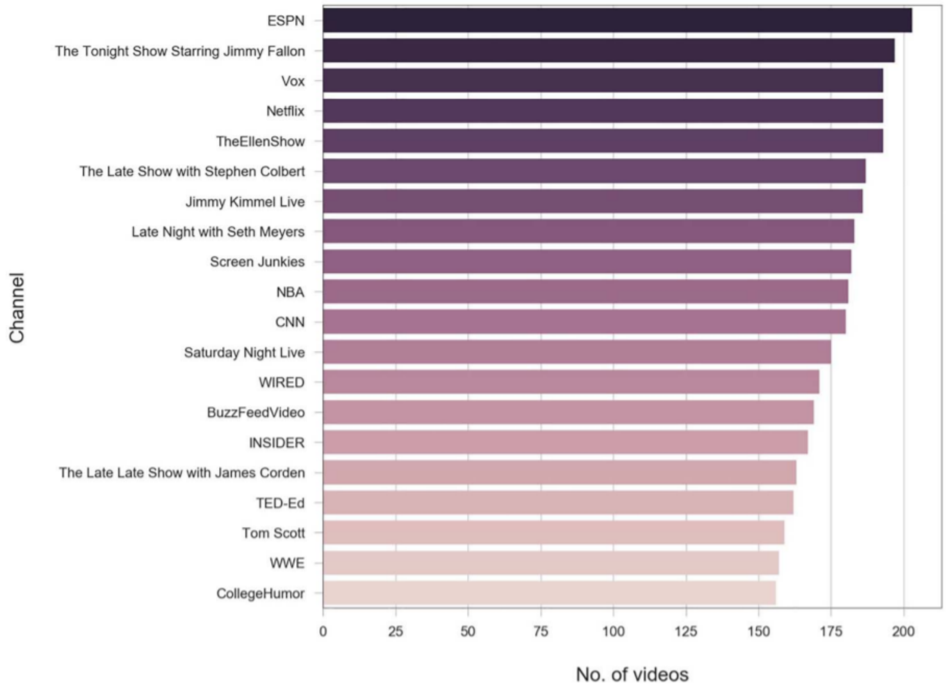


Figure 2.8: Bar Chart for Top 20 Channel

Figure 2.8 shows how Bar Chart is used to visualize the number of videos that the top 20 channels has uploaded.

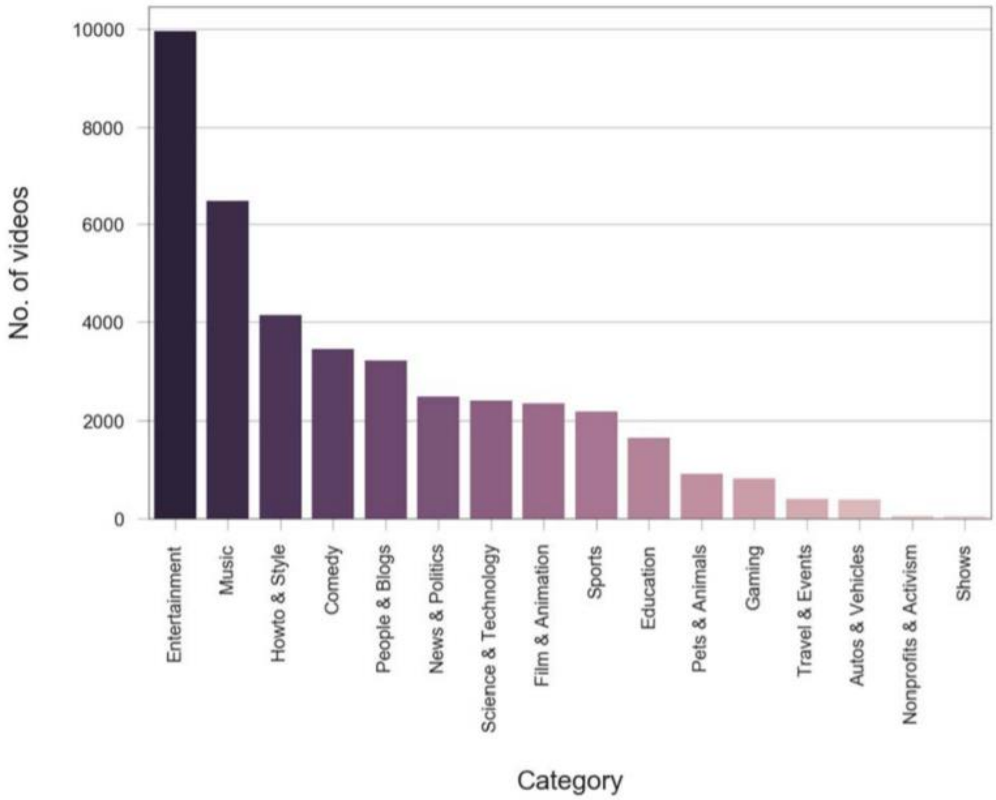


Figure 2.9: Bar Chart for Number of Videos based on Category

Figure 2.9 shows the number of videos based on the category of videos using a bar chart.

2.9 Direction of the Proposed Project

After a thorough reading of reviewed papers, papers from Airoldi, M., Beraldo, D., & Gandini, A. (2016), Severyn, A., Uryupina, O., Plank, B., Moschitti, A., & Filippova, K. (2014), Rinaldi, E., & Musdholifah, A. (2017, November) and Khanam, S., Tanweer, S., & Khalid, S. S. (2021) has deeply inspired me to complete the project. The topic discussed in the 4 papers is opinion mining (sentiment analysis), and exploratory analysis on YouTube.

Proposed Data Collection method: the dataset used will be a pre-crawled dataset from the Kaggle website, “YouTube Trending Video Dataset (updated daily)” uploaded by Rishav Sharma because of the high dimensionality of the dataset with collected data from 11 different countries which are India, USA, Great Britain, Germany, Canada, France, Russia, Brazil, Mexico, South Korea, and Japan respectively. The dataset is consisted of up to 200 listed trending videos per day. The dataset is fetched is separated by regions. The list of data attributes is as below:

i) Video ID

ii) Video title

iii) Date published

iv) Channel ID

v) Channel title

vi) Category ID

vii) Trending date

viii) Video tags

ix) Count of views

x) Likes

Proposed Data Pre-processing method; use Jupyter Notebook for code documentation. The cleaning dataset will be adjusted according to usability for Exploratory Data Analysis and Sentiment Analysis in the later steps.

The dataset will be pre-processed by removing punctuations and special characters, treating missing values, data segmentation and normalization of out-of-vocabulary words in the title/description and comments in the datasets, and conversion of lowercase and data annotation.

In addition, feature extraction and feature selection will be conducted to reduce the data dimension and further pull relevant data to use for hypothesis testing.

The project will then move to the data division process where data is split into training sets and testing sets.

For the modeling phase, the chosen analysis for the proposed project is firstly the project will undergo unsupervised learning using LDA modeling to figure out topics for each video. Later, the obtained topics will be trained for supervised learning to evaluate the accuracy of the LDA model.

The performance of the analysis will be evaluated by a classification report containing the confusion matrix, accuracy, precision, precision, recall, and F1-score.

Finally, the findings of the project are visualized using scatter plots, pair plots, histograms, heat maps, and *WordCloud.*

CHAPTER 3: METHODOLOGY

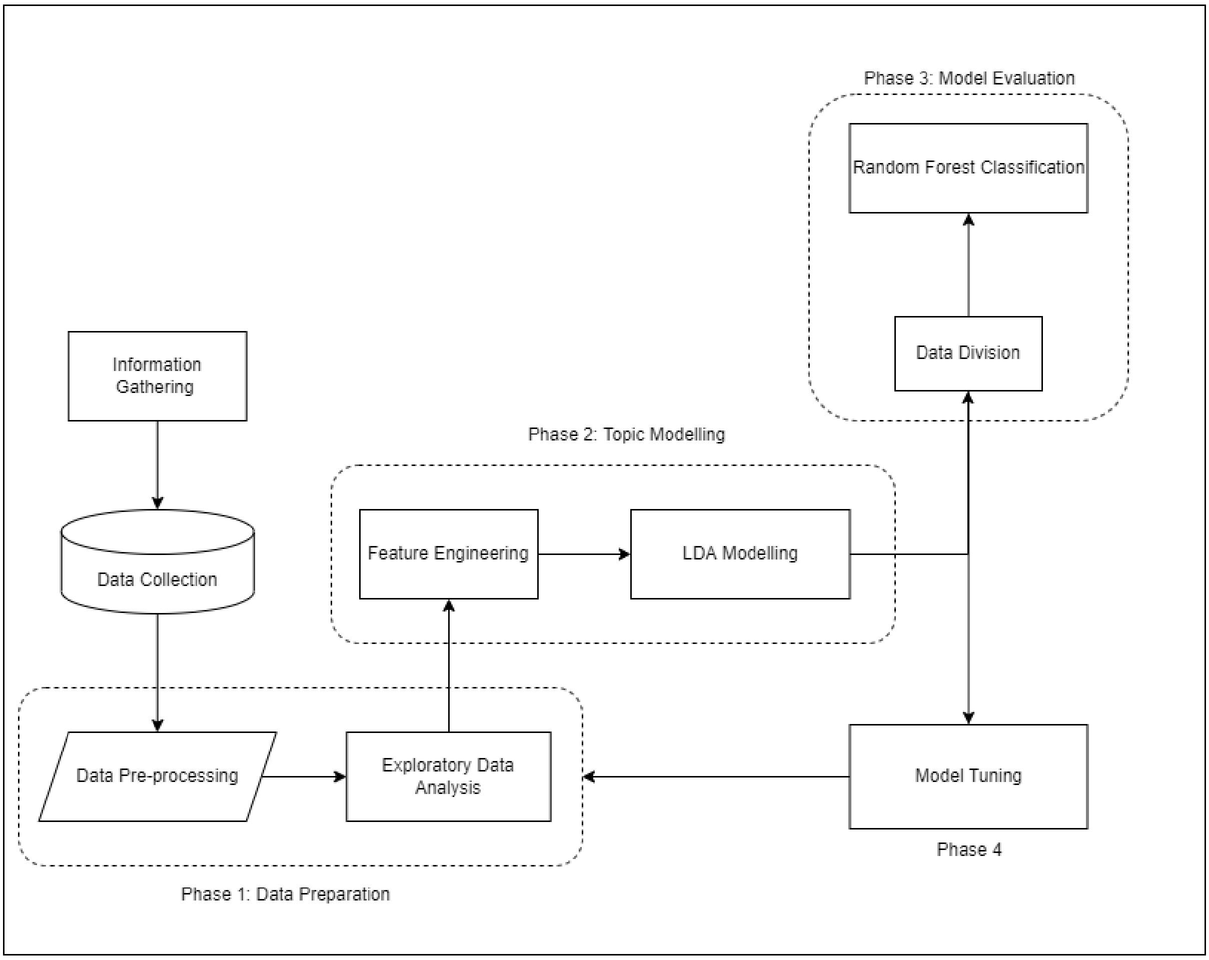


Figure 3.1: Detailed Workflow of the Project

Figure 3.1 above shows the detailed workflow of the project. This chapter will discuss the detailed methodology based on *Figure 3.1*.

3.1 Information Gathering:

i) Gather information about text classification techniques and time series analysis.

ii) Study the social and behavioral science for a better understanding of the relationship between the social network and user behavioral patterns.

iii) Reading similar research papers relating to YouTube

3.2 Data Collection:

i) For this project, the datasets are essentially extracted from YouTube API.

ii) The project is conducted using Python language.

z

Kaggle.com is a crowd-sourced platform that allows developers and data scientists to write, share codes, and host datasets. The purpose of Kaggle is to serve as a platform to attract data scientists all around the world, nurture, and challenges them to solve data science, machine learning, and predictive analytics problems according to Usmani Z. (2017)

Then, the dataset acquired undergoes the assessment of data quality. The assessment process of data quality is important to ensure scraped dataset using a scraping bot is suitable and testable according to project requirements. This process is performed by checking the extracted data by scraping the bot that has scraped the correct elements and fields. In this project, the correct elements and fields are correct comments from the intended link provided in the source code.

- Video ID: ensure the correct path of video ID is extracted.
- Video title: ensure the video title is not merged
- Date published: Ensure timestamp data extracted are true to their published date
- Channel ID: ensure all extracted rows have channel ID filled.
- Channel Title: Different videos may have the same owner ID. Ensure that video details are not collapsed

- Category ID: Ensure all video has their categories
- Trending date: Ensure timestamp data extracted are true to their trending date
- Video tags: Ensure tags are extracted to their respective column
- Count of views: Ensure the view count is accurate according to the YouTube page
- Likes and Dislike: Ensure likes and dislikes are distributed to their respective columns.

3.3 Phase 1: Data Preparation

For documentation containing notes, live codes, equations, and visualizations, the project is being documented using the classic Jupyter Notebook. Jupyter Notebook is an online application for creating and sharing computational documents. It offers a user-friendly, efficient, document-focused experience for documentation purposes.

The dataset is imported to Jupyter Notebook along with relevant libraries that will be mentioned in the later process.

3.3.1 Data Pre-processing

For data pre-processing, the package tools used for this process are Pandas, Numpy, and Klib from Python libraries. In the data pre-processing, the numerical and categorical feature is explored for better data understanding. The next step is to inspect whether the dataset has missing values or not.

Then, the missing values are treated. Next, outliers are observed and treated to avoid lower accuracy of model training later. The dataset is normalized for more consistency, which improves the model's ability to forecast results. After exploring numerical and categorical features, some features are better converted to be able to feed into machine learning algorithms later. As some columns are contained in texts, it consists of human language which has an abundance of stop words. By getting rid of these words, text columns are more focused on the key information by eliminating the low-level information. Then, punctuation and special characters are removed from columns containing texts to avoid complicacy when building the model later.

3.3.2 Exploratory Data Analysis

Exploratory Data Analysis is used for more comprehensive views of what given data in the dataset is about. By doing so, the best practice of machine learning and the roadmap of the project is identified. This is done by using python packages such as Pandas and NumPy, along with statistical methods and data visualization packages. The purpose of doing Exploratory Data Analysis is to provide better insights from statistical evidence in Exploratory Data Analysis and utilize it to determine the results in findings.

The analysis involved are:

- Univariate Analysis
- Bivariate Analysis
- Time-series Analysis
- Multivariate Analysis

3.3.2.1 Univariate Analysis

Univariate analysis is the simplest form of analysis which involves summarization and pattern of only one "Uni" variable in the dataset dimension. This analysis is conducted to discover the meaning of each column of data, whether it is categorical or continuous, or independent or dependent on other variables in the dataset

3.3.2.2 Bivariate Analysis

Compared to univariate analysis, Bivariate analysis is a correlation analysis that is conducted to gather insights into the causal relationship between two "Bi" variables.

3.3.2.3 Time Series Analysis

Because the dataset contains timestamps, time series analysis is necessary to discover hidden insights based on time intervals. Time series analysis is a specific approach to analyzing a set of data points accumulated over an extended period. Time series analysis is done by manipulating the record of the data points over a set period that is extracted from video published time.

3.3.2.4 Multivariate Analysis

Multivariate analysis is the statistical analysis of correlations between several measurements made on each experimental unit and where the relationship between multivariate measurements and their structure is crucial to understanding the experiment. It is done to find suitable features to feed into machine learning models, or to decide whether the data type should be transformed or not in the feature engineering process.

3.4 Phase 2: Topic Modeling

The purpose of using topic modeling is to discover specifically what topic is used in video titles, descriptions, and tags to achieve higher interaction. In phase 2, the combined text of video title, descriptions, and tags is filtered in the feature engineering process and later fed into the LDA model.

3.4.1 Feature Engineering

The process of feature extraction and selection is to reduce the possibility of overfitting when building models to determine the weighted polarity of the data. Based on Ghojogh, B., Samad, M. N., Mashhadi, S. A., Kapoor, T., Ali, W., Karray, F., & Crowley, M. (2019)'s paper on feature selection and feature extraction in pattern analysis, feature selection, and feature extraction is used to assist the models perform better, valuable information can be in the form of better data representation or better class discrimination.

3.4.1.1 Feature Extraction

The feature extraction method used for this project is the n-gram method. N-gram is the count of *n* sequence of words where *n* could be one-word level, Unigram, and two-word level, Bigram. In this project, both unigram and bigram are applied.

N-gram method is used to find out the correlation between trending videos and word occurrence in the video title, tags, and video description.

3.4.1.2 Feature Selection

The feature Selection method is used for reducing the dimension of data to feed into the LDA model later. Reducing the dimension of the data helps to improve the accuracy of the model and reduce the execution time when building and tuning the model. Feature Selection includes the process of lemmatization, removing stop words, removing non-readable words, and removing null rows in the dataset. An example of non-readable words is single characters or a mesh of unreadable characters after the lemmatization process.

This is done to ensure only meaningful words in fed into the LDA model. The lemmatized document is then relayed to create a dictionary by assigning an integer value to each word in the document. Creating a dictionary of the lemmatized document is necessary to create a corpus Bag-of-Word needed for topic modeling.

3.4.2 LDA Modeling

For the LDA modeling process, there are 2 steps involved:

- Building LDA base model
- Compute the perplexity and coherence score
- Assign each document to its predicted topic

3.4.2.1 Building LDA Base Model

The LDA base model is built by assigning a reasonable initial number of topics where each topic is made up of several keywords, and each term has a specific weight in the topic.

3.4.2.2 Compute Perplexity and Coherence Score

The LDA model is partially evaluated using perplexity and coherence scores. Perplexity is a statistical measure of how well a probability model predicts a sample in the LDA model. Typically, the lower the perplexity, the better the LDA model is produced.

Topic modeling uses the coherence score to gauge how comprehensible the topics are to people. Generally, the higher the coherence score, the better the LDA model is built.

3.4.2.3 Assign Each Document to its Predicted Topics

Each row of the document in the dataset is assigned and labeled to its predicted topic for later use in supervised learning.

3.5 Phase 3: Model Evaluation

To achieve better accuracy of the topic used in the video title, descriptions, and tags, the model is evaluated by a cross-validation process using supervised learning for Bag-Of-Word and TF-IDF model evaluation.

3.5.1 Data Division

In this process, the labeled dataset is split into two parts, the training set, and the testing set. The training set is a subset used to train the model and the testing set is a subset to test a trained model. The training set is sliced up to 80% of the dataset and the testing set takes 20%.

3.5.2 Random Forest Classification

Random Forest Classification is built for Bag-Of-Word model evaluation and TF-IDF model evaluation. The evaluation is observed from the value of precision, recall F-1 score, and support obtained after the cross-validation process from training and testing sets.

3.6 Phase 4: Model Tuning

To increase the performance of the LDA model, there are 2 steps of model tuning:

- Build LDA Mallet Model.

- Find the optimal number of topics, *k*.

3.6.1 Build LDA Mallet Model

Mallet's LDA algorithm uses Gibbs Sampling which usually returns better results for long text documents. However, emitting better accuracy, the time consumed to build the LDA Mallet model is higher than the general LDA model. The LDA Mallet model is only chosen if it returns a better coherence score than the normal LDA model.

3.6.2 Find the Optimal Number of Topics, *k*

Hyperparameter is manipulated to determine the optimal number of topics, *k*. This is done by building many LDA models and then plotting a computed coherence score graph for a range of *k*. Generally, the *k* value is determined by the model that has the highest coherence score before flattening out.

3.7 Conclusion

In summary, this chapter is explained based on the project overview in *Figure 3.1* where there is a data understanding phase along with 4 major phases involved which are:

Phase 1 – Data preparation

Phase 2 – Topic Modeling

Phase 3 – Model Evaluation

Phases 4 – Model Tuning

In Phase 3: Model Evaluation, until the optimal model that gives the greatest accuracy is figured out, phase 4 which is Model Tuning will loop back to Phase 1: Data Preparation and Phase 2: Topic Modeling. The purpose of the loopback process is to avoid noisy data being fed into machine learning algorithms by treating outliers and reducing the dimension of data.

Lastly, all data results are visualized by their respective step. Data visualization is observed and inferred to acquire findings. Discussion of the findings is elaborated in Chapter 5. The data is visualized in various forms such as histograms, correlation plots, heatmaps, and *WordCloud.*

CHAPTER 4: IMPLEMENTATION

In this chapter, the project is conducted based on stated steps in the project pipeline overview. Then, each phase will be discussed in detail. The chapter will consist of a data understanding phase along with 4 major phases stated in Figure 4.1 below:

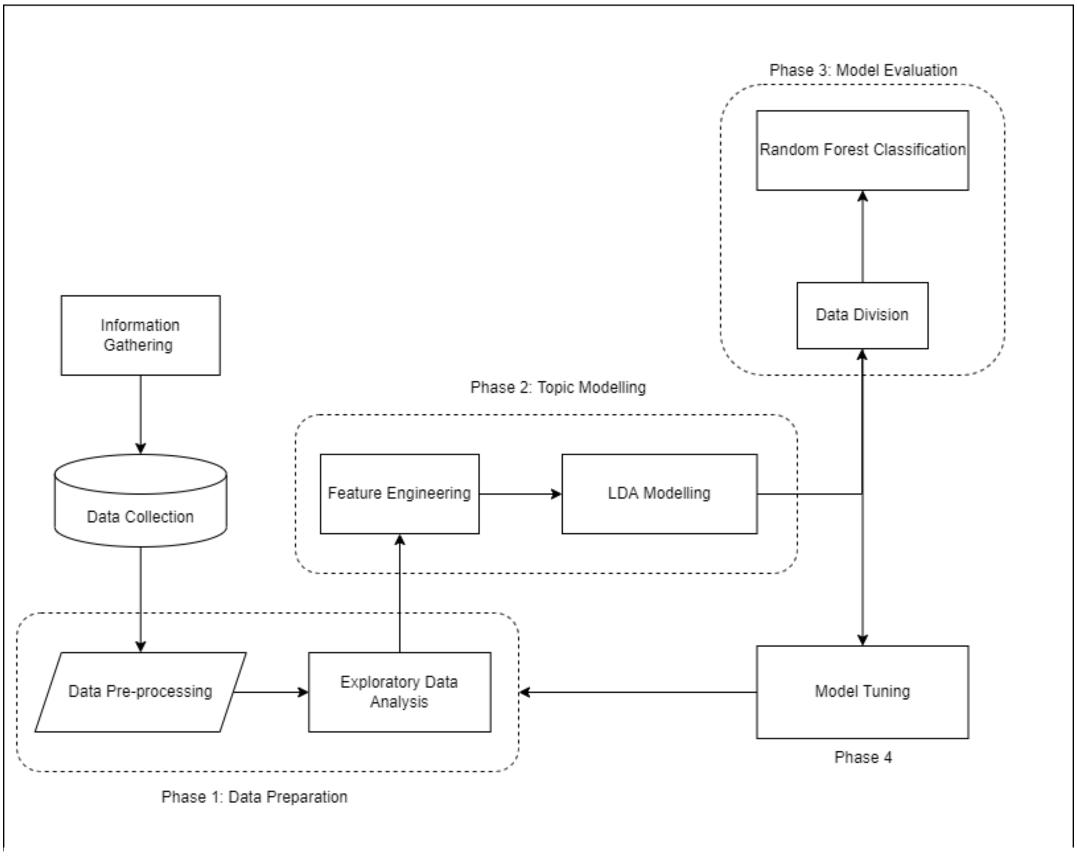
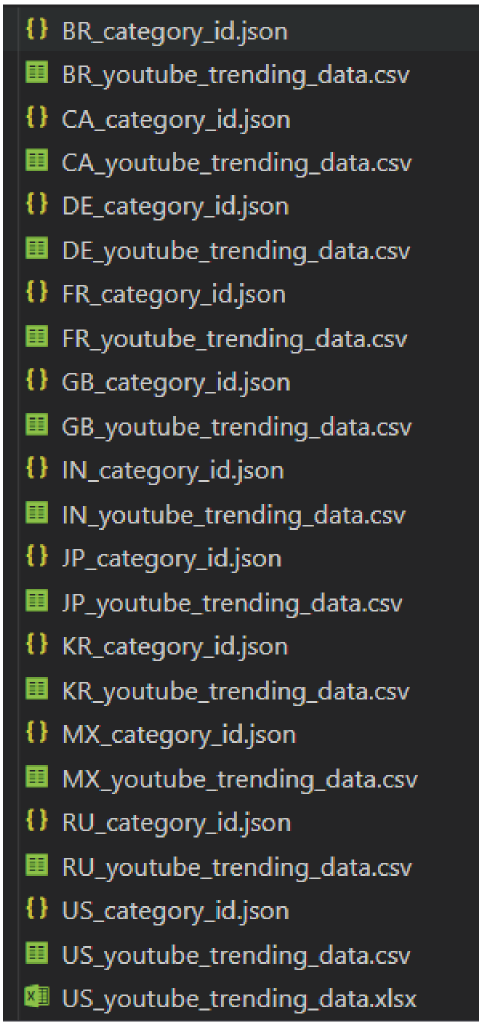


Figure 4.1 Project Pipeline Overview

Phase 1, Data preparation consists of Data Pre-processing steps and Exploratory Data Analysis. Phase 2, Topic Modeling consists of the Feature Engineering step and LDA modeling. Phase 3, Model Evaluation consists of Data Division and Supervised Learning Random Forest Classification for evaluation of the LDA model. Lastly, Phase 4 Model Tuning.

4.1 Data Collection

Trending YouTube videos Dataset is downloaded from the link https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset. This dataset includes several months of data on daily trending YouTube videos. Data is included for the IN, US, GB, DE, CA, FR, RU, BR, MX, KR, and JP regions (India, USA, Great Britain, Germany, Canada, France, Russia, Brazil, Mexico, South Korea, and Japan respectively), with up to 200 listed trending videos per day where each region's data has its separate files. The category ID is stored in JSON files while video title, channel title, publish time, tags, views, likes, dislikes, description, and comment count are stored in CSV files. *Figure 4.2* below shows the original dataset obtained from the website.

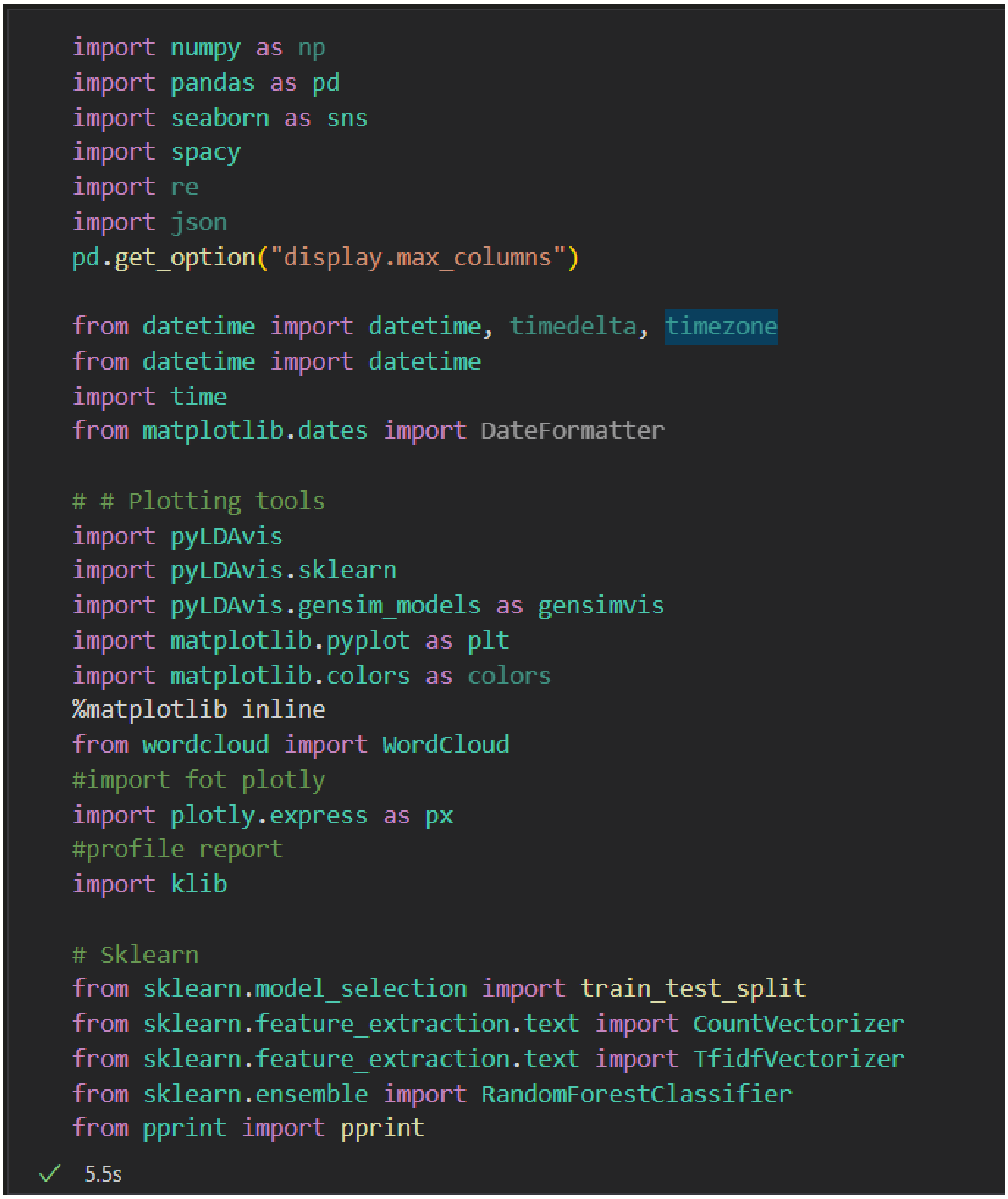


```
BR_category_id.json
BR_youtube_trending_data.csv
CA_category_id.json
CA_youtube_trending_data.csv
DE_category_id.json
DE_youtube_trending_data.csv
FR_category_id.json
FR_youtube_trending_data.csv
GB_category_id.json
GB_youtube_trending_data.csv
IN_category_id.json
IN_youtube_trending_data.csv
JP_category_id.json
JP_youtube_trending_data.csv
KR_category_id.json
KR_youtube_trending_data.csv
MX_category_id.json
MX_youtube_trending_data.csv
RU_category_id.json
RU_youtube_trending_data.csv
US_category_id.json
US_youtube_trending_data.csv
US_youtube_trending_data.xlsx
```

Figure 4.2: Original Dataset

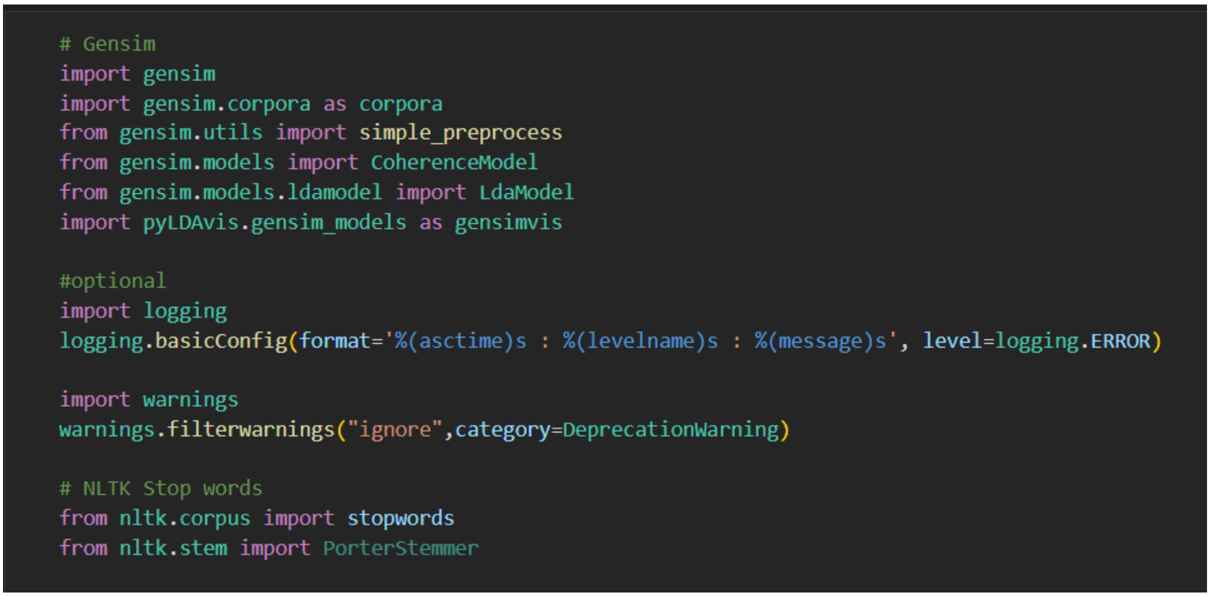
4.2 Phase 1: Data Preparation

The project will be conducted using Python language version 3.9.1.2. The documentation of the project which includes the line of code used, results, visualizations, and findings are saved in the Jupyter Notebook (.ipynb File) as “FYP2 Implementation. ipynb”. *Microsoft Visual Studio* is used to connect with Jupyter Notebook. All relevant Python packages throughout this project are listed in Figure 4.3 and Figure 4.4 below:



```
import numpy as np
import pandas as pd
import seaborn as sns
import spacy
import re
import json
pd.get_option("display.max_columns")

from datetime import datetime, timedelta, timezone
from datetime import datetime
import time
from matplotlib.dates import DateFormatter

# # Plotting tools
import pyLDAvis
import pyLDAvis.sklearn
import pyLDAvis.gensim_models as gensimvis
import matplotlib.pyplot as plt
import matplotlib.colors as colors
%matplotlib inline
from wordcloud import WordCloud
#import fot plotly
import plotly.express as px
#profile report
import klib

# Sklearn
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from pprint import pprint
```

Figure 4.3: Python Package used Part 1

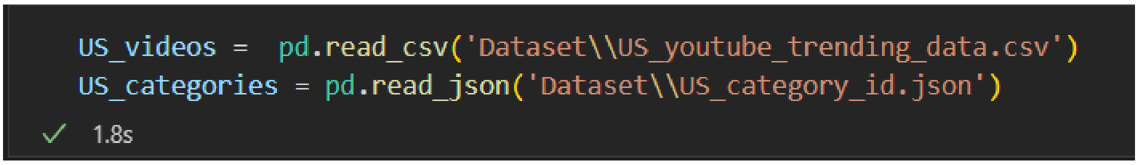


```python
# Gensim
import gensim
import gensim.corpora as corpora
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel
from gensim.models.ldamodel import LdaModel
import pyLDAvis.gensim_models as gensimvis

#optional
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.ERROR)

import warnings
warnings.filterwarnings("ignore",category=DeprecationWarning)

# NLTK Stop words
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

Figure 4.4: Python package used Part 2

4.2.1 Data Pre-processing

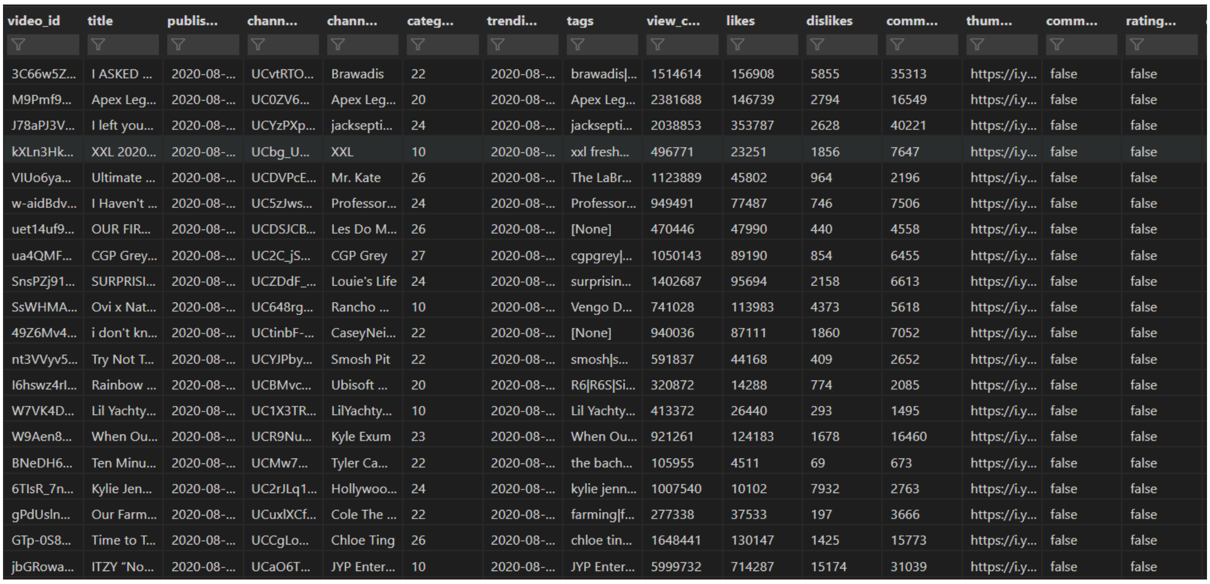
In this project, only videos in the US region are observed. Dataset is called into Jupyter Notebook after importing relevant libraries.



```python
US_videos =  pd.read_csv('Dataset\\US_youtube_trending_data.csv')
US_categories = pd.read_json('Dataset\\US_category_id.json')
```

✓ 1.8s

Figure 4.5: Importing Dataset

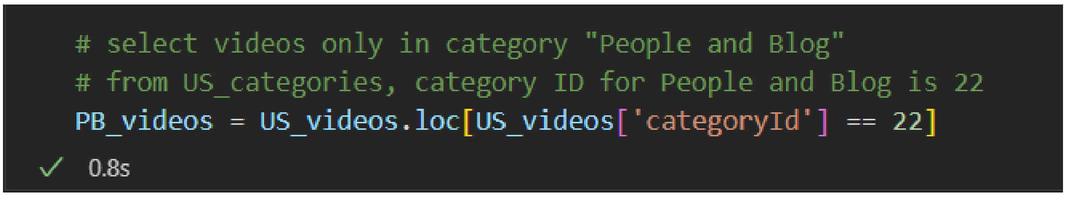


video_id	title	publis...	chann...	chann...	categ...	trendi...	tags	view_c...	likes	dislikes	comm...	thum...	comm...	rating...
3C66w5Z...	I ASKED ...	2020-08-...	UCvtRTO...	Brawadis	22	2020-08-...	brawadis|...	1514614	156908	5855	35313	https://i.y...	false	false
M9Pmf9...	Apex Leg...	2020-08-...	UC0ZV6...	Apex Leg...	20	2020-08-...	Apex Leg...	2381688	146739	2794	16549	https://i.y...	false	false
J78aPJ3V...	I left you...	2020-08-...	UCYzPXp...	jacksepti...	24	2020-08-...	jacksepti...	2038853	353787	2628	40221	https://i.y...	false	false
kXLn3Hk...	XXL 2020...	2020-08-...	UCbg_U...	XXL	10	2020-08-...	xxl fresh...	496771	23251	1856	7647	https://i.y...	false	false
VlUo6ya...	Ultimate ...	2020-08-...	UCDVPcE...	Mr. Kate	26	2020-08-...	The LaBr...	1123889	45802	964	2196	https://i.y...	false	false
w-aidBdv...	I Haven't ...	2020-08-...	UC5zJws...	Professor...	24	2020-08-...	Professor...	949491	77487	746	7506	https://i.y...	false	false
uet14uf9...	OUR FIR...	2020-08-...	UCDSJCB...	Les Do M...	26	2020-08-...	[None]	470446	47990	440	4558	https://i.y...	false	false
ua4QMF...	CGP Grey...	2020-08-...	UC2C_jS...	CGP Grey	27	2020-08-...	cgpgrey|...	1050143	89190	854	6455	https://i.y...	false	false
SnsPZj91...	SURPRISI...	2020-08-...	UCZDdF_...	Louie's Life	24	2020-08-...	surprisin...	1402687	95694	2158	6613	https://i.y...	false	false
SsWHMA...	Ovi x Nat...	2020-08-...	UC648rg...	Rancho ...	10	2020-08-...	Vengo D...	741028	113983	4373	5618	https://i.y...	false	false
49Z6Mv4...	i don't kn...	2020-08-...	UCtinbF-...	CaseyNei...	22	2020-08-...	[None]	940036	87111	1860	7052	https://i.y...	false	false
nt3VVyv5...	Try Not T...	2020-08-...	UCYJPby...	Smosh Pit	22	2020-08-...	smosh|s...	591837	44168	409	2652	https://i.y...	false	false
I6hswz4rl...	Rainbow ...	2020-08-...	UCBMvc...	Ubisoft ...	20	2020-08-...	R6|R6S|Si...	320872	14288	774	2085	https://i.y...	false	false
W7VK4D...	Lil Yachty...	2020-08-...	UC1X3TR...	LilYachty...	10	2020-08-...	Lil Yachty...	413372	26440	293	1495	https://i.y...	false	false
W9Aen8...	When Ou...	2020-08-...	UCR9Nu...	Kyle Exum	23	2020-08-...	When Ou...	921261	124183	1678	16460	https://i.y...	false	false
BNeDH6...	Ten Minu...	2020-08-...	UCMw7...	Tyler Ca...	22	2020-08-...	the bach...	105955	4511	69	673	https://i.y...	false	false
6TIsR_7n...	Kylie Jen...	2020-08-...	UC2rJLq1...	Hollywoo...	24	2020-08-...	kylie jenn...	1007540	10102	7932	2763	https://i.y...	false	false
gPdUsln...	Our Farm...	2020-08-...	UCuxlXCf...	Cole The ...	22	2020-08-...	farming|f...	277338	37533	197	3666	https://i.y...	false	false
GTp-0S8...	Time to T...	2020-08-...	UCCgLo...	Chloe Ting	26	2020-08-...	chloe tin...	1648441	130147	1425	15773	https://i.y...	false	false
jbGRowa...	ITZY “No...	2020-08-...	UCaO6T...	JYP Enter...	10	2020-08-...	JYP Enter...	5999732	714287	15174	31039	https://i.y...	false	false

Figure 4.6 Original Dataset imported as "US_videos"

Figure 4.6 above shows a portion of the original dataset for trending videos only in the United States (US) region.

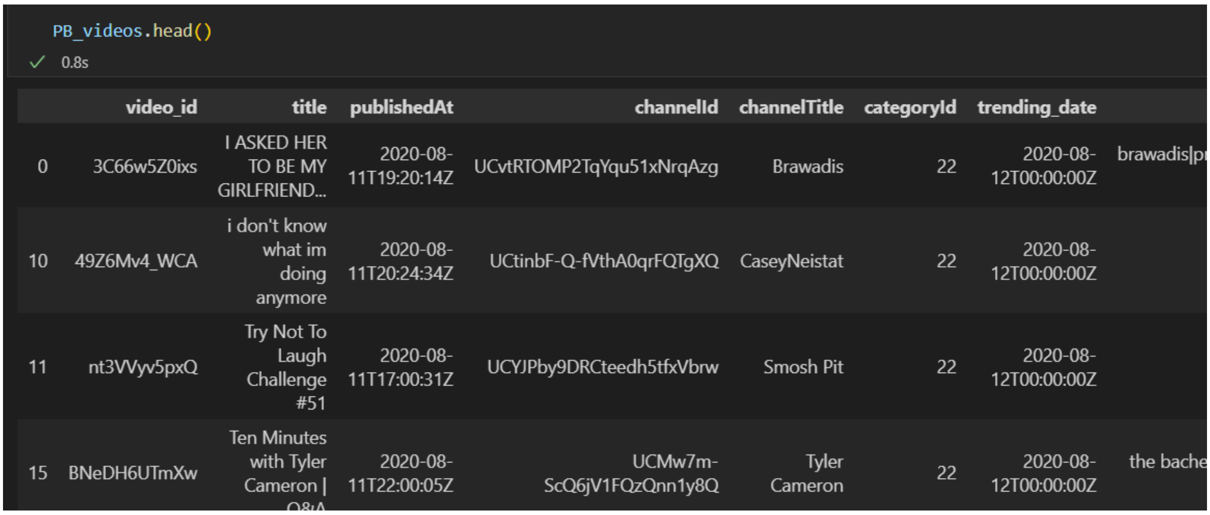
Specifically, this project analyzed videos only in the category “People and Blogs”. *Figure 4.7* below shows the filter code for only the category “People and Blog” which equals 22



```
# select videos only in category "People and Blog"
# from US_categories, category ID for People and Blog is 22
PB_videos = US_videos.loc[US_videos['categoryId'] == 22]
```

Figure 4.7:Filter to category People and Blog

Then, *Figure 4.8* is shown to ensure the dataset is filtered to only one category



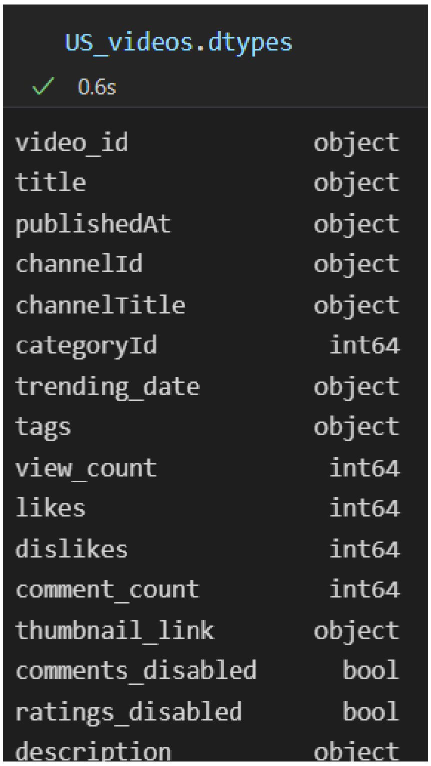
```
PB_videos.head()
```

✓ 0.8s

	video_id	title	publishedAt	channelId	channelTitle	categoryId	trending_date	
0	3C66w5Z0ixs	I ASKED HER TO BE MY GIRLFRIEND...	2020-08-11T19:20:14Z	UCvtRTOMP2TqYqu51xNrqAzg	Brawadis	22	2020-08-12T00:00:00Z	brawadis|pr
10	49Z6Mv4_WCA	i don't know what im doing anymore	2020-08-11T20:24:34Z	UCtinbF-Q-fVthA0qrFQTgXQ	CaseyNeistat	22	2020-08-12T00:00:00Z	
11	nt3VVyv5pxQ	Try Not To Laugh Challenge #51	2020-08-11T17:00:31Z	UCYJPby9DRCteedh5tfxVbrw	Smosh Pit	22	2020-08-12T00:00:00Z	
15	BNeDH6UTmXw	Ten Minutes with Tyler Cameron | Q&A	2020-08-11T22:00:05Z	UCMw7m-ScQ6jV1FQzQnn1y8Q	Tyler Cameron	22	2020-08-12T00:00:00Z	the bache

Figure 4.8: People and Blog dataset

To have a better comprehension of the People and Blog dataset, the data types are observed as shown in *Figure 4.9.*



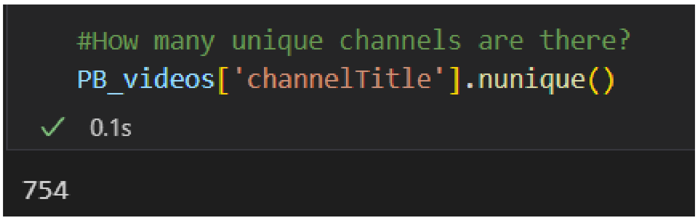
```
US_videos.dtypes
```

✓ 0.6s

```
video_id             object
title                object
publishedAt          object
channelId            object
channelTitle         object
categoryId            int64
trending_date        object
tags                 object
view_count            int64
likes                 int64
dislikes              int64
comment_count         int64
thumbnail_link       object
comments_disabled      bool
ratings_disabled       bool
description          object
```

Figure 4.9: Data types

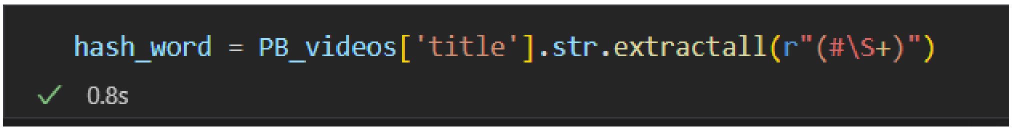
Unique channels are identified as shown below in *Figure 4.10*.



```
#How many unique channels are there?
PB_videos['channelTitle'].nunique()
✓ 0.1s

754
```

Figure 4.10: Unique Channels

Hashtag words are extracted as shown below in *Figure 4.11*



```
hash_word = PB_videos['title'].str.extractall(r"(#\S+)")
✓ 0.8s
```

Figure 4.11 Extraction of Hashtag Words

The missing values are identified using *Klib* plotting library as shown below in *Figure 4.12*

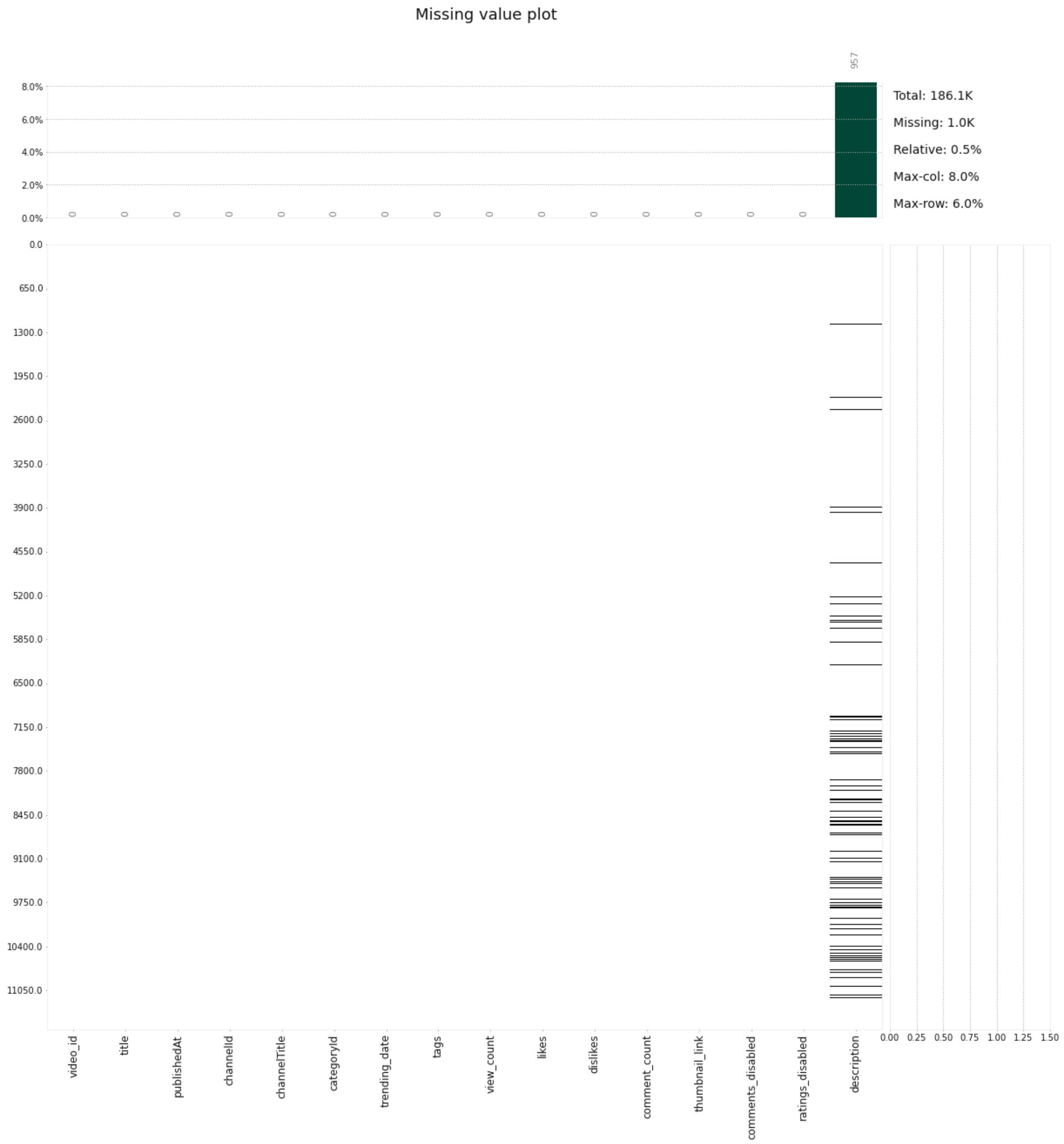
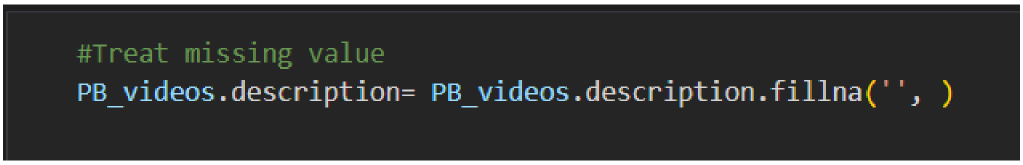


Figure 4.12: Missing Values Plot

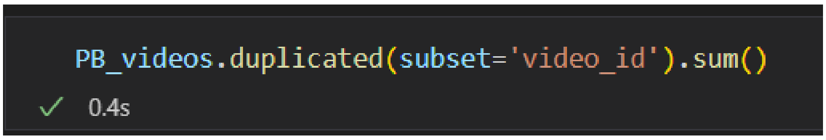
Based on *Figure 4.12* above, missing values are detected only in one column "description". Next, the missing values are treated with " " for text sampling later as shown below in *Figure 4.13.*



```
#Treat missing value
PB_videos.description= PB_videos.description.fillna('', )
```

Figure 4.13: Treating Missing Values

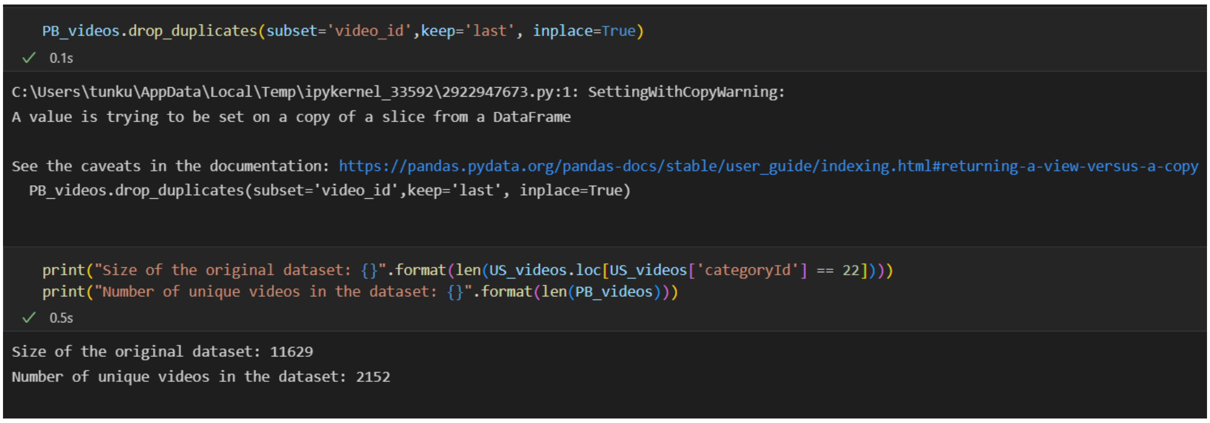
When doing data cleaning it is necessary to find and remove duplicated data. The process of finding duplicated data is based on column ‘video_id’ because every video ID in the dataset should be unique. The step is shown below in *Figure 4.14.*



```
PB_videos.duplicated(subset='video_id').sum()
✓ 0.4s
```

Figure 4.14: Finding Duplicated data

Based on the step above, the line of code returns a total of 9477 duplicated data contained in the dataset. The process of removing duplicated data is shown in *Figure 4.15* below.

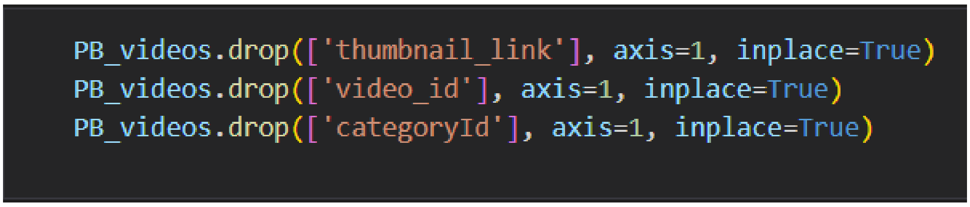


```
PB_videos.drop_duplicates(subset='video_id',keep='last', inplace=True)
✓ 0.1s
C:\Users\tunku\AppData\Local\Temp\ipykernel_33592\2922947673.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  PB_videos.drop_duplicates(subset='video_id',keep='last', inplace=True)


print("Size of the original dataset: {}".format(len(US_videos.loc[US_videos['categoryId'] == 22])))
print("Number of unique videos in the dataset: {}".format(len(PB_videos)))
✓ 0.5s
Size of the original dataset: 11629
Number of unique videos in the dataset: 2152
```

Figure 4.15: Removing Duplicated data

Based on the step above in *Figure 4.16*, the duplicated rows are dropped while keeping the last founded occurrence in the dataset as the last occurrence has the latest trending date.

Next, the columns that are meaningless to observe are dropped as shown in *Figure 4.16.*



```
PB_videos.drop(['thumbnail_link'], axis=1, inplace=True)
PB_videos.drop(['video_id'], axis=1, inplace=True)
PB_videos.drop(['categoryId'], axis=1, inplace=True)
```

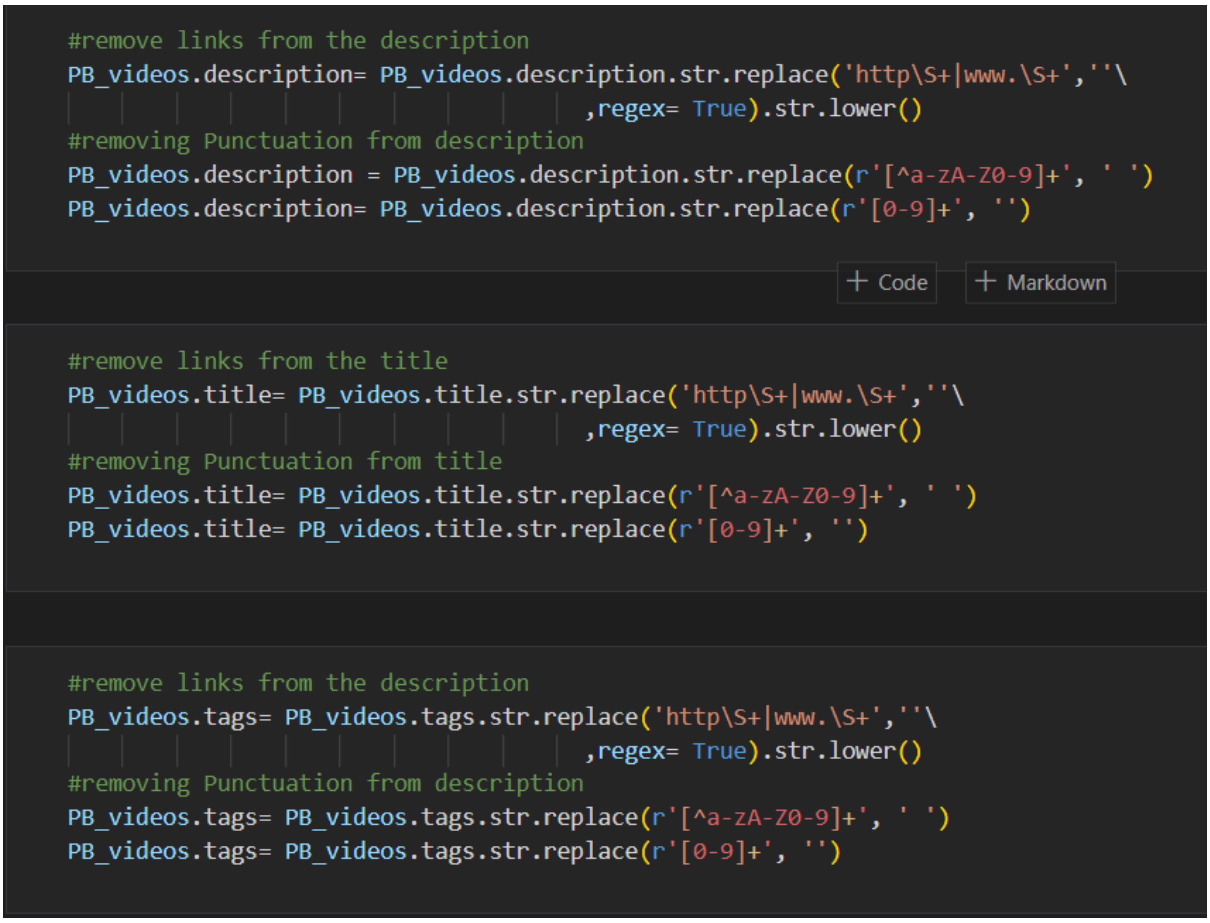
Figure 4.16: Drop Irrelevant Columns

Based on *Figure 4.16,* the column 'thumbnail_link' is dropped because it is a float data type that is unlearnable by machine learning. The column 'video_id' is dropped because the dataset has its unique index for each row. Lastly, column 'category_Id' is dropped because it provides a non-observable feature since all rows in the "People and Blog" dataset is filtered to category ID = 22.

Data cleaning of text columns is done to 3 columns which are:

- 'title' column
- 'description' column
- 'tags' column

Firstly, the link and punctuation from all 3 columns are removed as shown in *Figure 4.17 below.*



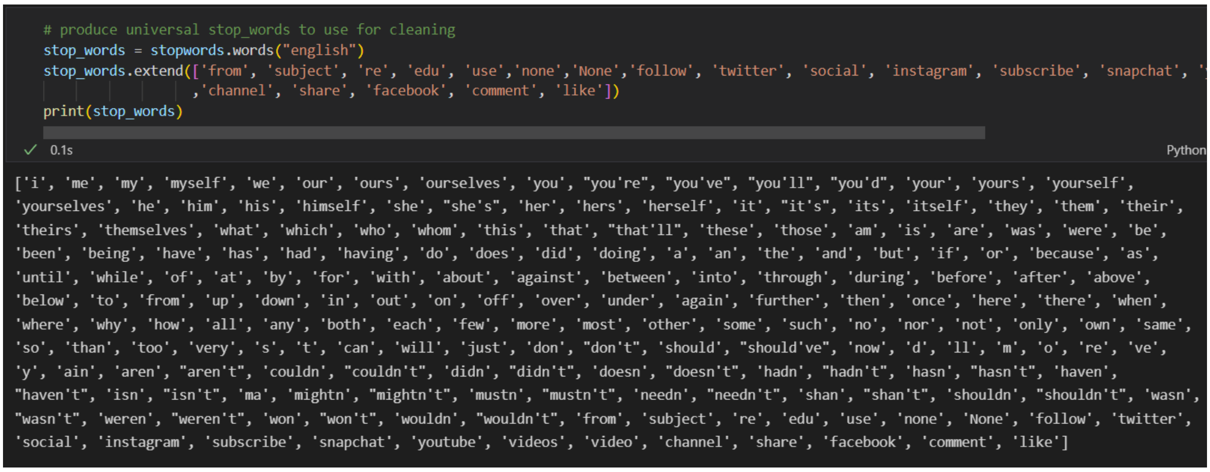
```
#remove links from the description
PB_videos.description= PB_videos.description.str.replace('http\S+|www.\S+',''\
                                                        ,regex= True).str.lower()
#removing Punctuation from description
PB_videos.description = PB_videos.description.str.replace(r'[^a-zA-Z0-9]+', ' ')
PB_videos.description= PB_videos.description.str.replace(r'[0-9]+', '')
```

```
#remove links from the title
PB_videos.title= PB_videos.title.str.replace('http\S+|www.\S+',''\
                                                        ,regex= True).str.lower()
#removing Punctuation from title
PB_videos.title= PB_videos.title.str.replace(r'[^a-zA-Z0-9]+', ' ')
PB_videos.title= PB_videos.title.str.replace(r'[0-9]+', '')
```

```
#remove links from the description
PB_videos.tags= PB_videos.tags.str.replace('http\S+|www.\S+',''\
                                                        ,regex= True).str.lower()
#removing Punctuation from description
PB_videos.tags= PB_videos.tags.str.replace(r'[^a-zA-Z0-9]+', ' ')
PB_videos.tags= PB_videos.tags.str.replace(r'[0-9]+', '')
```

Figure 4.17: Link and Punctuation Removal for Text Column

Secondly, from the NLTK library, the 'stopwords' library is used to define the function for removing stop words from text later. The "stopwords' library is extended with extra words as shown in *Figure 4.18 below.*



```
# produce universal stop_words to use for cleaning
stop_words = stopwords.words("english")
stop_words.extend(['from', 'subject', 're', 'edu', 'use','none','None','follow', 'twitter', 'social', 'instagram', 'subscribe', 'snapchat', '
                  ,'channel', 'share', 'facebook', 'comment', 'like'])
print(stop_words)
```

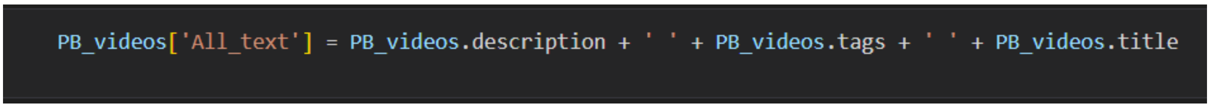
✓ 0.1s Python

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself',
'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their',
'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be',
'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as',
'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above',
'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when',
'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same',
'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've',
'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven',
"haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn',
"wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't", 'from', 'subject', 're', 'edu', 'use', 'none', 'None', 'follow', 'twitter',
'social', 'instagram', 'subscribe', 'snapchat', 'youtube', 'videos', 'video', 'channel', 'share', 'facebook', 'comment', 'like']
```

Figure 4.18: Initialize Stop Words Library with Extended Word

Based on *Figure 4.18* above, extended words are the most frequent words in the text columns that provide less to no value when building a topic model later.

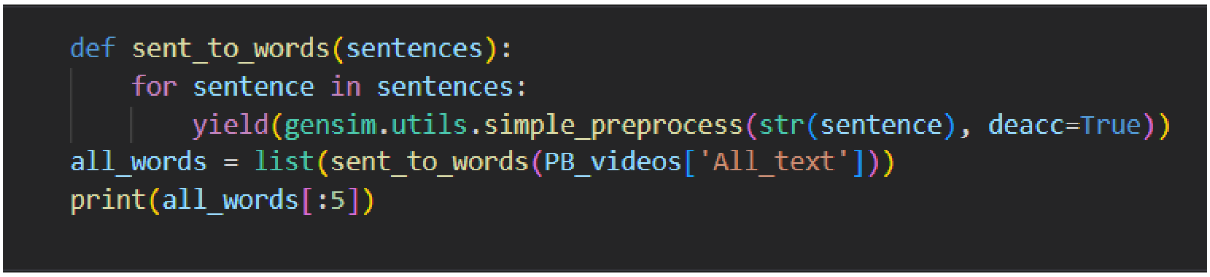
Next, text columns ‘title’, ‘description’, and ‘tags’ are merged for data pre-processing for the LDA model later as shown in *Figure 4.19*.



```
PB_videos['All_text'] = PB_videos.description + ' ' + PB_videos.tags + ' ' + PB_videos.title
```

Figure 4.19: Merging 3 text columns

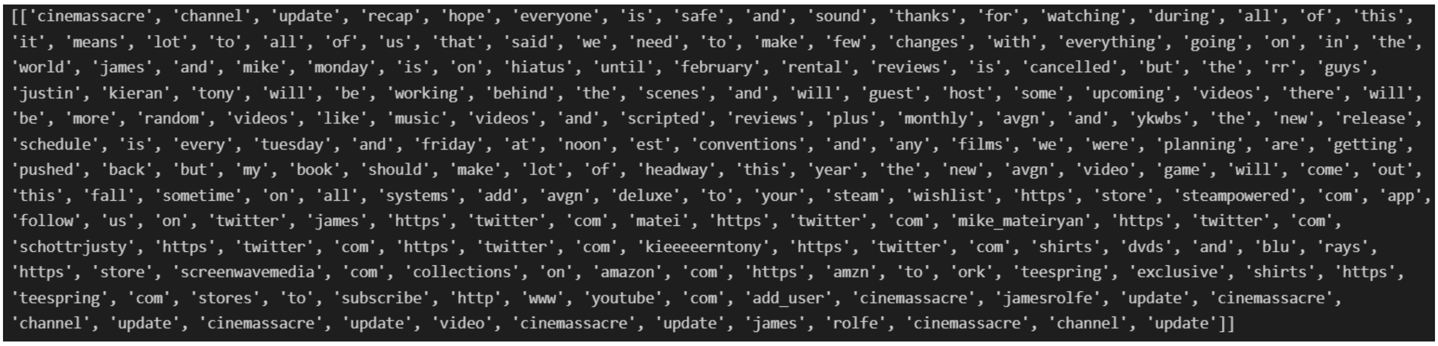
The merged column ‘All_text’ is then converted from sentence to words using the function in *Figure 4.20* below.



```
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))
all_words = list(sent_to_words(PB_videos['All_text']))
print(all_words[:5])
```

Figure 4.20: Function Sentence to Words

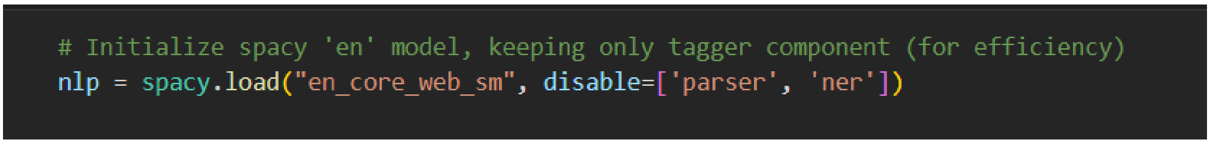
One of the results converted sentence is shown in *Figure 4.21* below.



```
[['cinemassacre', 'channel', 'update', 'recap', 'hope', 'everyone', 'is', 'safe', 'and', 'sound', 'thanks', 'for', 'watching', 'during', 'all', 'of', 'this',
'it', 'means', 'lot', 'to', 'all', 'of', 'us', 'that', 'said', 'we', 'need', 'to', 'make', 'few', 'changes', 'with', 'everything', 'going', 'on', 'in', 'the',
'world', 'james', 'and', 'mike', 'monday', 'is', 'on', 'hiatus', 'until', 'february', 'rental', 'reviews', 'is', 'cancelled', 'but', 'the', 'rr', 'guys',
'justin', 'kieran', 'tony', 'will', 'be', 'working', 'behind', 'the', 'scenes', 'and', 'will', 'guest', 'host', 'some', 'upcoming', 'videos', 'there', 'will',
'be', 'more', 'random', 'videos', 'like', 'music', 'videos', 'and', 'scripted', 'reviews', 'plus', 'monthly', 'avgn', 'and', 'ykwbs', 'the', 'new', 'release',
'schedule', 'is', 'every', 'tuesday', 'and', 'friday', 'at', 'noon', 'est', 'conventions', 'and', 'any', 'films', 'we', 'were', 'planning', 'are', 'getting',
'pushed', 'back', 'but', 'my', 'book', 'should', 'make', 'lot', 'of', 'headway', 'this', 'year', 'the', 'new', 'avgn', 'video', 'game', 'will', 'come', 'out',
'this', 'fall', 'sometime', 'on', 'all', 'systems', 'add', 'avgn', 'deluxe', 'to', 'your', 'steam', 'wishlist', 'https', 'store', 'steampowered', 'com', 'app',
'follow', 'us', 'on', 'twitter', 'james', 'https', 'twitter', 'com', 'matei', 'https', 'twitter', 'com', 'mike_mateiryan', 'https', 'twitter', 'com',
'schottrjusty', 'https', 'twitter', 'com', 'https', 'twitter', 'com', 'kieeeeerntony', 'https', 'twitter', 'com', 'shirts', 'dvds', 'and', 'blu', 'rays',
'https', 'store', 'screenwavemedia', 'com', 'collections', 'on', 'amazon', 'com', 'https', 'amzn', 'to', 'ork', 'teespring', 'exclusive', 'shirts', 'https',
'teespring', 'com', 'stores', 'to', 'subscribe', 'http', 'www', 'youtube', 'com', 'add_user', 'cinemassacre', 'jamesrolfe', 'update', 'cinemassacre',
'channel', 'update', 'cinemassacre', 'update', 'video', 'cinemassacre', 'update', 'james', 'rolfe', 'cinemassacre', 'channel', 'update']]
```

Figure 4.21Example of Converted Sentence to Words

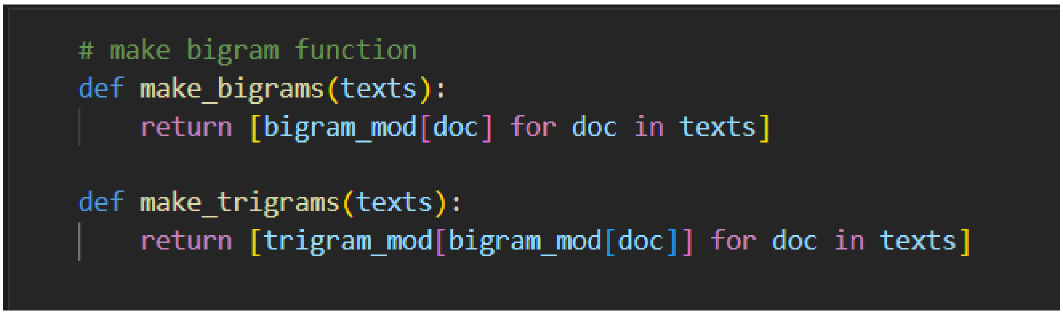
Then, the bigram and trigram words from the converted text are created. Bigrams are pairs of words that commonly appear together in a text. Meanwhile, three words are known as trigrams. Before making bigrams and trigrams, the "English" natural language process is loaded into the file using SpaCy to reduce the noise of word data and increase efficiency by keeping only the tagger component of words. The initialization function is shown in *Figure 4.22* below.



```
# Initialize spacy 'en' model, keeping only tagger component (for efficiency)
nlp = spacy.load("en_core_web_sm", disable=['parser', 'ner'])
```

Figure 4.22: Load NLP for the English Language

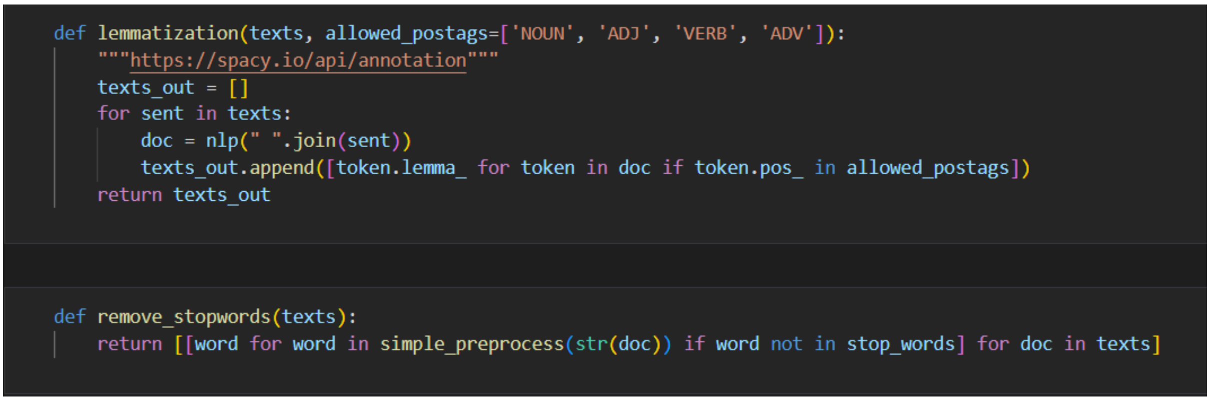
The English NLP model for bigrams and trigrams function is initialized as shown in *Figure 4.23* below.



```python
# make bigram function
def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]

def make_trigrams(texts):
    return [trigram_mod[bigram_mod[doc]] for doc in texts]
```

Figure 4.23: Make Bigrams and Trigrams Function

Next, the lemmatization and removing stop words function is initialized as shown in *Figure 4.24* below.

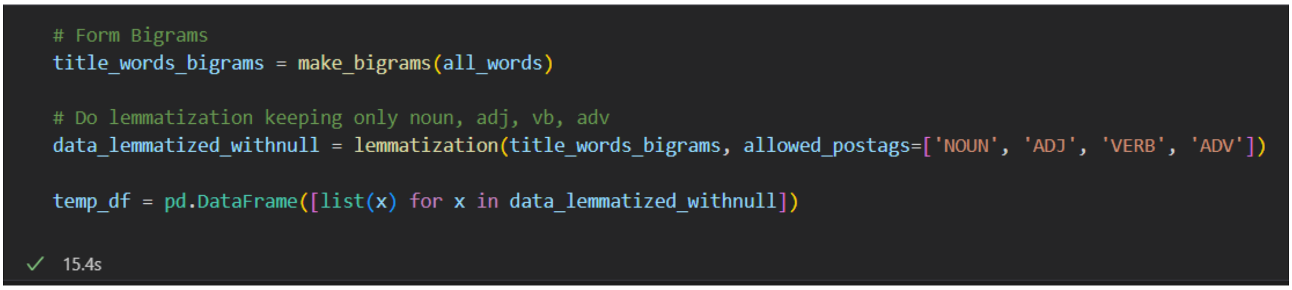


```python
def lemmatization(texts, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):
    """https://spacy.io/api/annotation"""
    texts_out = []
    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])
    return texts_out
```

```python
def remove_stopwords(texts):
    return [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc in texts]
```

Figure 4.24: Initialization of Lemmatization and Removing Stop Words

The initialized functions are then used to do lemmatization while keeping only nouns, adjectives, verbs, and adverbs. The process of lemmatization is shown using the line of code below.

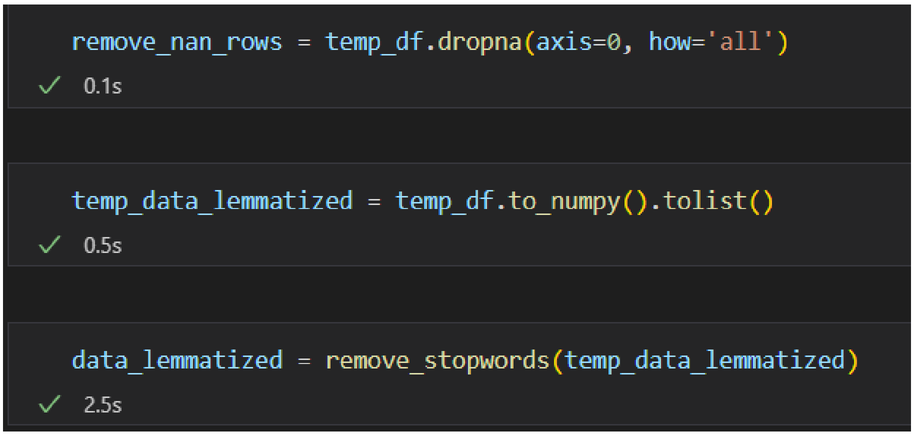


```python
# Form Bigrams
title_words_bigrams = make_bigrams(all_words)

# Do lemmatization keeping only noun, adj, vb, adv
data_lemmatized_withnull = lemmatization(title_words_bigrams, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV'])

temp_df = pd.DataFrame([list(x) for x in data_lemmatized_withnull])
```

✓ 15.4s

Figure 4.25: Lemmatization process

Some of the videos are not in English, therefore some rows have a ‘NaN’ value. The ‘NaN’ rows might affect the corpus model and later in the LDA model so it is necessary to remove the rows. The ‘NaN’ rows are removed using the dropping function how = ‘all’ shown in *Figure 4.26* below.



```
remove_nan_rows = temp_df.dropna(axis=0, how='all')
✓ 0.1s

temp_data_lemmatized = temp_df.to_numpy().tolist()
✓ 0.5s

data_lemmatized = remove_stopwords(temp_data_lemmatized)
✓ 2.5s
```

Figure 4.26: Remove NaN rows

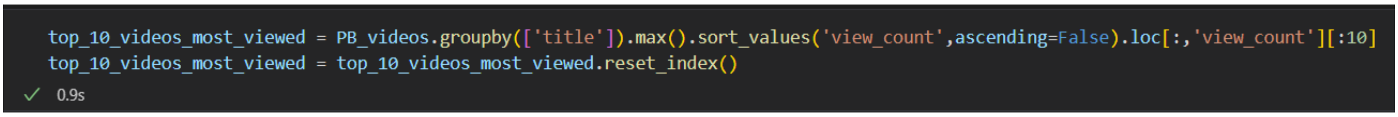
Based on *Figure 4.26* above, the stop words are removed after the removal of NaN rows.

4.2.2 Exploratory Data Analysis (EDA)

In Exploratory Data Analysis, the process is used to summarise and highlight the important aspects of the data to be investigated. It is also to detect the feature that is plausible to feed into a machine learning algorithm. The analysis involved Univariate Analysis, Bivariate Analysis, Multivariate Analysis, and Time Series Analysis.

4.2.2.1 Univariate Analysis

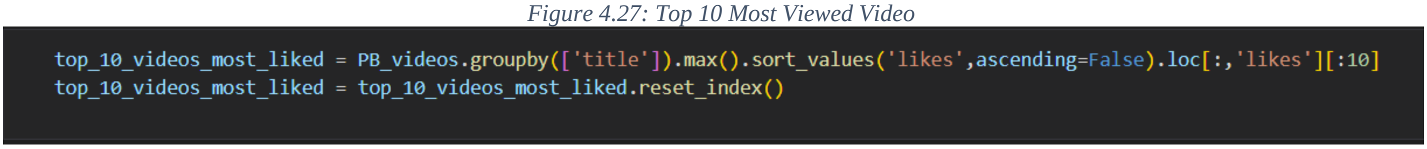
Firstly, the top 10 most viewed videos in the category “People and Blog” are observed using the line of code in *Figure 4.27* below.



```
top_10_videos_most_viewed = PB_videos.groupby(['title']).max().sort_values('view_count',ascending=False).loc[:,'view_count'][:10]
top_10_videos_most_viewed = top_10_videos_most_viewed.reset_index()
```

✓ 0.9s

Figure 4.27: Top 10 Most Viewed Video

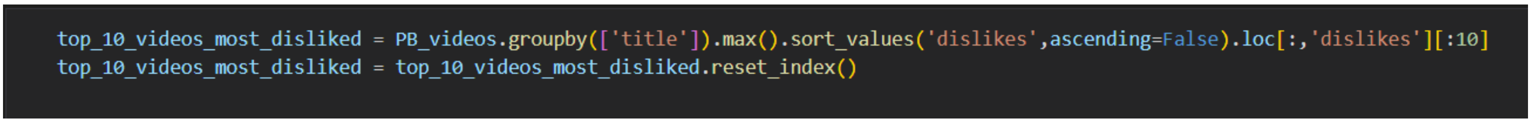


```
top_10_videos_most_liked = PB_videos.groupby(['title']).max().sort_values('likes',ascending=False).loc[:,'likes'][:10]
top_10_videos_most_liked = top_10_videos_most_liked.reset_index()
```

Secondly, the top 10 most liked videos in the category “People and Blog” are observed using the line of code in *Figure 4.28* below

Figure 4.28: Top 10 Most Liked Videos

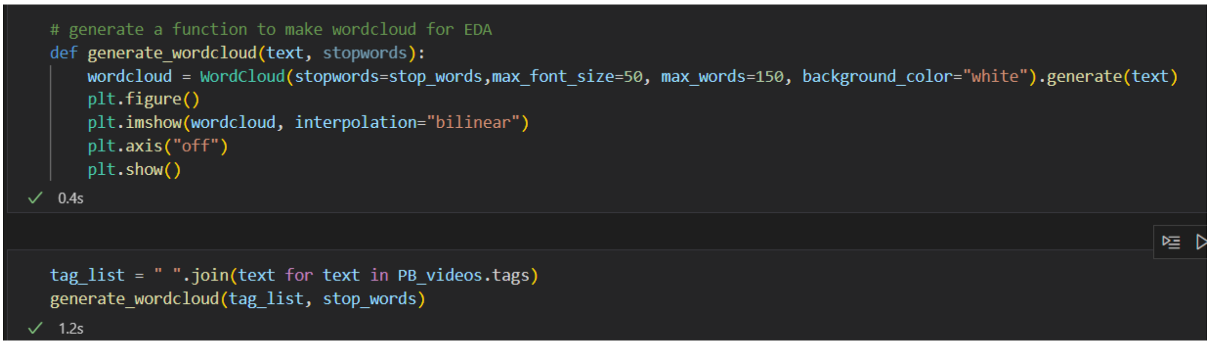
Next, the top 10 most disliked videos in the category “People and Blog” are observed using the line of code in *Figure 4.29* below



```
top_10_videos_most_disliked = PB_videos.groupby(['title']).max().sort_values('dislikes',ascending=False).loc[:,'dislikes'][:10]
top_10_videos_most_disliked = top_10_videos_most_disliked.reset_index()
```

Figure 4.29: Top 10 Most Disliked Videos

For text columns, the univariate analysis is conducted by observing text column ‘tags’ as shown in *Figure 4.30* below.



```
# generate a function to make wordcloud for EDA
def generate_wordcloud(text, stopwords):
    wordcloud = WordCloud(stopwords=stop_words,max_font_size=50, max_words=150, background_color="white").generate(text)
    plt.figure()
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.show()
✓ 0.4s
```

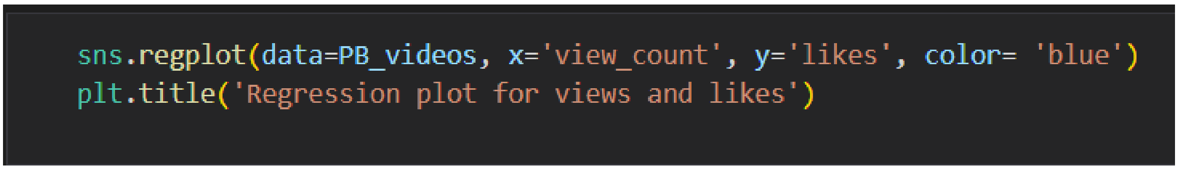
```
tag_list = " ".join(text for text in PB_videos.tags)
generate_wordcloud(tag_list, stop_words)
✓ 1.2s
```

Figure 4.30:Generate WordCloud for tags

Based on *Figure 4.30* above, the function to generate WordCloud is initialized for future use in other Exploratory Data Analysis techniques.

4.2.2.2 Bivariate Analysis

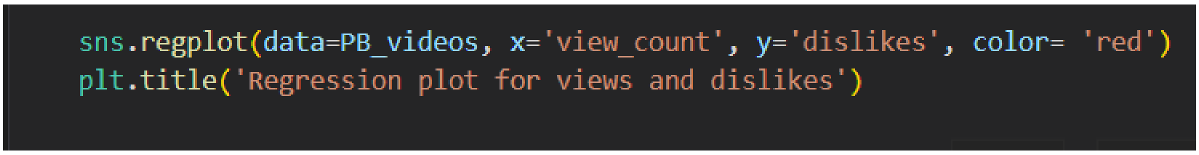
Bivariate analysis is conducted to observe bicorrelation between 2 features. The bicorrelation is plotted using a regression plot. The findings of the different regression plots are discussed in Chapter 5 later.



```
sns.regplot(data=PB_videos, x='view_count', y='likes', color= 'blue')
plt.title('Regression plot for views and likes')
```

Figure 4.31: Regression Plotting for Views and Likes

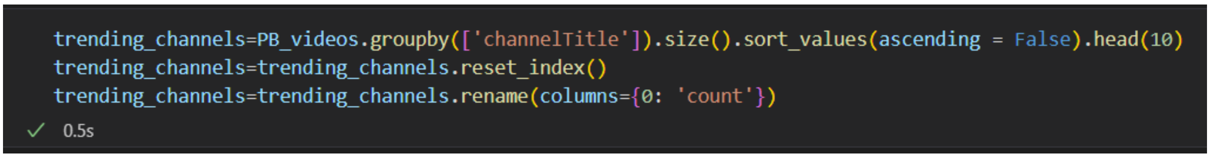
Figure 4.31 above shows the plotting process for regression plot Views vs Likes



```
sns.regplot(data=PB_videos, x='view_count', y='dislikes', color= 'red')
plt.title('Regression plot for views and dislikes')
```

Figure 4.32: Regression Plotting for Views and Dislikes

Figure 4.32 above shows the plotting process for regression plot Views vs Dislikes. Next, for the observation of channels, the Top 10 channels with the most trending videos on trending are observed as shown in *Figure 4.33* below.



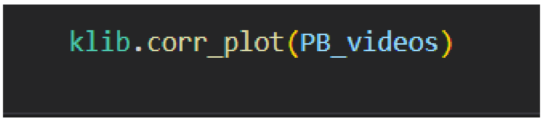
```
trending_channels=PB_videos.groupby(['channelTitle']).size().sort_values(ascending = False).head(10)
trending_channels=trending_channels.reset_index()
trending_channels=trending_channels.rename(columns={0: 'count'})
```

✓ 0.5s

Figure 4.33: Top 10 channels with most videos on trending

4.2.2.3 Multivariate Analysis

Multivariate Analysis is a statistical analysis of correlations between several features in the dataset. In this dataset, the correlation of numerical features is plotted using a heatmap from the *Klib* library.



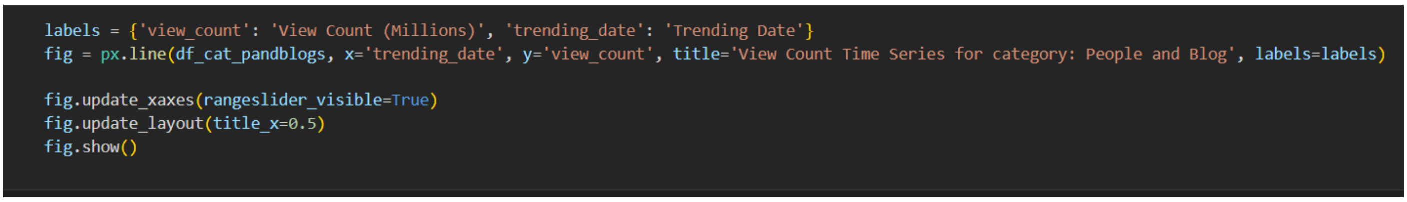
```
klib.corr_plot(PB_videos)
```

Figure 4.34: Correlation Plotting for numerical features in the dataset

Based on Figure 4.34 above, the findings of the heatmap are discussed in detail later in Chapter 5.

4.2.2.4 Time Series Analysis

Time Series Analysis is used to discover hidden insights based on time intervals. In this dataset, the general topic is observed using the view counts to search for the highest point of engagement across the timestamp. The time series plotting is shown in *Figure 4.35 below*.



```
labels = {'view_count': 'View Count (Millions)', 'trending_date': 'Trending Date'}
fig = px.line(df_cat_pandblogs, x='trending_date', y='view_count', title='View Count Time Series for category: People and Blog', labels=labels)

fig.update_xaxes(rangeslider_visible=True)
fig.update_layout(title_x=0.5)
fig.show()
```

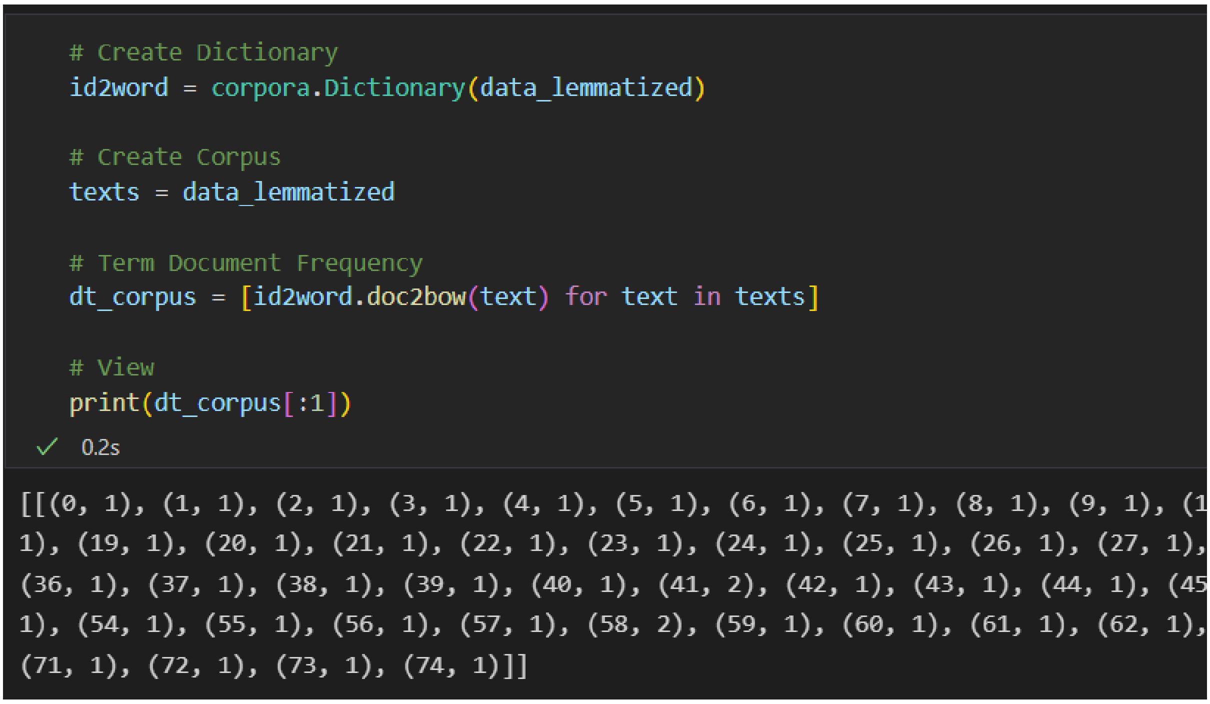
Figure 4.35: Time Series for view counts

4.4 Phase 2: Topic Modeling

In the “People and Blog” category, although it is categorized as “People and Blog” there are various types of videos that can be posted in this category. From lifestyle to storytelling, the variance of a topic is still in the grey area. Topic Modeling is an unsupervised machine learning algorithm that works by using the distribution of words in a collection of documents to find latent patterns of word occurrence. In this project, topic Modeling is used to discover what topic of video exactly raised high engagements using the LDA machine learning model.

4.4.1 Feature Engineering

In the data pre-processing steps earlier, the text column ‘title’, ‘description’, and ‘tags’ are merged into a new text column. This is because using only one column such as the title is insufficient to predict what topic is the video about. After the lemmatization process, some of the rows were cleaned up until they became ‘NaN’ rows. Some of them also have less than 3 words after the cleaning process. Therefore, to improve the dictionary corpus dimensionality, the 3 columns are transformed and merged into one single column.



```
# Create Dictionary
id2word = corpora.Dictionary(data_lemmatized)

# Create Corpus
texts = data_lemmatized

# Term Document Frequency
dt_corpus = [id2word.doc2bow(text) for text in texts]

# View
print(dt_corpus[:1])
```

✓ 0.2s

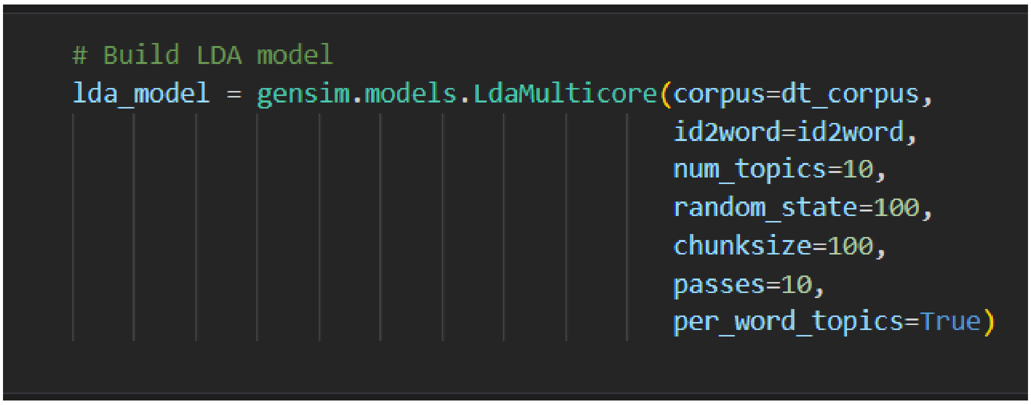
```
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (1
1), (19, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1),
(36, 1), (37, 1), (38, 1), (39, 1), (40, 1), (41, 2), (42, 1), (43, 1), (44, 1), (45
1), (54, 1), (55, 1), (56, 1), (57, 1), (58, 2), (59, 1), (60, 1), (61, 1), (62, 1),
(71, 1), (72, 1), (73, 1), (74, 1)]]
```

Figure 4.36: Create a dictionary, corpus of texts, and TF-IDF frequency

Figure 4.36 above shows that lemmatized data is then used to create a dictionary, corpus of texts, and TF-IDF frequency.

4.4.2 LDA Modeling

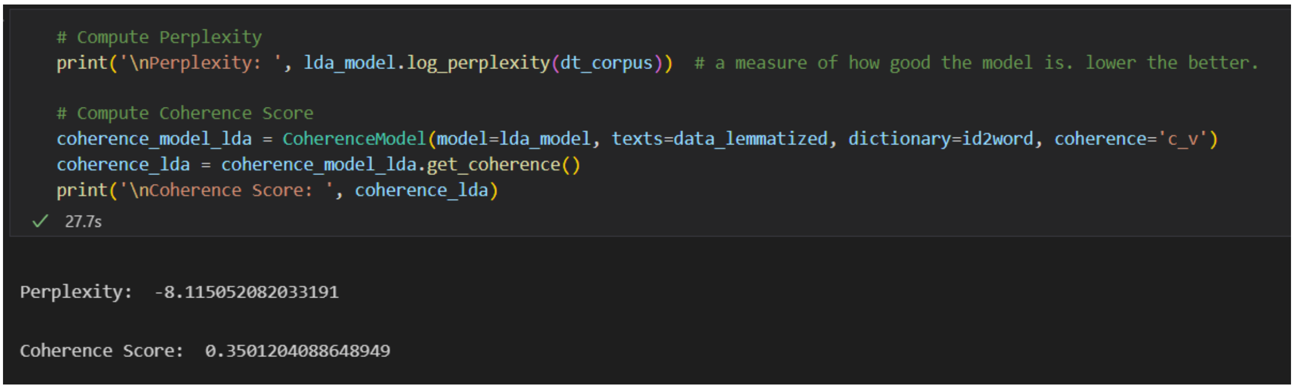
LDA Model is built using a randomly assigned number of topics, *k*. In this project, LDA Model is initialized with the number of topics, *k* = 10. The snippet code is shown in *Figure 4.37* below.



```
# Build LDA model
lda_model = gensim.models.LdaMulticore(corpus=dt_corpus,
                                       id2word=id2word,
                                       num_topics=10,
                                       random_state=100,
                                       chunksize=100,
                                       passes=10,
                                       per_word_topics=True)
```

Figure 4.37: Build base LDA Model

Then, to partially evaluate the LDA model perplexity and coherence scores are computed as shown in *Figure 4.38* below.



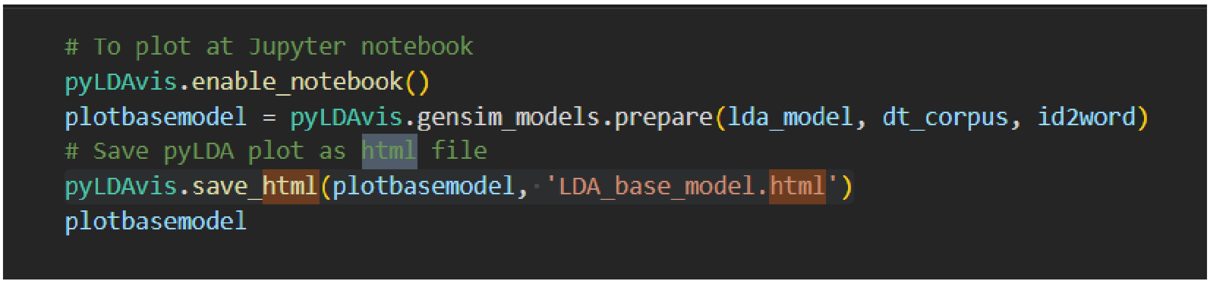
```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(dt_corpus))  # a measure of how good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

✓ 27.7s

```
Perplexity:  -8.115052082033191

Coherence Score:  0.3501204088648949
```

Figure 4.38: Compute Perplexity and Coherence Score

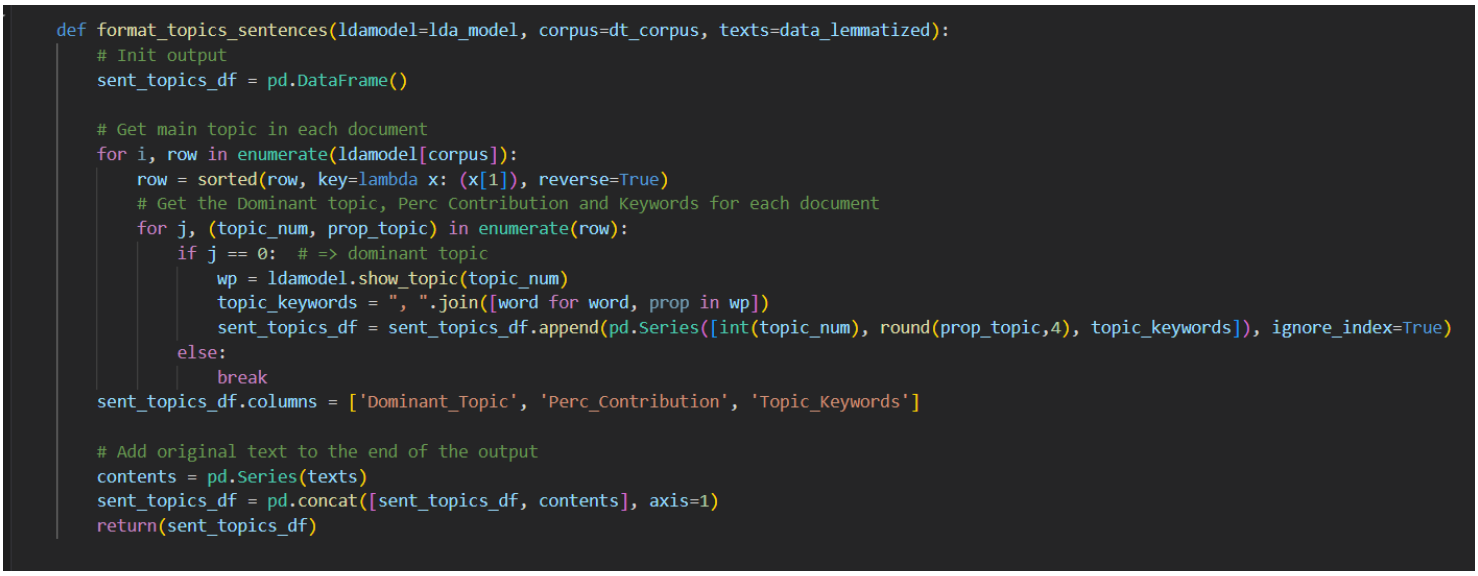
The base LDA model is then plotted for better visualization of topic distributions. The snippet code to plot the base LDA model is shown in *Figure 4.39* below.



```python
# To plot at Jupyter notebook
pyLDAvis.enable_notebook()
plotbasemodel = pyLDAvis.gensim_models.prepare(lda_model, dt_corpus, id2word)
# Save pyLDA plot as html file
pyLDAvis.save_html(plotbasemodel, 'LDA_base_model.html')
plotbasemodel
```

Figure 4.39: Plot base LDA Model

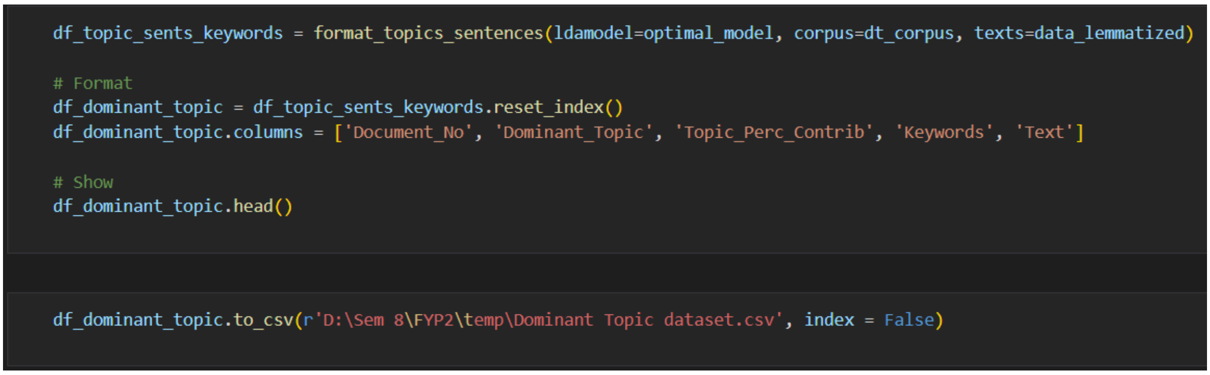
Then, the weighted number of topics obtained is pushed into a new Data Frame and later used to predict the main topic of each document based on the percentage of contribution of the document. *Figure 4.40* below shows the snippet code to



```python
def format_topics_sentences(ldamodel=lda_model, corpus=dt_corpus, texts=data_lemmatized):
    # Init output
    sent_topics_df = pd.DataFrame()

    # Get main topic in each document
    for i, row in enumerate(ldamodel[corpus]):
        row = sorted(row, key=lambda x: (x[1]), reverse=True)
        # Get the Dominant topic, Perc Contribution and Keywords for each document
        for j, (topic_num, prop_topic) in enumerate(row):
            if j == 0:  # => dominant topic
                wp = ldamodel.show_topic(topic_num)
                topic_keywords = ", ".join([word for word, prop in wp])
                sent_topics_df = sent_topics_df.append(pd.Series([int(topic_num), round(prop_topic,4), topic_keywords]), ignore_index=True)
            else:
                break
    sent_topics_df.columns = ['Dominant_Topic', 'Perc_Contribution', 'Topic_Keywords']

    # Add original text to the end of the output
    contents = pd.Series(texts)
    sent_topics_df = pd.concat([sent_topics_df, contents], axis=1)
    return(sent_topics_df)
```

Figure 4.40 Function to Format Topic Sentence

Figure 4.41 below shows the snippet code of the dataset of the dominant topic fetched into CSV files.



```
df_topic_sents_keywords = format_topics_sentences(ldamodel=optimal_model, corpus=dt_corpus, texts=data_lemmatized)

# Format
df_dominant_topic = df_topic_sents_keywords.reset_index()
df_dominant_topic.columns = ['Document_No', 'Dominant_Topic', 'Topic_Perc_Contrib', 'Keywords', 'Text']

# Show
df_dominant_topic.head()
```

```
df_dominant_topic.to_csv(r'D:\Sem 8\FYP2\temp\Dominant Topic dataset.csv', index = False)
```

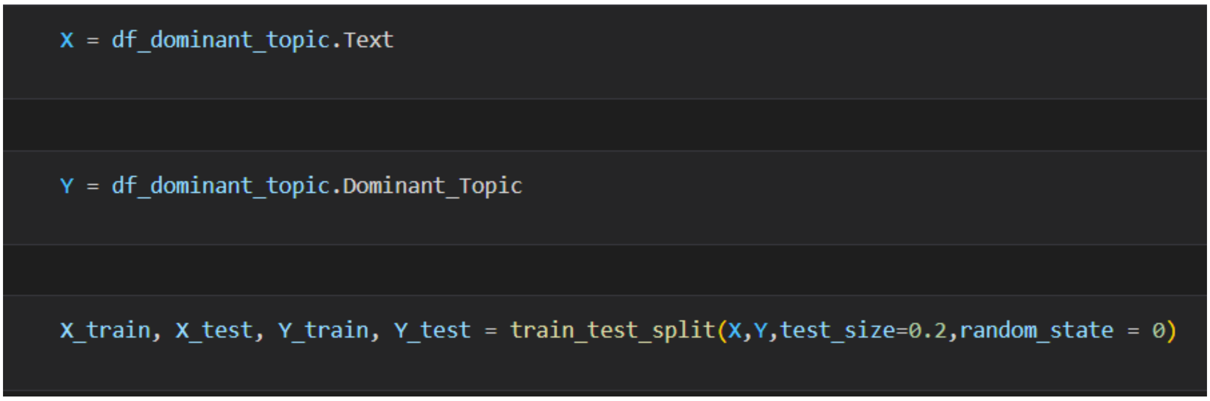
Figure 4.41Dominant Topic dataset for each document

4.5 Phase 3: Model Evaluation

In this phase, based on the base LDA Model’s topic computed in Phase 2, all rows in column ‘All_text’ are assigned to their predicted dominant topic. Then, the data is divided into train and test sets. The split data is then fed into supervised learning, Random Forest Classification.

4.5.1 Data Division

Using the dominant topic discovered by the base LDA model, the text is assigned according to the weightage of the word that occurs in the documents. Then, the data ‘Text’ and ‘Dominant_Topic’ is split into training and testing data. The ratio of data division is 80:20 for training sets and testing sets respectively. The code snippet of the data split process is shown in *Figure 4.42* below.



```
X = df_dominant_topic.Text

Y = df_dominant_topic.Dominant_Topic

X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.2,random_state = 0)
```

Figure 4.42: Split Training and Testing Set

4.5.2 Random Forest Classification

To measure the performance of the model, Random Forest Classification is implemented in 2 models used:

- Bag-Of-Word (BoW) model
- TF-IDF model

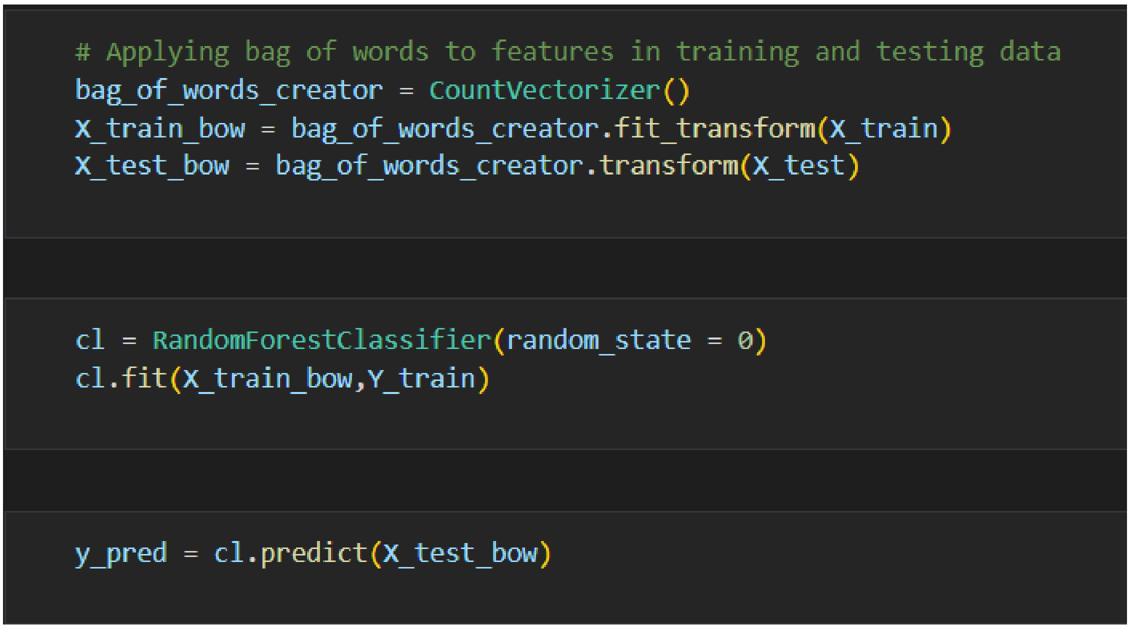
BoW model is used to measure how well the document classification is based on the frequency occurrence of each word when it is used as a feature in the model classifier.

While TF-IDF model is used to measure how well the document classification is based on the most relevant word in the document.

The performance of both supervised learnings is using a confusion matrix.

4.5.2.1 Bag-Of-Word model

BoW model is built based on snippet code in *Figure 4.43* below.



```
# Applying bag of words to features in training and testing data
bag_of_words_creator = CountVectorizer()
X_train_bow = bag_of_words_creator.fit_transform(X_train)
X_test_bow = bag_of_words_creator.transform(X_test)
```

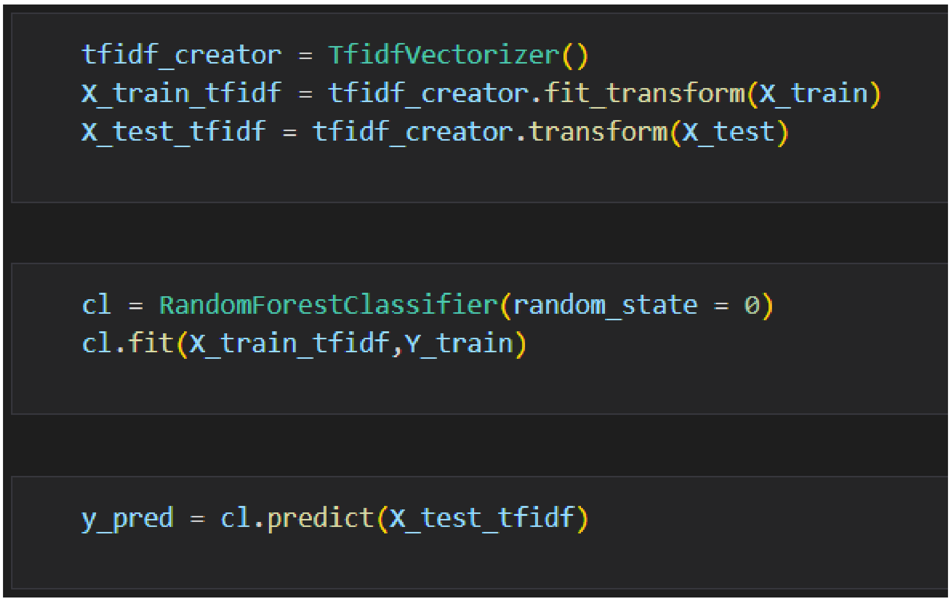
```
cl = RandomForestClassifier(random_state = 0)
cl.fit(X_train_bow,Y_train)
```

```
y_pred = cl.predict(X_test_bow)
```

Figure 4.43: Building BoW Model

4.5.2.2 TF-IDF model

TF-IDF model is built based on snippet code in *Figure 4.44* below.



```
tfidf_creator = TfidfVectorizer()
X_train_tfidf = tfidf_creator.fit_transform(X_train)
X_test_tfidf = tfidf_creator.transform(X_test)
```

```
cl = RandomForestClassifier(random_state = 0)
cl.fit(X_train_tfidf,Y_train)
```

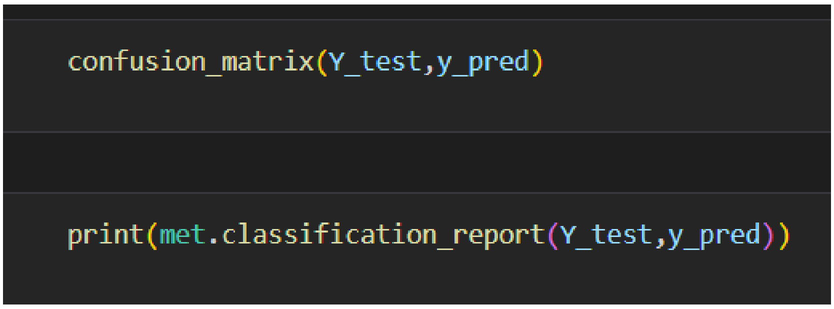
```
y_pred = cl.predict(X_test_tfidf)
```

Figure 4.44: Building TF-IDF Model

4.5.2.3 Classification Evaluation: Confusion Matrix

The confusion matrix of both the BoW model and TF-IDF model is computed and observed.

The snippet code of the confusion matrix compute is shown in *Figure 4.45* below.



```
confusion_matrix(Y_test,y_pred)
```

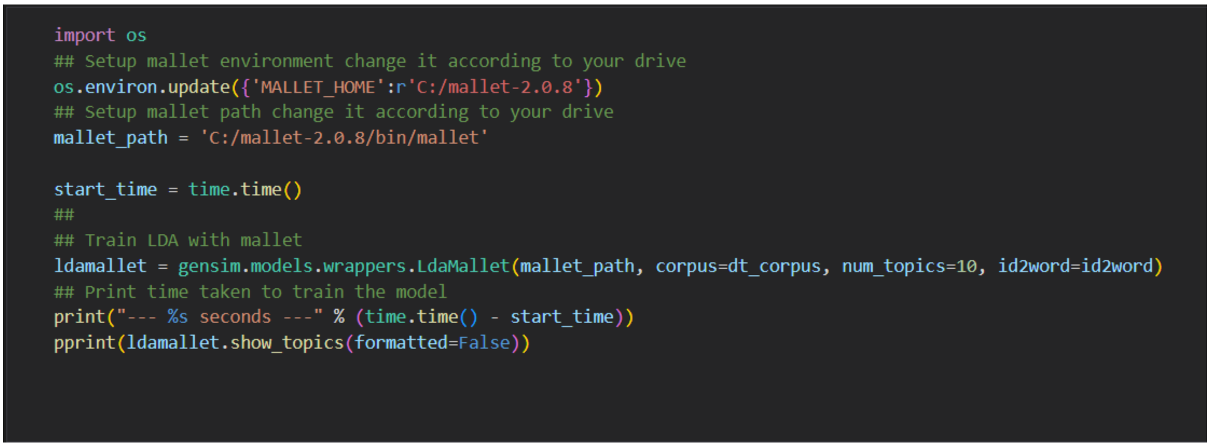
```
print(met.classification_report(Y_test,y_pred))
```

Figure 4.45: Compute Confusion matrix

4.6 Phase 4: Model Tuning

4.6.1 Building LDA Mallet Model

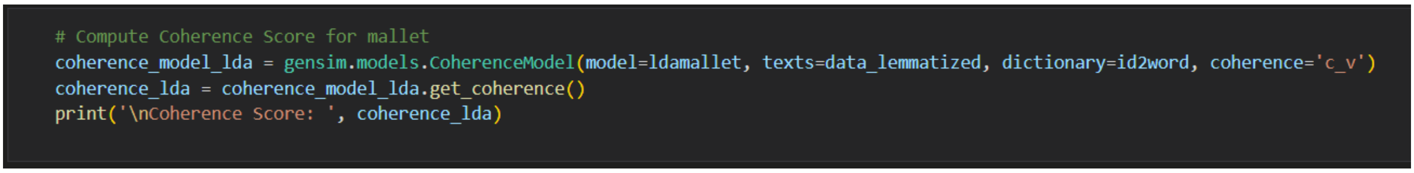
In model tuning, firstly LDA Mallet model is built and observed to whether the coherence score is improved or not. The snippet code for the LDA Mallet model is shown in *Figure 4.46* below.



```python
import os
## Setup mallet environment change it according to your drive
os.environ.update({'MALLET_HOME':r'C:/mallet-2.0.8'})
## Setup mallet path change it according to your drive
mallet_path = 'C:/mallet-2.0.8/bin/mallet'

start_time = time.time()
##
## Train LDA with mallet
ldamallet = gensim.models.wrappers.LdaMallet(mallet_path, corpus=dt_corpus, num_topics=10, id2word=id2word)
## Print time taken to train the model
print("--- %s seconds ---" % (time.time() - start_time))
pprint(ldamallet.show_topics(formatted=False))
```

Figure 4.46: Initiate LDA Mallet Model

Based on *Figure 4.46* above, the number of topics, *k* is assigned constantly the same as the base LDA model. Then, after LDA Mallet Model is trained the coherence score is computed. *Figure 4.47* below shows the snippet code to compute the coherence score of the LDA Mallet Model.



```python
# Compute Coherence Score for mallet
coherence_model_lda = gensim.models.CoherenceModel(model=ldamallet, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

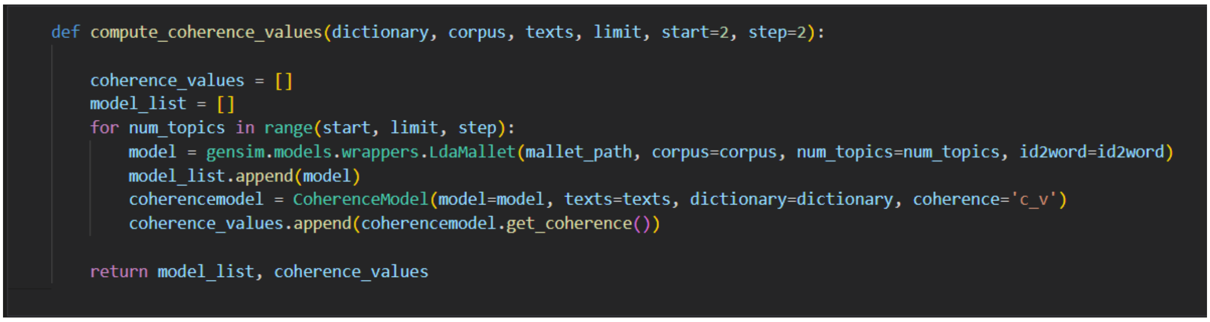
Figure 4.47: Compute Coherence Score for LDA Mallet Model

4.6.2 Finding the Optimal Number of Topics, *k*

The LDA model is tuned by adjusting the hyperparameters of the model. LDA model hyperparameters are:

- alpha, α represents the density of document-topic
- Beta, β represents the density of topic-word
- Number of topics, *k*

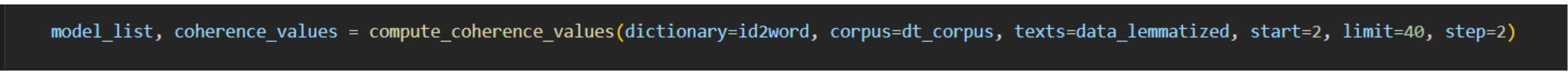
For this project, the tuning process only involves adjusting the number of topics. Multiple models are built with different values of *k* assigned to each model. The results are observed by computing the coherence score for each model. The initialization of the function to compute multiple LDA Mallet is shown in *Figure 4.48* below.



```
def compute_coherence_values(dictionary, corpus, texts, limit, start=2, step=2):

    coherence_values = []
    model_list = []
    for num_topics in range(start, limit, step):
        model = gensim.models.wrappers.LdaMallet(mallet_path, corpus=corpus, num_topics=num_topics, id2word=id2word)
        model_list.append(model)
        coherencemodel = CoherenceModel(model=model, texts=texts, dictionary=dictionary, coherence='c_v')
        coherence_values.append(coherencemodel.get_coherence())

    return model_list, coherence_values
```

Figure 4.48: Function to Compute Coherence for Multiple LDA Mallet Model

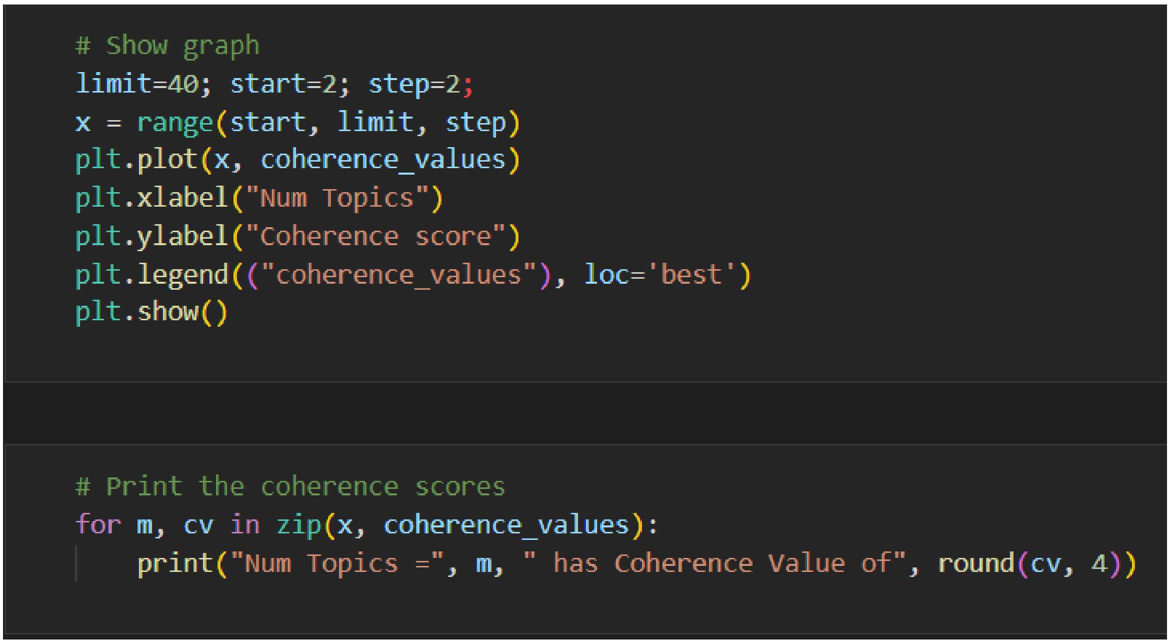
The process of training and computing coherence score for multiple LDA Mallet models is shown in *Figure 4.49* below.



```
model_list, coherence_values = compute_coherence_values(dictionary=id2word, corpus=dt_corpus, texts=data_lemmatized, start=2, limit=40, step=2)
```

Figure 4.49: Build and compute coherence score of multiple LDA Mallet model

The discussion of choosing the optimal number of topics, k is discussed in detail in Chapter 5. The graph of the model list is plotted for Coherence score 'c_v' vs the number of topics, k for k is in the range of $2 \leq k \leq 40$. Then, coherence score of models is listed as shown in *Figure 4.50* below.



```
# Show graph
limit=40; start=2; step=2;
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()
```

```
# Print the coherence scores
for m, cv in zip(x, coherence_values):
    print("Num Topics =", m, " has Coherence Value of", round(cv, 4))
```

Figure 4.50: Graph Plot for Coherence score vs Number of Topics

The model is tuned until reasonable performance is acquired in the Model Evaluation process. The reasonable performance is discussed in detail in Chapter 5.

4.7 Conclusion

This chapter has discussed the configuration software used, along with the Python packages involved. The implementation of this project was documented using the Jupyter Notebook extension in *Microsoft VS Code*. A detailed explanation for steps involved in Phase 1, Phase 2, Phase 3, and Phase 4 is also included along with Figures containing the Snippet code used. The findings based on the EDA result analysis and the topic modeling will be discussed in the next chapter.

CHAPTER 5: ANALYSIS & RESULTS DISCUSSION

In this chapter, findings based on Exploratory Data Analysis will be discussed on how the findings are related to real-world situations, while topic modeling will be discussed on how the optimal number of topics, *k* is obtained. Lastly, the results of findings based on topic modeling will be discussed on how the findings fit the problem statement of the project.

5.1 Findings Based on EDA

In this subsection, findings are described based on 4 types of analysis:

- Univariate Analysis

- Bivariate Analysis

- Multivariate Analysis

- Time Series Analysis

5.1.1 Univariate Analysis Findings

Univariate analysis is conducted based on how videos in the dataset are distributed respectively to view counts, like counts and dislikes count, and tags.

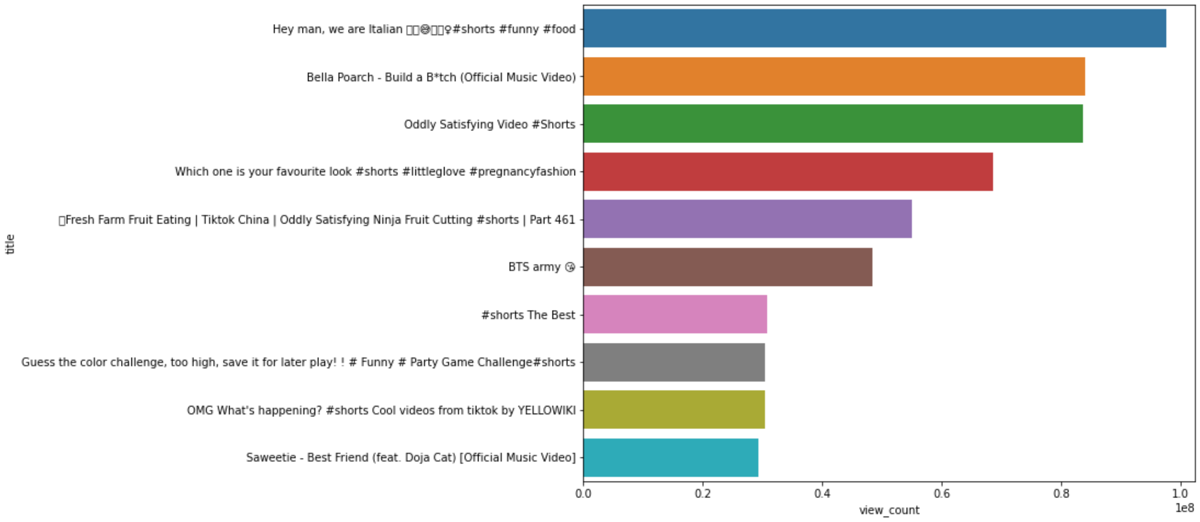


Figure 5.1: Top 10 most viewed videos in Category People and Blog

Based on *Figure 5.1* above, proves that most viewed videos from the category "People and blog" have wide topics posted. Other than that, the majority of the top viewed videos consist of hashtag #shorts in the count of 7 out of 10 most viewed videos. Roughly, #shorts are used to indicate that the duration of the video is short.

Table 1: Most Frequent Hashtags for Videos in the category People and Blog

Hashtags	Count
#shorts	846
#viral	58
#funny	47
#POV	35
#food	25
#4	32
#meme	19
#trending	16

#minecraft	16
#FYP	16
#Dpeezy2099	16
#1	16
#2	13
#ad	12
#dowehaveaproblem	11

YouTube has a new feature that enables content creators to upload their videos in a form of short videos. Based on *Table 1* above, the analysis has proved that usage of the new feature is effective to attract viewers. After removing the word ‘short’ along with other stop words, the WorkCloud for tags used in the videos is generated as shown in *Figure 5.2*.

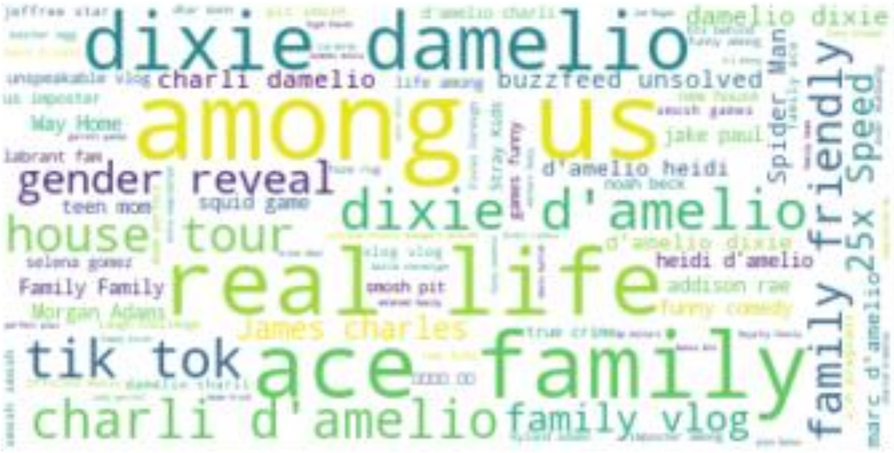


Figure 5.2: WorkCloud of Most Tags used

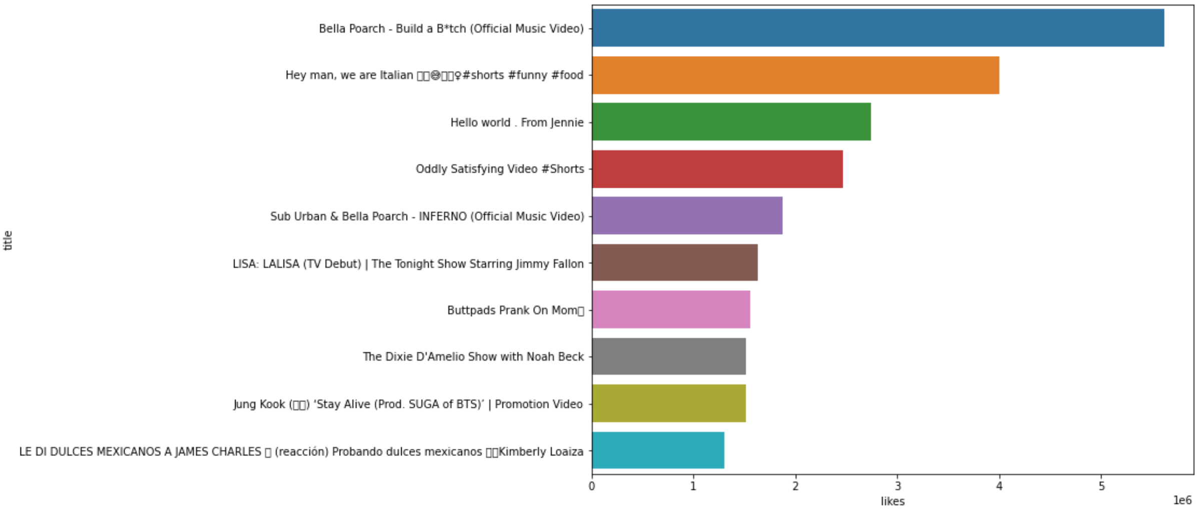


Figure 5.3: Top 10 most liked videos in Category People and Blog

Based on *Figure 5.3* above, there are slight changes in the order of the Top 10 videos. This indicates that not every trending video has positive feedback despite higher views.

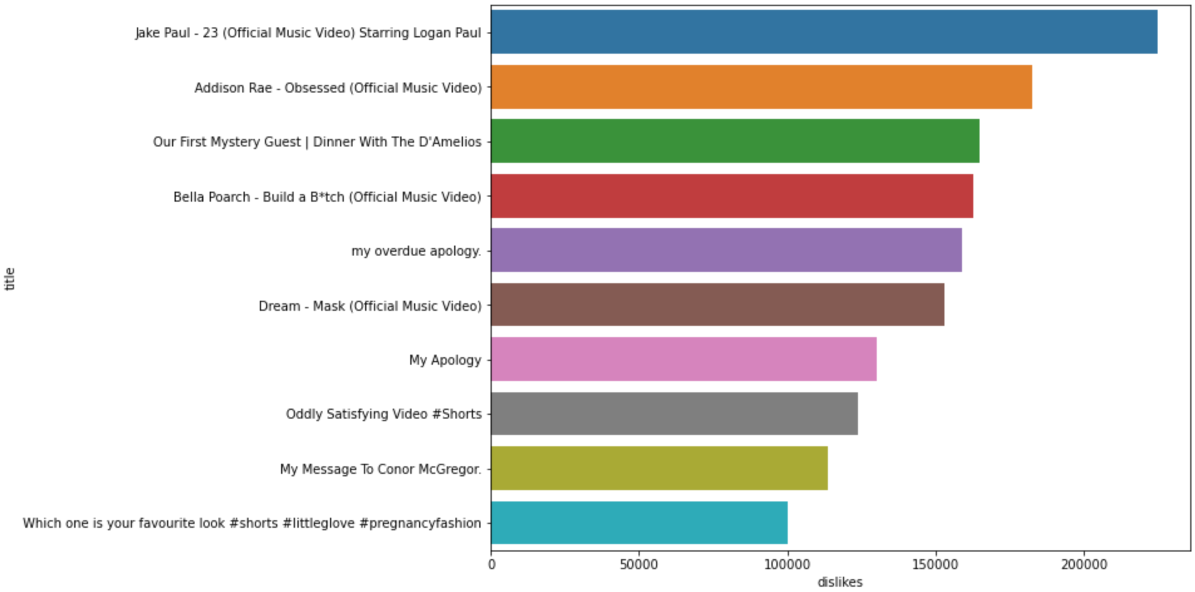


Figure 5.4: Top 10 most disliked videos in Category People and Blog

Based on *Figure 5.4* above, generally, videos with high views tend to have negative feedback too.

5.1.2 Bivariate Analysis Findings

Findings of Bivariate Analysis are conducted by bicorrelation of numeric features.

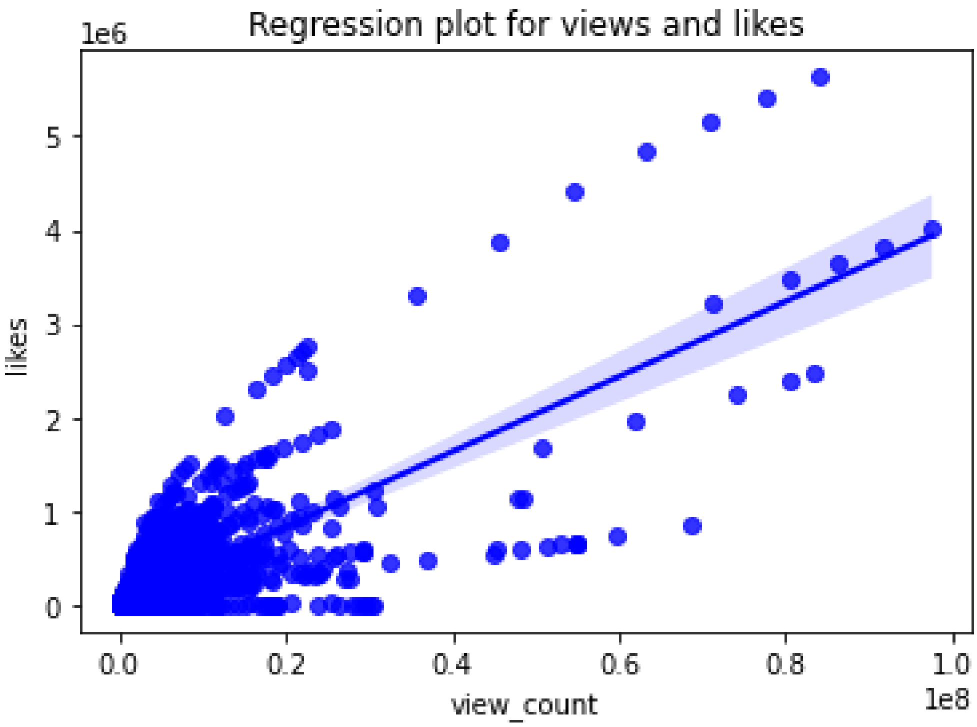


Figure 5.5: Regression plot for Views and Likes

Figure 5.5 above shows that the relationship between views and likes is linearly related. From the plot, a large percentile of videos is scattered close to the simple linear regression line while a minor percentile of videos is considered outliers. However, the residual value of the outliers is distributed linearly too. This resulting the videos having more than one pattern. The simple assumption can be made that likes do not always dependent on the view count. Some possibilities are taken from the scattered plot, where view counts of 2 videos are the same, but one is highly skewed from the line of best fit likes and another one has lowlily skewed from the line of best fit likes.

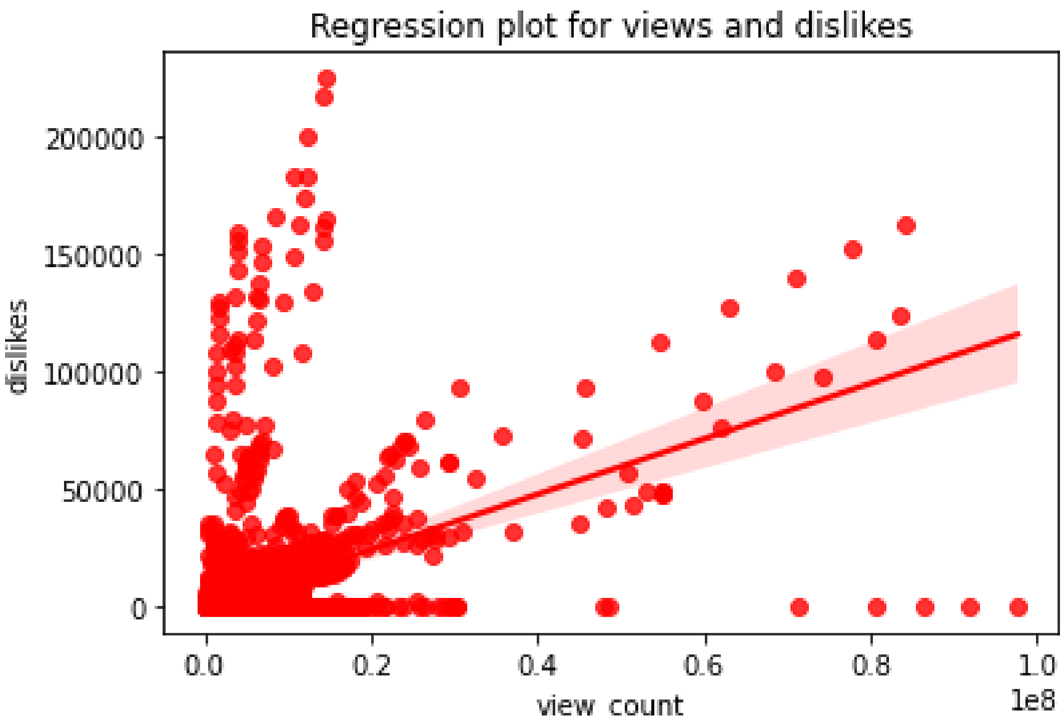


Figure 5.6: Regression Plot for views and dislikes

Figure 5.6 shows that the simple linear regression line does not fit views and likes. While a large percentile of videos is scattered close to the line, some outliers potentially show different behavioral patterns. The usual case is where the greater view counts, the greater amount of dislikes a video has. However, the behavioral pattern mentioned earlier is that there are videos that despite having low views, the videos might have a great number of dislikes. This can be interpreted as videos being extremely disliked by the viewer, but they watch them anyway. Another behavioral pattern is there are videos that despite having high views, the videos might have a lesser number of dislikes (compared to the best of fit line). This can be interpreted as the videos being extremely liked by the viewers. Notes that the dataset is originally extracted from only high-interacted videos where the proposition videos of low views in the statistic are assumed higher than average videos on YouTube. From assumptions made earlier, the deduction is the video does not particularly need to be likable to viewers to reach high interactions.

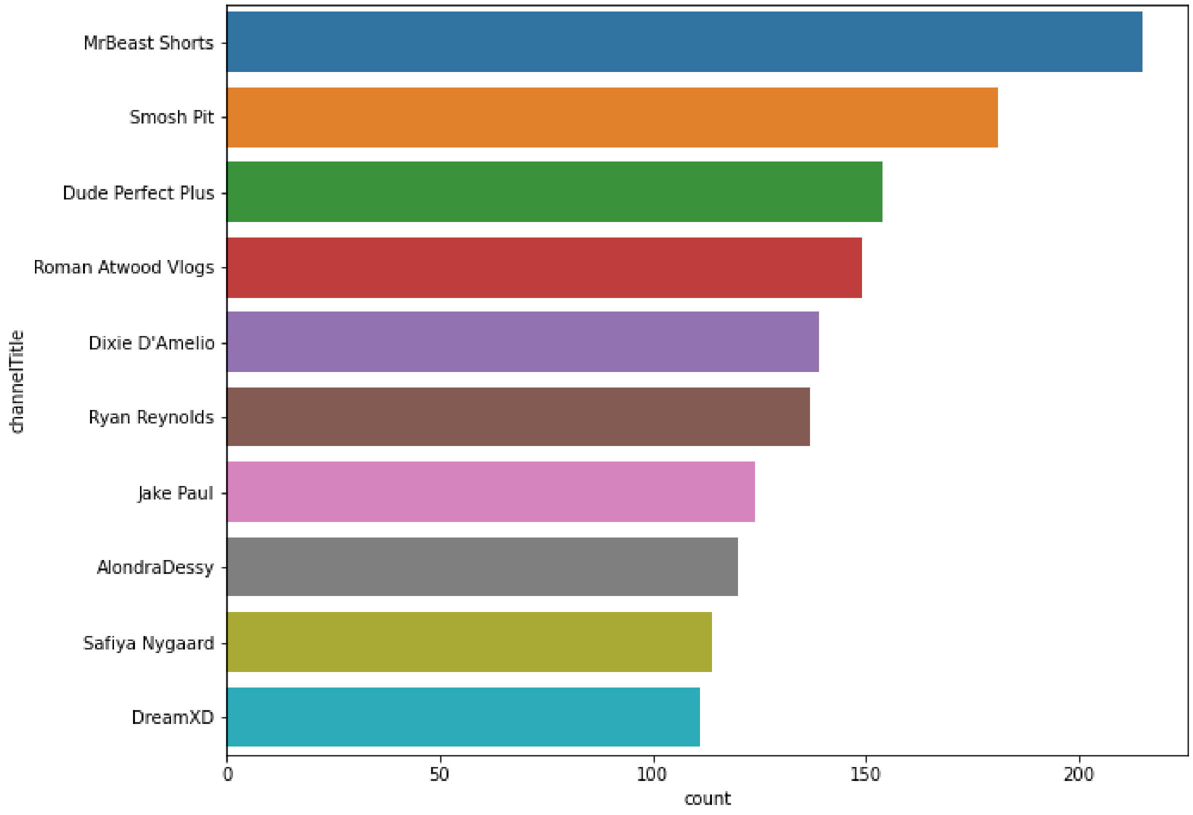


Figure 5.7: Top 10 Channels with most videos on trending

Figure 5.7 above shows that channels that have higher subscribers tend to have more of their uploaded videos trending.

5.1.3 Multivariate Analysis Findings

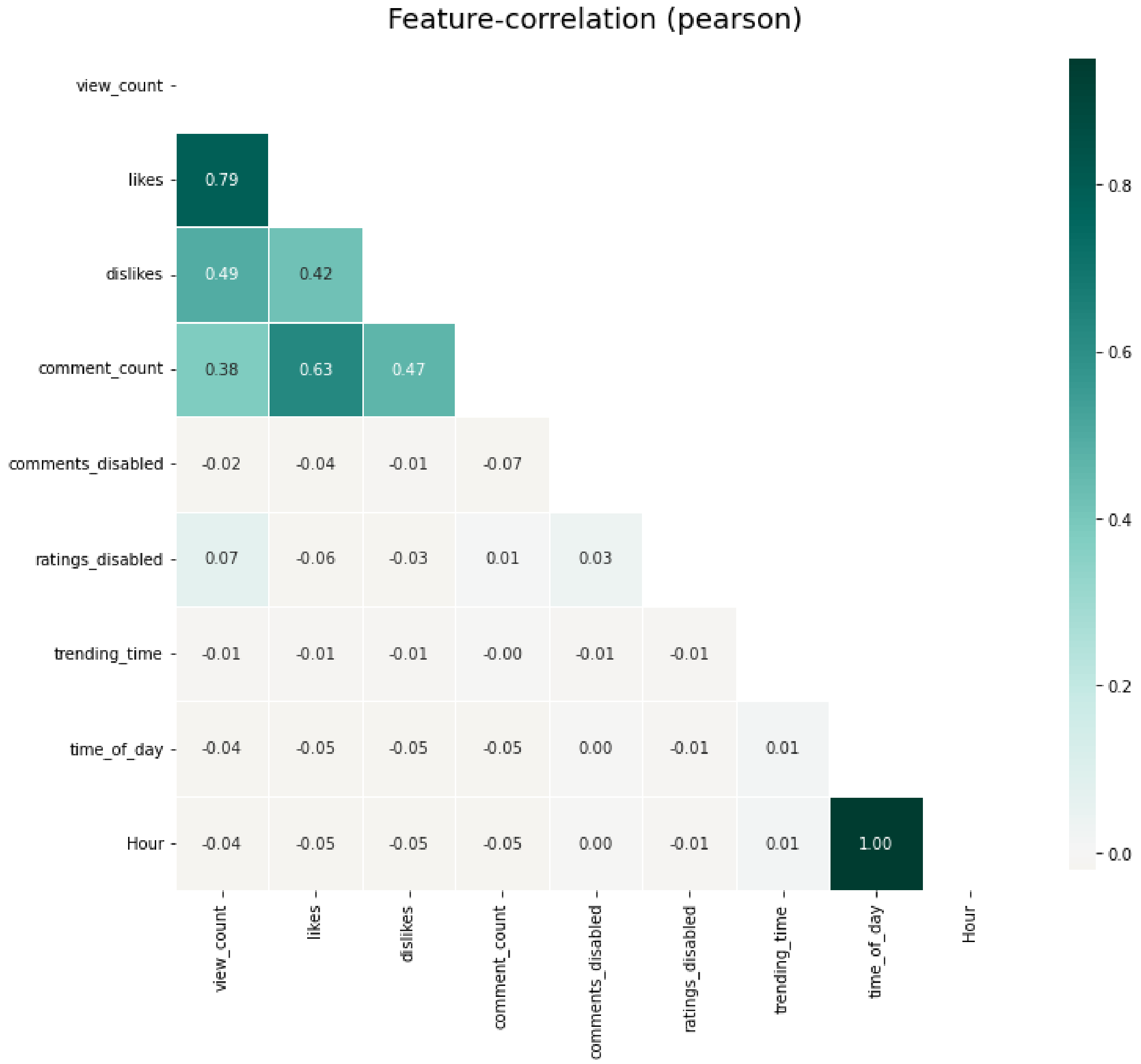


Figure 5.8: Pearson Feature Correlation Chart

Figure 5.8 shows the Pearson correlation chart for numerical features in the dataset. From the chart, some assumptions can be made. The first assumption is, likes are highly positively correlated to view counts with a 0.79 score meanwhile dislikes are on average positively correlated to view counts. Comments counts are highly positively correlated to likes, however, it has an average influence on view counts and dislikes.

5.1.4 Time Series Analysis Findings

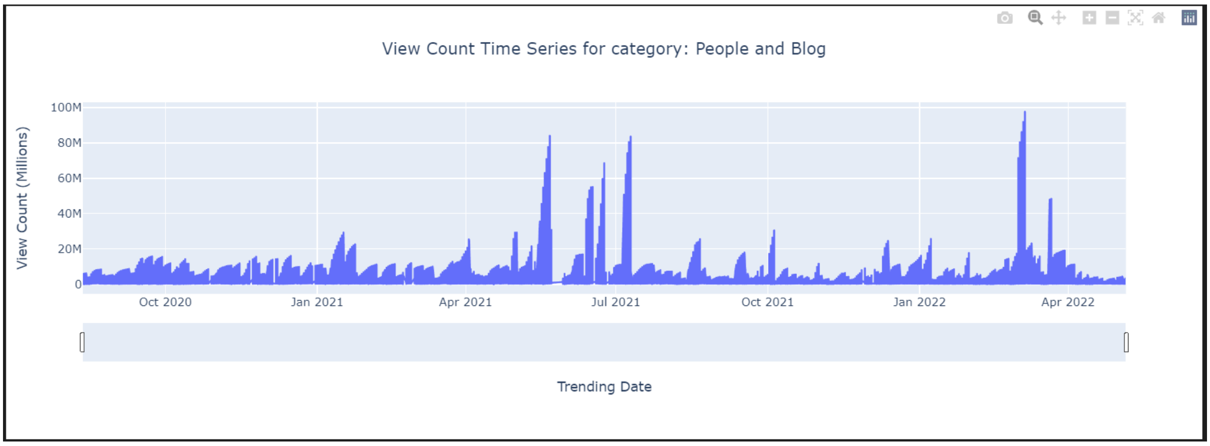


Figure 5.9: View Count Time Series for category People and Blog

Figure 5.9 shows that there are some unusual spikes of view counts over 80 million views at the timeframe:

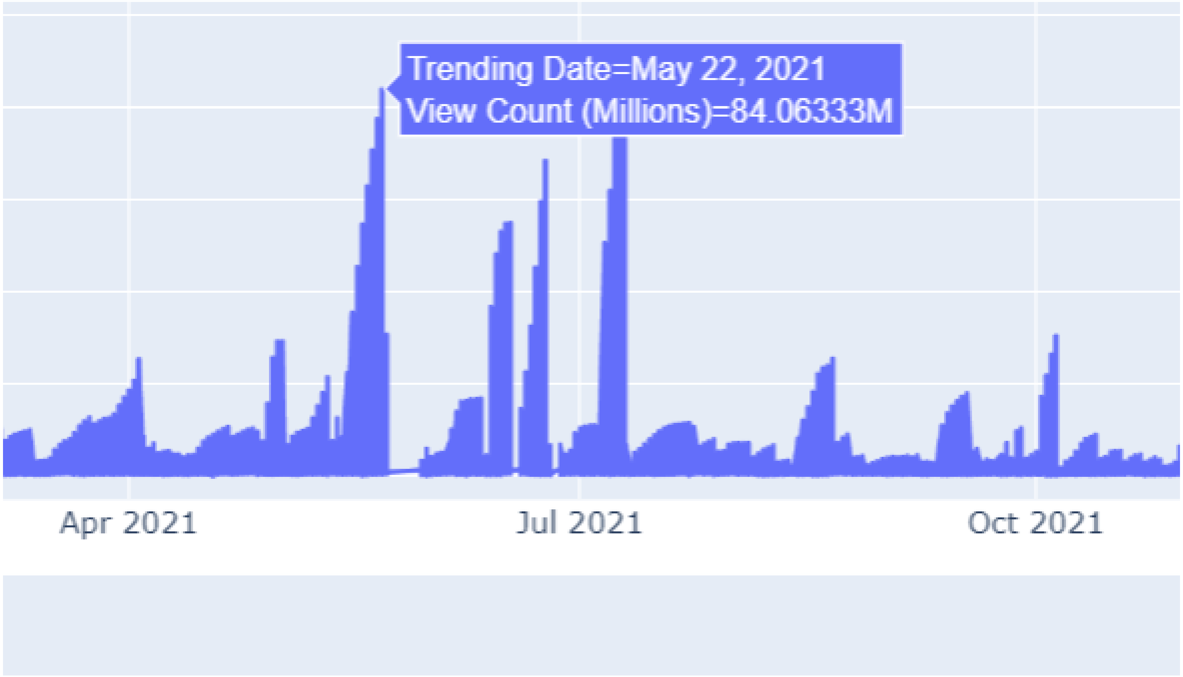


Figure 5.10: Views spike on 22 May 2021

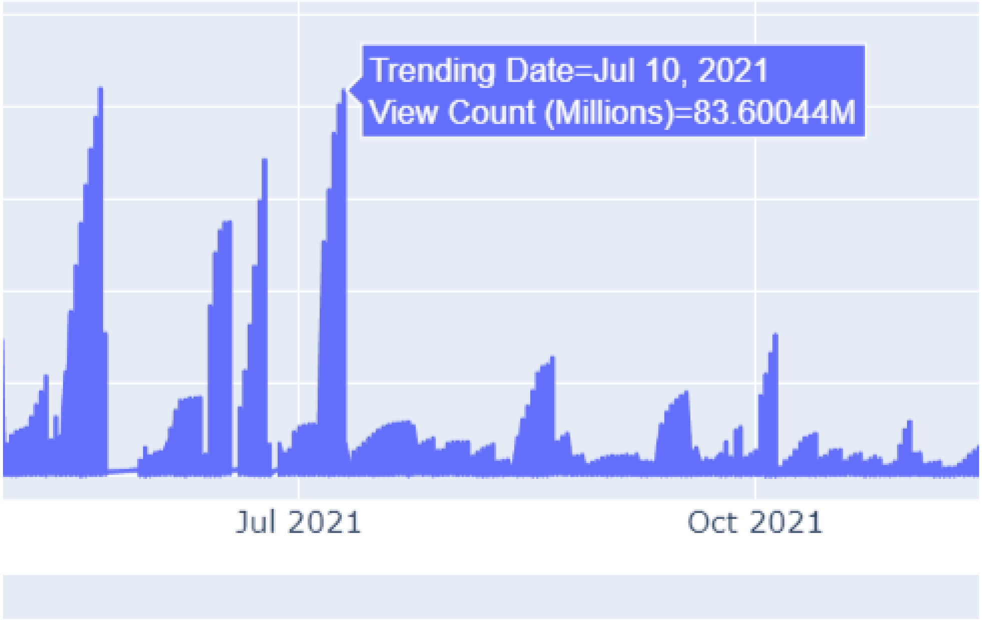


Figure 5.11: Views spike on 10 July 2021

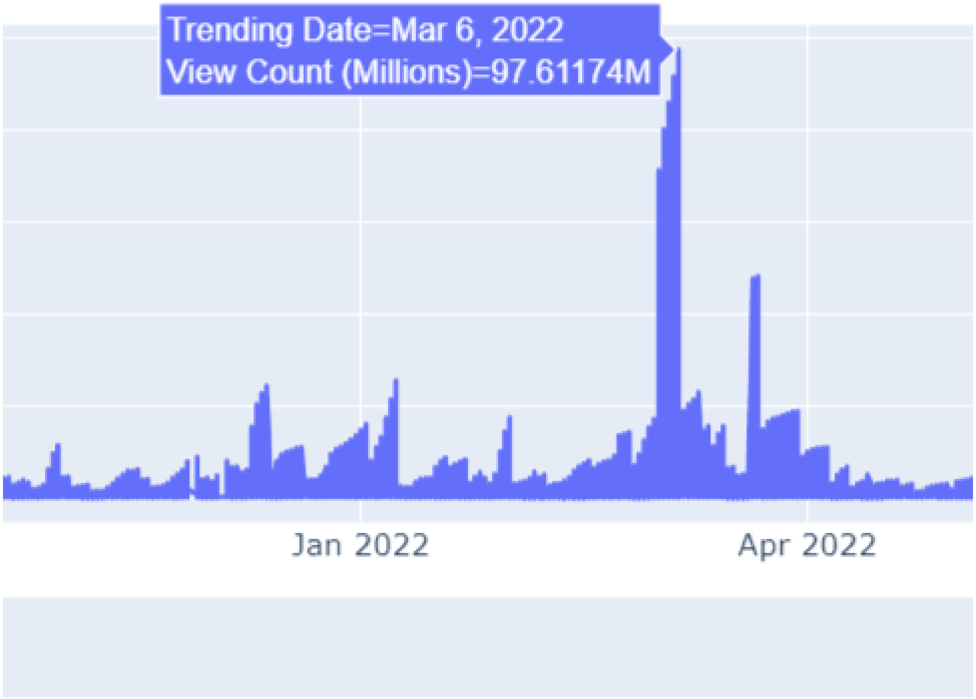


Figure 5.12: Views spike on 6 March 2022

Figure 5.10, *Figure 5.11*, and *Figure 5.12* show the exact date view count spiked on the time series plot. The timeframe is:

- 16th May 2021 to 23rd May 2021
- 6th July 2021 to 10th July 2021
- 22nd March 2022 to 26th March 2022

The unusual spikes may indicate that the view counts are influenced by events in the real world that happened at the timeframes. WordCloud is used to illustrate what events happened in the timeframe.

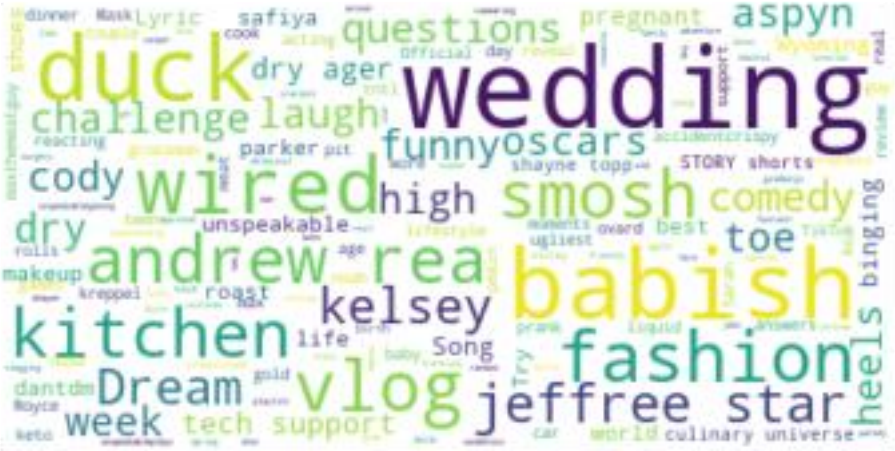


Figure 5.13: WordCloud for Timeframe 16 May 2021 – 23 May 2021

Figure 5.13 above shows the WordCloud timeframe that illustrates what videos are posted in the timeframe from 16th May 2021 to 23rd May 2021. From WordCloud, the words "Babish", "kitchen", and "Andrew rea" indicate that the video about cooking is trending in this timeframe. The first assumption is the views are influenced by the first pandemic wave that takes place in the timeframe. People around the world were stuck at home and that explains why cooking videos reached high interactions at that time.

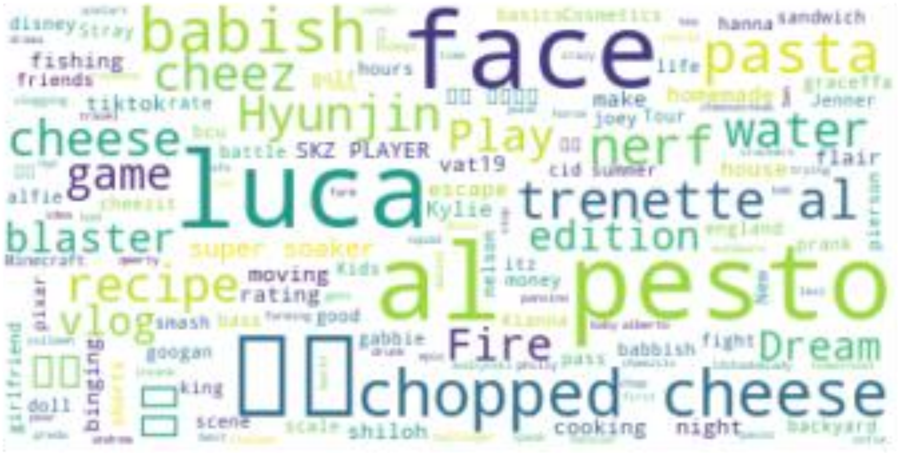


Figure 5.14: WordCloud for Timeframe 6 July 2021 to 10 July 2021

Figure 5.14 above shows the WordCloud that illustrates what videos are posted in the timeframe from 6th July 2021 to 10th July 2021. Some familiar words from the earlier timeframe discussed are spotted as this timeframe is not too far from the first timeframe discussed earlier. Word “pasta”, “Babish”, “Al pesto”, “chopped cheese”, “Luca”, and “trenette” indicates that the video about cooking is trending in this timeframe. The second assumption is the views are influenced by a second pandemic wave that takes place in the timeframe. People around the world were stuck at home once again like the first pandemic wave and that explains why cooking videos reached high interactions at that time.

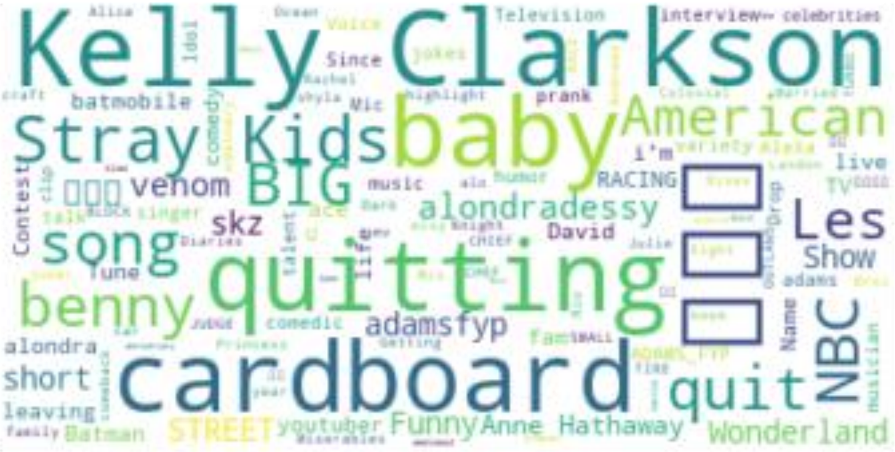


Figure 5.15: WordCloud for TimeFrame 22 March 2022 until 26 March 2022

Figure 5.15 above shows the WordCloud that illustrates what videos are posted in the timeframe from 22nd March 2022 to 26th March 2022. The words “Kelly Clarkson” and “Stray Kids” indicates singer. Many assumptions can be made about why sudden spike of views based on words related to the singer. For example, the singers produce new songs or reality TV shows about the singers. The words “quit” and “quitting” may indicate many people decided to change or quit their current jobs. Word “Cardboard” may indicate that videos about manipulating cardboards are trending at the time.

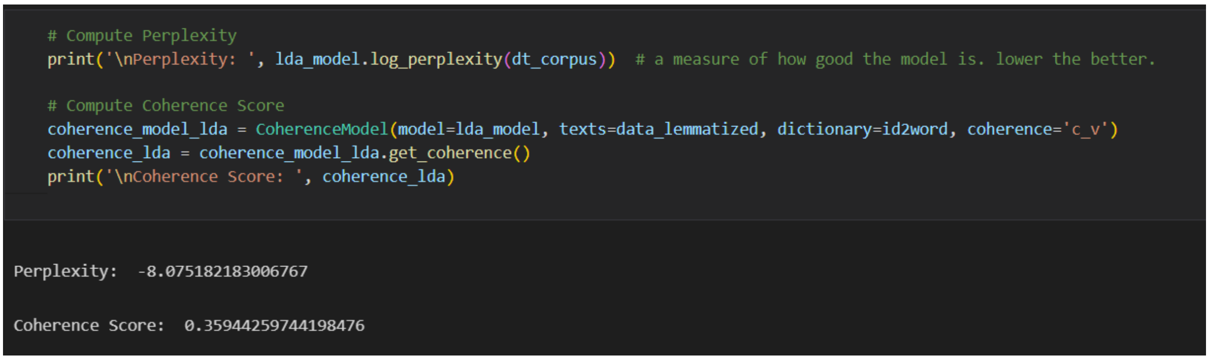
From 3 generated WordCloud, the finding is real-world events do impact greatly on view counts. The videos that are related to exact events in the real world tend to get more interactions than videos that are not related.

5.2 Discussion of Topic Modeling

The purpose of using topic modeling is to figure out what is the topic used on trending videos in the category “People and Blog”. Based on WordCloud in *Figure 5.13, Figure 5.14,* and *Figure* 5.15, familiar frequent words are spotted and some of them are inter-related. For example, the word “food” may consist of a different topic in the video context. An example of video variation that can be made is “videos about cooking food”, “videos about food reviews” or “food and hunger crisis”. Although the word “food” is used in the said topic, the context of each video has distinct values and different weightage. Topic modeling is intended to dive deep down into each word in the document to predict the topic of each video.

5.3 Results and Findings Based on Topic Modeling

For the base model, the number of topics is randomly assigned for *k* = 10 (in Python num_of_topic = 10). The result of the base LDA model is shown in *Figure 5.16* below.



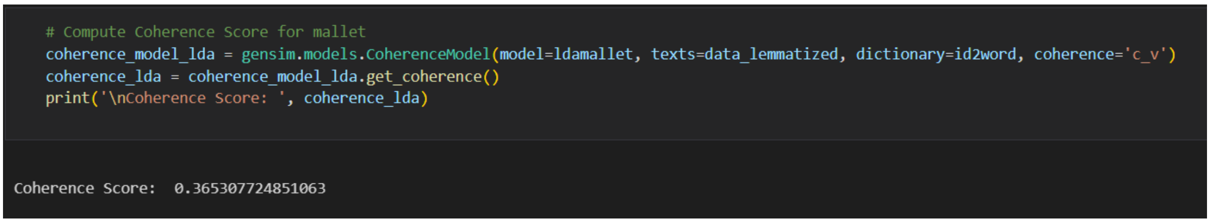
```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(dt_corpus))  # a measure of how good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Perplexity:  -8.075182183006767

Coherence Score:  0.35944259744198476
```

Figure 5.16: Perplexity and Coherence Score for base LDA model

Based on *Figure 5.16* above, the perplexity and coherence scores are observed. Observation is made to measure how good the LDA model is. For perplexity score, the lower the score, the higher chance that the base LDA model is good. A coherence score is used to evaluate a single topic's score by gauging the degree of semantic similarity between the topic's top-scoring words. The higher the coherence score, the better the chance the LDA model is good.

Next, the base LDA model is compared to the base LDA Mallet model that uses the Gibbs Sampling method. *Figure 5.17* below shows the result of the coherence score for the base LDA Mallet model.



```
    # Compute Coherence Score for mallet
    coherence_model_lda = gensim.models.CoherenceModel(model=ldamallet, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
    coherence_lda = coherence_model_lda.get_coherence()
    print('\nCoherence Score: ', coherence_lda)
```

```
Coherence Score:  0.365307724851063
```

Figure 5.17: Coherence score for base LDA Mallet model

Based on *Figure* 5.18, the coherence score for the base LDA Mallet model is slightly higher than the base LDA model. Hence, the LDA Mallet model is chosen to compute the optimal model. Then, the next step is tuning the model by finding the optimal number of topics. This is done by computing multiple LDA Mallet models and calculating their coherence score 'c_v' vs the number of topics, *k* for *k* is in the range of $2 \leq k \leq 40$. The results are plotted in a graph as shown in *Figure 5.18* below.

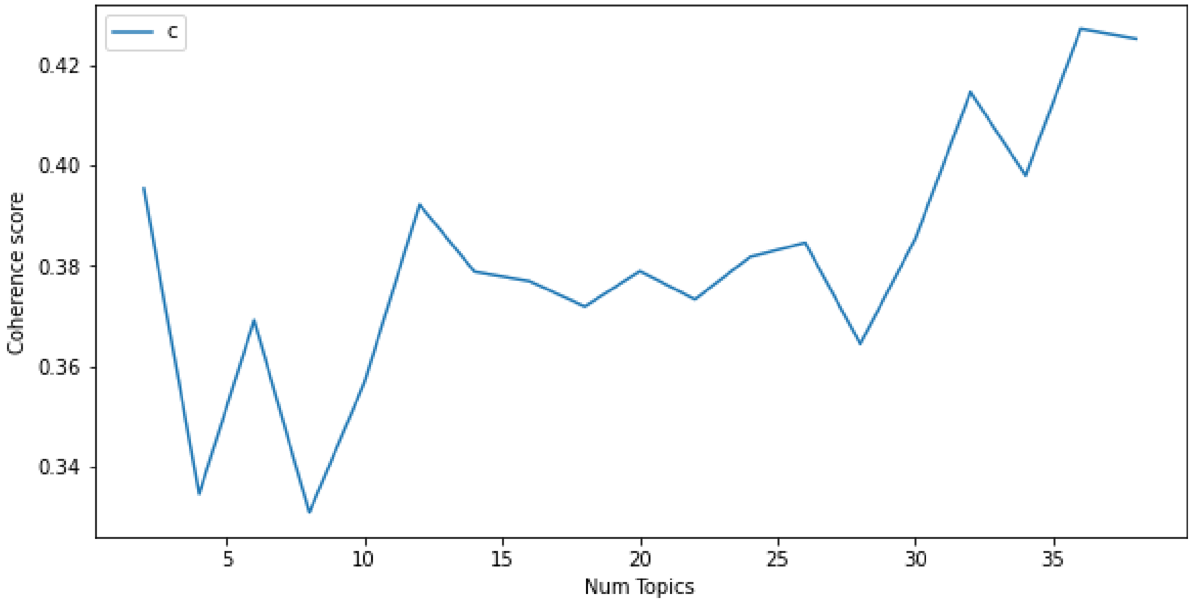
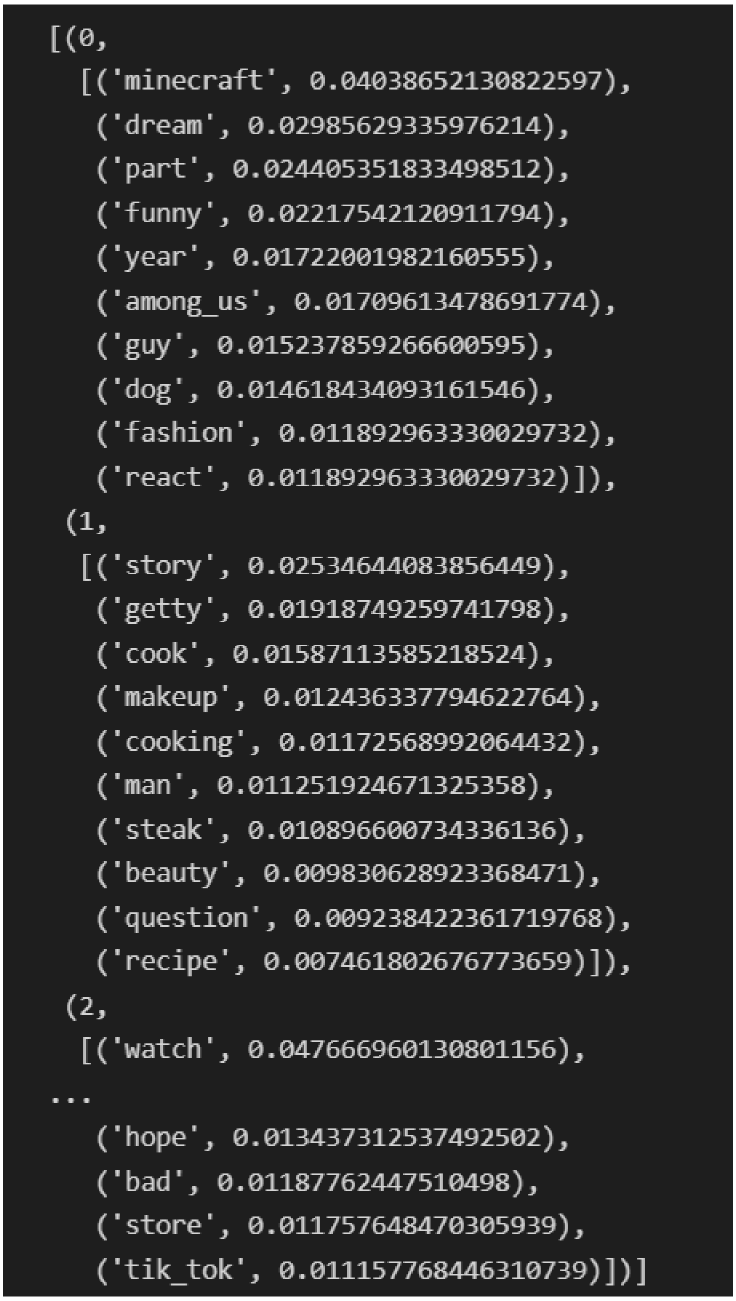


Figure 5.18: Graph Coherence Score vs Number of Topic

Based on the graph in *Figure 5.18*, the optimal number of topics is chosen subjectively at the point where the coherence score is at the peak before flattening out. Hence the chosen optimal number of topics, $k = 12$. By choosing $k = 12$, the optimal LDA Mallet is then built.

Next, each dominant topic is computed with the weightage of the keywords as shown in *Figure 5.19* below.

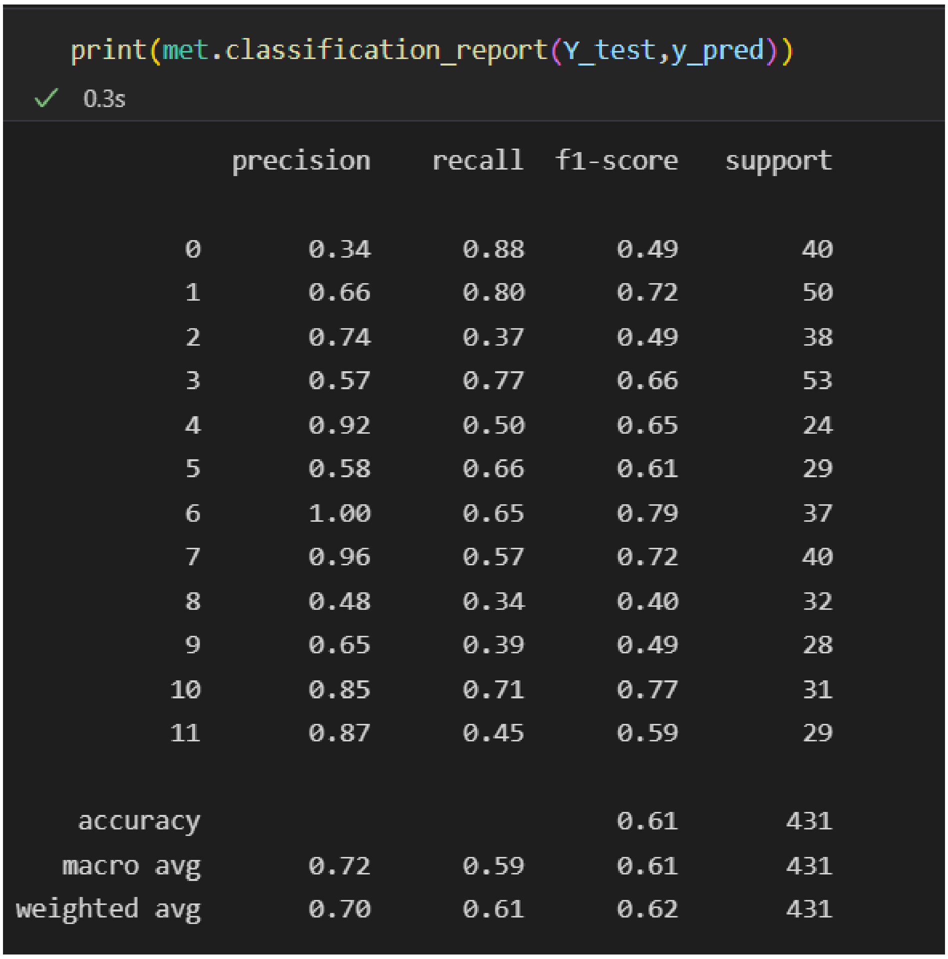


```
[(0,
  [('minecraft', 0.04038652130822597),
   ('dream', 0.02985629335976214),
   ('part', 0.024405351833498512),
   ('funny', 0.02217542120911794),
   ('year', 0.01722001982160555),
   ('among_us', 0.01709613478691774),
   ('guy', 0.015237859266600595),
   ('dog', 0.014618434093161546),
   ('fashion', 0.011892963330029732),
   ('react', 0.011892963330029732)]),
 (1,
  [('story', 0.02534644083856449),
   ('getty', 0.01918749259741798),
   ('cook', 0.01587113585218524),
   ('makeup', 0.012436337794622764),
   ('cooking', 0.01172568992064432),
   ('man', 0.011251924671325358),
   ('steak', 0.010896600734336136),
   ('beauty', 0.009830628923368471),
   ('question', 0.009238422361719768),
   ('recipe', 0.007461802676773659)]),
 (2,
  [('watch', 0.047666960130801156),
...
   ('hope', 0.013437312537492502),
   ('bad', 0.01187762447510498),
   ('store', 0.011757648470305939),
   ('tik_tok', 0.011157768446310739)])]
```

Figure 5.19: Dominant Topic for Optimal Model

The dominant topics are then used to predict each video in the dataset. To evaluate the performance of the optimal model, the confusion matrix is calculated by feeding the dominant topic and combining text (video title, video description, and video tags) in each video into two models, the Bag-of-Word model and TF-IDF model using supervised Random Forest Classification.

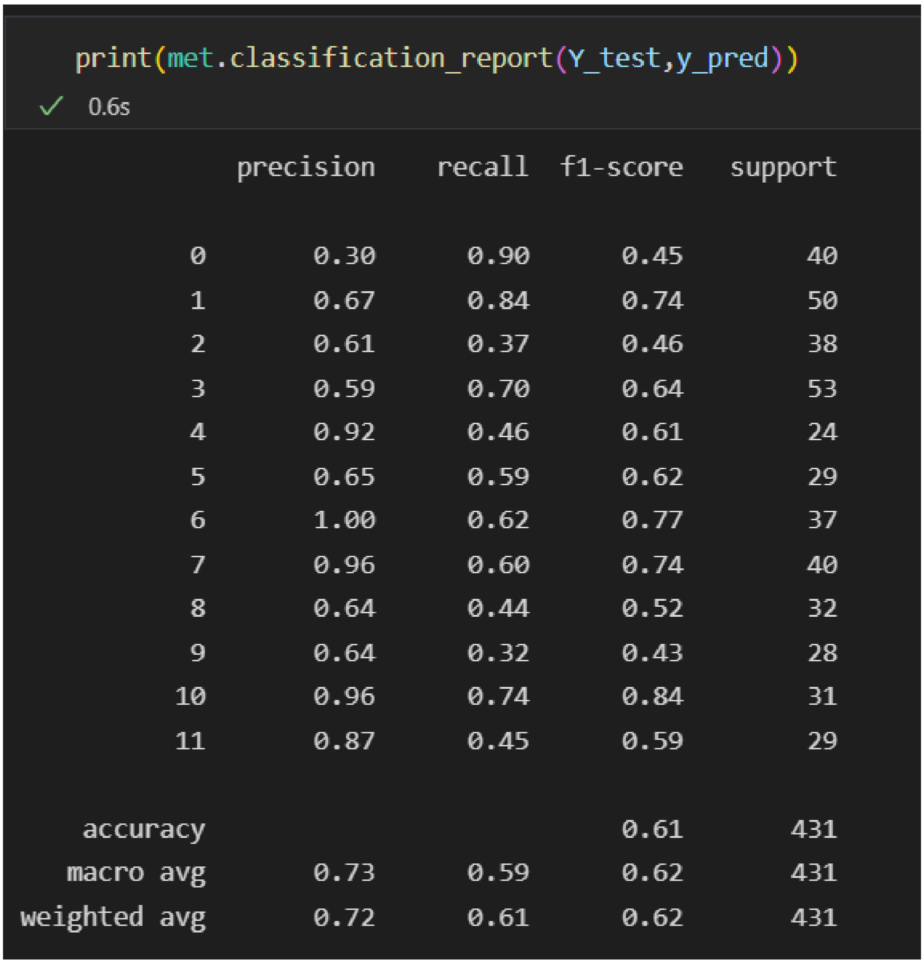
The classification report for Bag-of-model is shown in *Figure 5.20* below.



```
print(met.classification_report(Y_test,y_pred))
✓ 0.3s
              precision    recall  f1-score   support

           0       0.34      0.88      0.49        40
           1       0.66      0.80      0.72        50
           2       0.74      0.37      0.49        38
           3       0.57      0.77      0.66        53
           4       0.92      0.50      0.65        24
           5       0.58      0.66      0.61        29
           6       1.00      0.65      0.79        37
           7       0.96      0.57      0.72        40
           8       0.48      0.34      0.40        32
           9       0.65      0.39      0.49        28
          10       0.85      0.71      0.77        31
          11       0.87      0.45      0.59        29

    accuracy                           0.61       431
   macro avg       0.72      0.59      0.61       431
weighted avg       0.70      0.61      0.62       431
```

Figure 5.20: BoW Classification Report

Based on *Figure 5.20*, the performance of the BoW model is 0.61. The purpose of using the BoW model is to measure how well the document classification is based on the frequency occurrence of each word when it is used as a feature in the model classifier. While having a decent precision model, the low recall score made the F-1 score lowered to 0.62. For TF-IDF model evaluation, the classification result is shown in *Figure 5.21* below.



```
print(met.classification_report(Y_test,y_pred))
✓ 0.6s

              precision    recall  f1-score   support

           0       0.30      0.90      0.45        40
           1       0.67      0.84      0.74        50
           2       0.61      0.37      0.46        38
           3       0.59      0.70      0.64        53
           4       0.92      0.46      0.61        24
           5       0.65      0.59      0.62        29
           6       1.00      0.62      0.77        37
           7       0.96      0.60      0.74        40
           8       0.64      0.44      0.52        32
           9       0.64      0.32      0.43        28
          10       0.96      0.74      0.84        31
          11       0.87      0.45      0.59        29

    accuracy                           0.61       431
   macro avg       0.73      0.59      0.62       431
weighted avg       0.72      0.61      0.62       431
```

Figure 5.21: Classification Report for TF-IDF model

The purpose of using the TF-IDF model is to evaluate how well the document classification is based on the most relevant word in the document. Based on *Figure 5.21*, the performance of the TF-IDF model is also 0.61. While having a decent precision model, the low recall score made the F-1 score lowered to 0.61.

Based on the two evaluation model discussed, the performance of the model is assumed to be low bias but have high variance. The model is overfitted due to the fact when data pre-processing, only "English" words are selected to be fed into the model while the dataset should consist of different other languages' text. In the testing phase, the recall score has become low due to the incapability of the model to predict the text in other languages text. Hence the model only fitted for English Languages text.

5.5 Conclusion

This chapter has discussed Exploratory Data Analysis which consists of Univariate Analysis, Bivariate Analysis, Multivariate Analysis, and Time Series Analysis. The analysis has covered detailed reasonings of behavioral pattern for the data and relate to real-world events in term of statistic. The result of the analysis is visualized, and observation is explained. Other than that, the modeling process and results are discussed. Results and model performance is evaluated and compared in term of precision, recall, F-1 score, and support.

Chapter 6: Conclusion & Future Works

In this chapter, the objective achievements of the whole project, contributions, and limitations of the implementation of the project are outlined. The potential of future works for the future project is also discussed in this chapter to further enhance the research quality.

6.1 Project Achievement & Contributions

The purpose of this project is to discover how videos in the category of “People and Blogs” reached a high interaction. By the end of the project, three proposed objectives have been achieved as contributions of the project also been stated as follows:

Objective 1: To analyze the patterns of high-interaction videos on YouTube using Exploratory Data Analysis

Objective 2: To determine the topic consisted in the videos by using the Unsupervised Learning method.

Objective 3: To discover optimal settings needed on the posted videos to achieve high interactions.

Exploratory Data Analysis has shown several findings. Firstly, the usage of the new feature is effective to attract viewers. Secondly, likes are not always dependent on the view count and the deduction is the video does not particularly need to be likable to viewers to reach high interactions. Some possibilities are taken from the analysis where exceptional videos are made without having many likes. Next, the finding is real-world events do impact greatly on view counts. The videos that are related to exact events in the real world tend to get more interactions than videos that are not related. Using unsupervised machine learning, the main topic of trending video is discovered. Content creators might use the main topic produced as a guideline to generate title, description, and tags for their video as it was used by other successful content creators to make their video appearance stands out.

6.2 Limitations

- *Lack of computational power*

  One of the limitations would be the lack of computational power to train multiple LDA models. Training unsupervised Natural Language Processing learning takes a lot of resources to compute. This makes the model tuning process time-consuming and cannot fully tune every hyperparameter as it will take more RAM and CPU power.

- *Lack of other language resources*

  Lastly, even though the dataset is observed for trending videos in the United States, the dataset consists of videos from other regions that are successful globally. The text cleaning process relies heavily on Natural Language Processing for English. The videos cannot be treated as outliers as the number of videos from other regions is impactful in other analyses except text columns. Removing another language besides English has proven lower recall scores in the cross-validating process.

6.3 Future Works

As this project ended, there are future works that can be addressed as follows:

- Wider category exploration for YouTube videos
- The variance of the global NLP package to use for topic modeling
- Better computational power to sustain more NLP-based machine learning models

6.4 Conclusion

By the end of the project, three objectives have been achieved, and patterns of high-interaction videos on YouTube are discovered using EDA and topic modeling. The second objective, optimal settings needed on the posted videos to achieve high interaction has been achieved by using the main topic produced by the topic model. Lastly, small content creators gain insights from other successful content creators on how to grow more audiences by implementing the discussion and findings made in this project.

Reference

[1] Snickars, P., & Vonderau, P. (2009). *The youtube reader*. Kungliga biblioteket.

[2] Holmbom, M. (2015). The YouTuber: A qualitative study of popular content creators.

[3] Rowley, J. (2004). Online branding. Online Information Review, 28(2), 131-138.

[4] Biel, J. I., & Gatica-Perez, D. (2010, November). Vlogcast yourself: Nonverbal behavior and attention in social media. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction* (pp. 1-4).

[5] Biel, J. I., Aran, O., & Gatica-Perez, D. (2011, July). You are known by how you vlog: Personality impressions and nonverbal behavior in youtube. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 5, No. 1).

[6] Luers, W. (2007). *Cinema without show business: A poetics of vlogging* (Vol. 5, No. 1). Ann Arbor, MI: MPublishing, University of Michigan Library.

[7] Wauters, Robin. (2010, January 5). State of the vlogosphere. Retrieved from http://techcrunch.com/2010/01/05/mefeedia-state-of-the-vlogosphere-2010/

[8] Mitchell, J. C. (1969). The concept and use of social networks. In J. C. Mitchell (Ed.), Social networks in urban situations. Manchester, England: University of Manchester Press, 1969.

[9] Arthurs, J., Drakopoulou, S., & Gandini, A. (2018). Researching youtube. *Convergence*, *24*(1), 3-15.

[10] Dehghani, M., Niaki, M. K., Ramezani, I., & Sali, R. (2016). Evaluating the influence of YouTube advertising for attraction of young customers. *Computers in human behavior*, *59*, 165-172.

[11] Usmani, Z. (2017). *Kaggle for Beginners: with Kernel Code*. Gufhtugu Publications.

[12] Ghojogh, B., Samad, M. N., Mashhadi, S. A., Kapoor, T., Ali, W., Karray, F., & Crowley, M. (2019). Feature selection and feature extraction in pattern analysis: A literature review. *arXiv preprint arXiv:1905.02845*.

[13] Everitt, B. S., Landau, S., & Leese, M. (2001). Cluster Analysis, ch. 4.

[14] Wu, K. (2016). YouTube marketing: Legality of sponsorship and endorsements in advertising. *JL Bus. & Ethics*, *22*, 59.

[15] Firat, D. (2019). YouTube advertising value and its effects on purchase intention. *Journal of Global Business Insights*, *4*(2), 141-155.

[16] Bendersky, M., Garcia-Pueyo, L., Harmsen, J., Josifovski, V., & Lepikhin, D. (2014, August). Up next: retrieval methods for large scale related video suggestion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1769-1778).

[16] Choudhury, S., & Breslin, J. G. (2010). User sentiment detection: a YouTube use case.

[17] Rogers, R. (2013). *Digital methods*. MIT press.

[18] Cheng, X., Dale, C., & Liu, J. (2007). Understanding the characteristics of internet short video sharing: YouTube as a case study. *arXiv preprint arXiv:0707.3670*.

[19] Rinaldi, E., & Musdholifah, A. (2017, November). FVEC-SVM for opinion mining on Indonesian comments of youtube video. In *2017 International Conference on Data and Software Engineering (ICoDSE)* (pp. 1-5). IEEE.

[20] Heckert, N. A., Filliben, J. J., Croarkin, C. M., Hembree, B., Guthrie, W. F., Tobias, P., & Prinz, J. (2002). Handbook 151: NIST/SEMATECH e-Handbook of Statistical Methods.

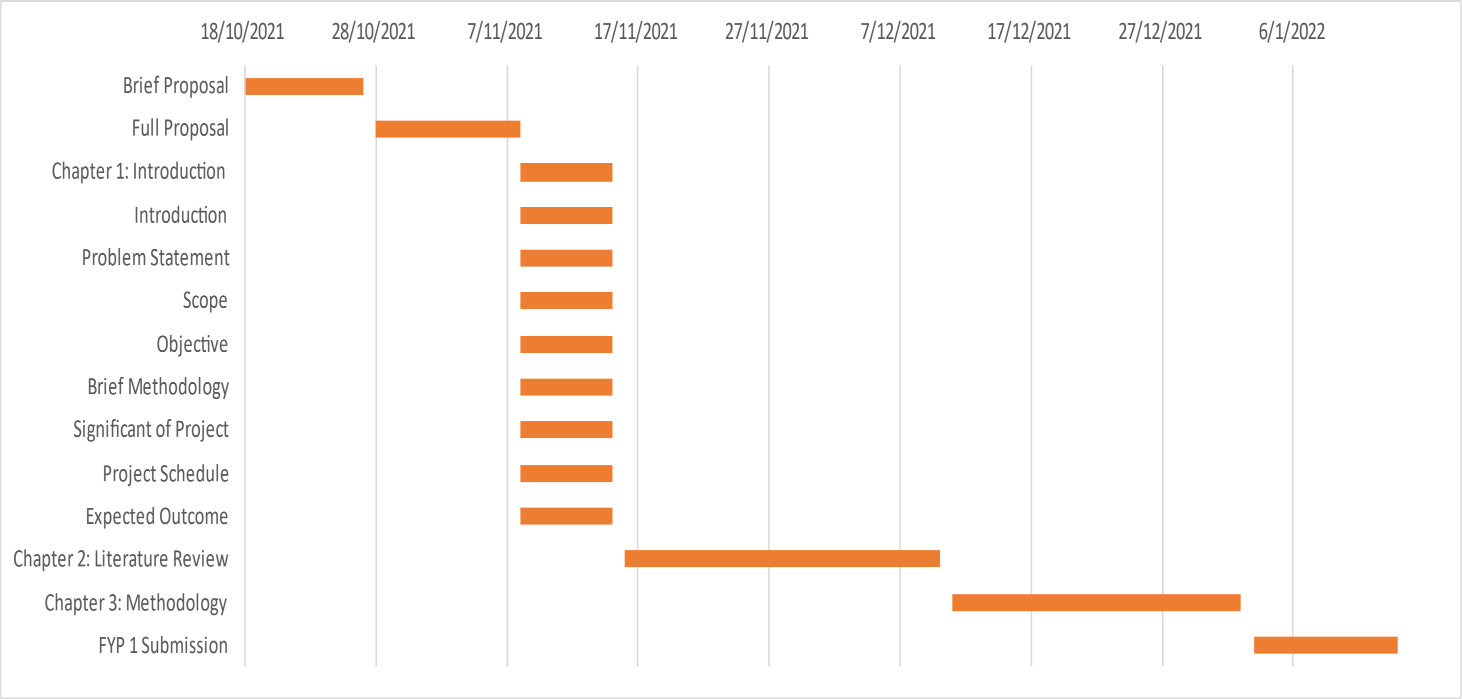
[21] Khanam, S., Tanweer, S., & Khalid, S. S. (2021). Youtube Trending Videos: Boosting Machine Learning Results Using Exploratory Data Analysis. *The Computer Journal*.

[22] Airoldi, M., Beraldo, D., & Gandini, A. (2016). Follow the algorithm: An exploratory investigation of music on YouTube. *Poetics*, *57*, 1-13.

[23] Severyn, A., Uryupina, O., Plank, B., Moschitti, A., & Filippova, K. (2014). Opinion mining on YouTube.

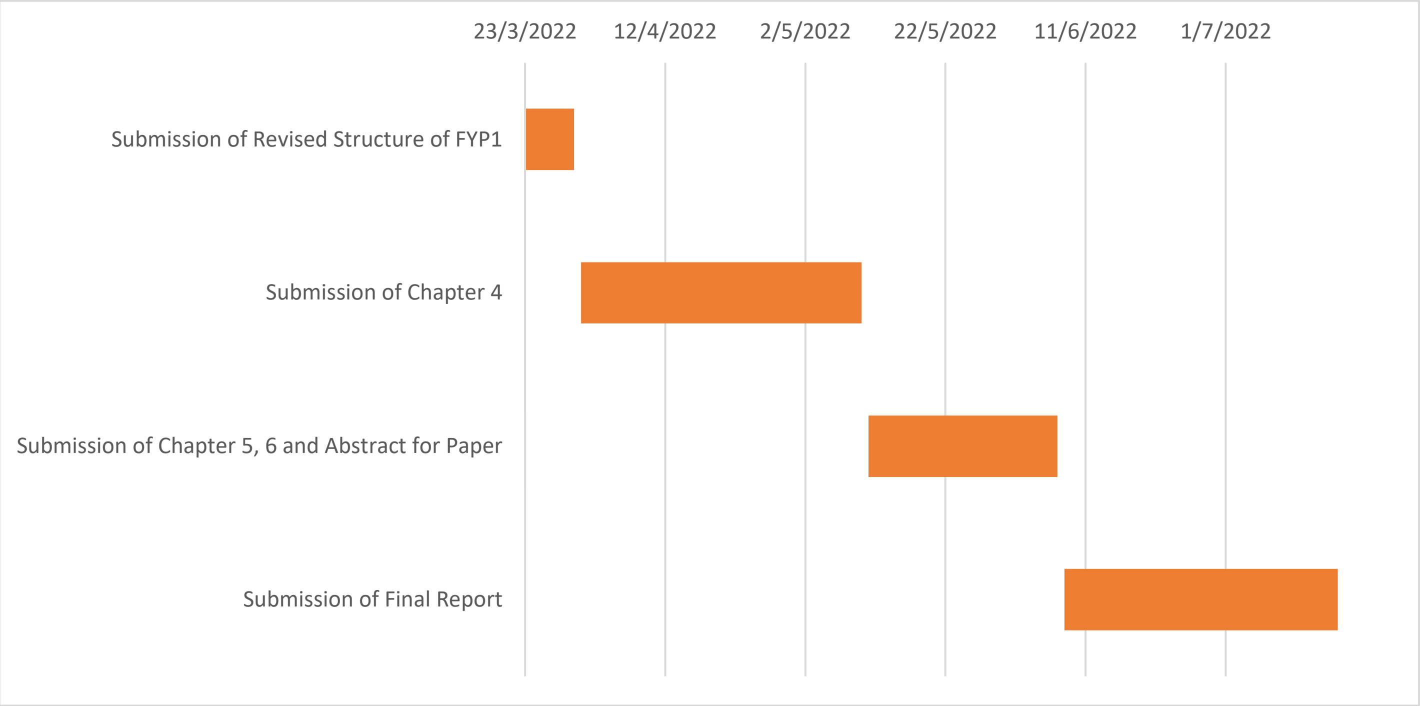
APPENDIX A

FYP 1 Project Schedule

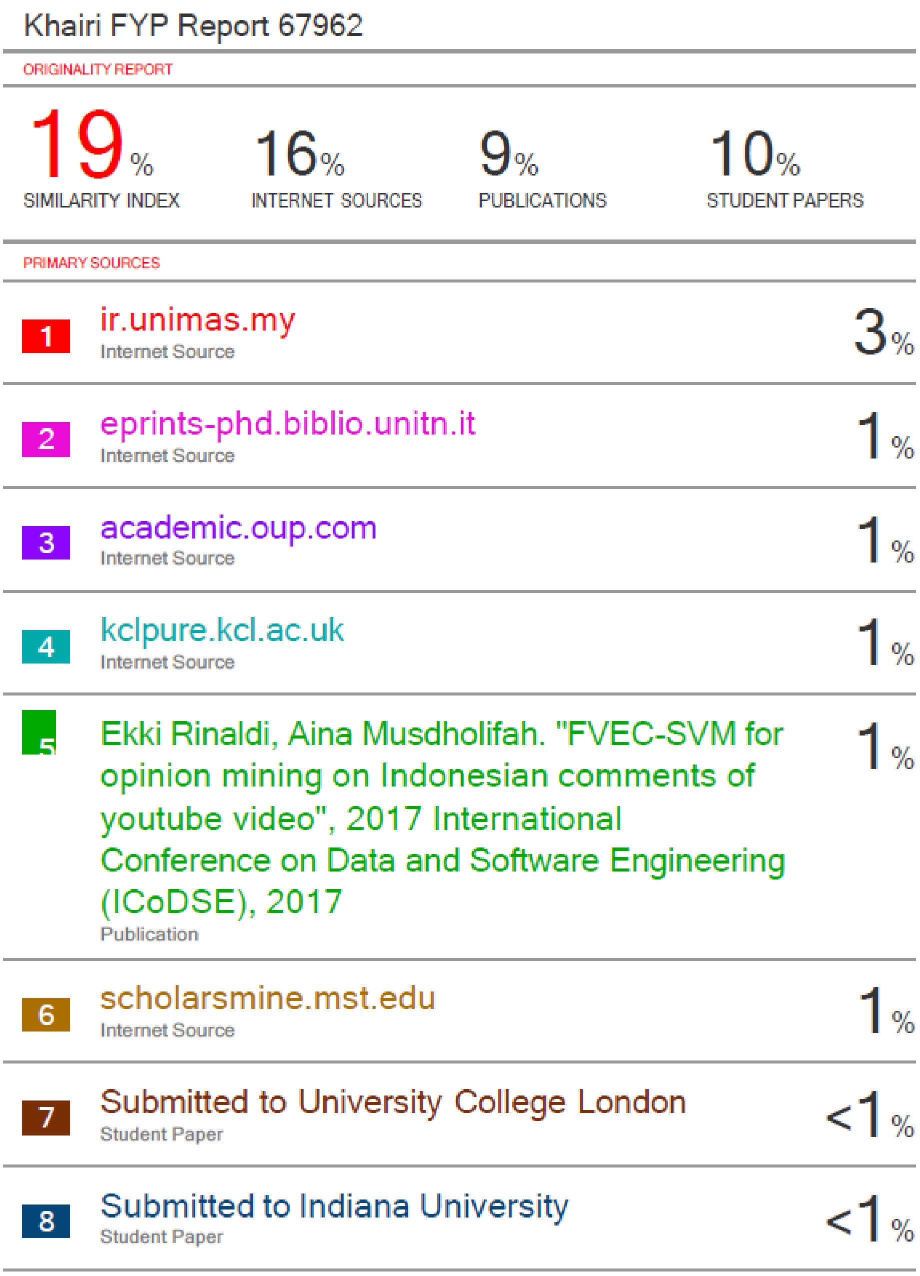


APPENDIX B

FYP 2 Project Schedule



APPENDIX C



Khairi FYP Report 67962

ORIGINALITY REPORT

19%	16%	9%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ir.unimas.my Internet Source	3%
2	eprints-phd.biblio.unitn.it Internet Source	1%
3	academic.oup.com Internet Source	1%
4	kclpure.kcl.ac.uk Internet Source	1%
5	Ekki Rinaldi, Aina Musdholifah. "FVEC-SVM for opinion mining on Indonesian comments of youtube video", 2017 International Conference on Data and Software Engineering (ICoDSE), 2017 Publication	1%
6	scholarsmine.mst.edu Internet Source	1%
7	Submitted to University College London Student Paper	<1%
8	Submitted to Indiana University Student Paper	<1%